Université Paris Nanterre

200 av. de la République
92001 Nanterre Cedex
www.u-paris10.fr

École doctorale 139 :
Connaissances, Langage,
Modélisation

Membre de l'université Paris Lumières

Ilaine Wang

Syntactic Similarity Measures in Annotated Corpora for Language Learning

Application to Korean Grammar

Thèse présentée et soutenue publiquement le 17/10/2017
en vue de l'obtention du doctorat de Sciences du langage
de l'Université Paris Nanterre

sous la direction de M. Sylvain Kahane (Université Paris Nanterre) et de Mme Isabelle Tellier (Université Paris 3 Sorbonne Nouvelle)

Jury :

Rapporteur·e :	Pr Angela Chambers	Professeur émérite, Université de Limerick
Rapporteur·e :	Dr Olivier Kraif	HDR, Université Grenoble-Alpes
Membre du jury :	Dr Benoît Crabbé	Université Paris-Diderot
Membre du jury :	Dr Jin-Ok Kim	Université Paris-Diderot
Membre du jury :	Dr Christian Surcouf	Université de Lausanne
Directeur :	Pr Sylvain Kahane	Université Paris Nanterre
Directrice :	Pr Isabelle Tellier	Université Paris 3 Sorbonne Nouvelle

Université Paris Nanterre

École doctorale 139 – Connaissance, Langage, Modélisation

Syntactic Similarity Measures in Annotated Corpora for Language Learning: Application to Korean Grammar

par Ilaine Wang

❧

Thèse présentée et soutenue publiquement le 17 octobre 2017

en vue de l'obtention du grade de
docteur en Traitement Automatique des Langues

sous la direction de Sylvain Kahane et d'Isabelle Tellier

Membres du jury:

Directeur: Pr Sylvain Kahane	Université Paris Nanterre, MoDyCo
Directrice: Pr Isabelle Tellier	Université Sorbonne-Nouvelle, LATTICE
Rapporteur: Emeritus Pr Angela Chambers	University of Limerick, CALS
Rapporteur: Dr HDR Olivier Kraif	Université Grenoble Alpes, LIDILEM
Examinateur: Dr Benoît Crabbé	Université Paris Diderot, LLF
Examinatrice: Pr Iris Eshkol-Taravella	Université Paris Nanterre, MoDyCo
Examinatrice: Dr Jin-Ok Kim	Université Paris Diderot, CRC, PLIDAM, GRAC
Examinateur: Dr Christian Surcouf	Université de Lausanne, EFLE

Abstract

Using queries to explore corpora is today part of the routine of not only researchers of various fields with an empirical approach to discourse, but also of non-specialists who use search engines daily. While both corpus linguistics softwares and search engines allow for complex keyword-based queries which can be extended with methods relying on lexical similarity measures, none seem to allow to find syntactically similar phrases so far. For instance, a person who is working on relative clauses cannot retrieve the two phrases "the person whom I see" and "that dream that you had", which share no common lexical items but the same syntactic structure, unless they do a specific query like "`DET NOUN which|that|who|whom PRO VERB`". Such queries require the use of regular expressions with grammatical words (or morphemes) eventually combined with morphosyntactic tags, which imply that users master both the query system of the tool and the tagset of the annotated corpus. However, non-specialists like language learners might want to focus on the output rather than spend time and effort on mastering a query language.

Indeed, when a language learner encounters an unknown grammatical construction, one solution is to look it up in textbooks or in grammars, where a definition, as well as several examples of canonical uses, are provided. However, in some cases, explicit rules and a small number of uses are not sufficient to fully comprehend a grammatical construction, especially if the learner's native language is typologically distant from the target language. The next step could be to search more examples, perhaps in authentic corpora to observe and analyse what is considered as natural and usual in the target language. Learners would

therefore be actors of the construction of their own knowledge, which was encouraged by Johns's Data-driven learning approach. However, using a grammatical construction as a query may not be as easy as using plain words to obtain concordances. Indeed, learners would need to provide a description of the construction, which is not self-evident for non-specialists.

In this study, we present our efforts to provide the missing link between examples taken from textbooks to illustrate grammatical constructions and subsidiary instances of those constructions that can be found in context in native corpora. We propose a methodology using common similarity measures (Dice, Jaccard and Levenshtein distance) that we adapted to syntax-related queries. Instead of comparing sequences of keywords, we measure the similarity between sequences of morphosyntactic tags. No prior knowledge is asked from users as the POS tags would automatically be provided by an open source morphological analysis tool which tagset is identical to the corpus tagset. Following this method, it is possible to use complex syntactic queries as long as the target language has a treebank and an effective parser. Our study describes variants which have been implemented and experimented on the Sejong Korean corpus.

From the user's perspective, the process simply works like a syntax-based search engine: from a sentence in input containing the targeted grammatical construction, our tool provides other sentences in context, ranked by the similarity of their construction. As an illustration, we could retrieve hundreds of relevant examples of a given construction based on a few examples displayed in a textbook, including similar constructions which are not mentioned in grammars as possible variations. The focus of our study is on Korean language learners, but the methodology could be extended to any language and teachers are the other evident target as this method can be useful in the preparation of teaching materials.

Contents

List of Figures

List of Tables

Transliteration

This classification of Korean graphemes is inspired by Chun Ji-Hye's classification [Chun, 2013], which is based on the recommendations of the National Institute of Korean Language[1]. The first row of the tables are graphemes in *hankul* 한글, the Korean alphabet, and the second row contains the transliteration of the sounds.

Like Ji-Hye, we correctly classified the graphemes ㅚ and ㅟ as diphtongs, and we added a third row in the tables to include the phonetics from the International Phonetic Alphabet (IPA). However, instead of the official Revised Romanisation of Korean (국어의 로마자 표기법), we chose to use the Yale transliteration, developed specifically for linguistics studies.

Vowels

Simple

ㅏ	ㅓ	ㅗ	ㅜ	ㅡ	ㅣ	ㅐ	ㅔ
a	e	o	u/wu	u	i	ay	ey
ɑ	ʌ	o	u	ɯ	i	ɛ	ɛ

Diphthongs

ㅑ	ㅕ	ㅛ	ㅠ	ㅒ	ㅖ	ㅚ	ㅘ	ㅙ	ㅝ	ㅞ	ㅟ	ㅢ
ya	ye	yo	yu	yay	yey	oy	wa	way	we	wey	wi	uy
jɑ	jʌ	jo	ju	jɛ	jɛ	wɛ	wɑ	wɛ	wʌ	wɛ	wi	ɯi

[1] http://www.korean.go.kr/front_eng/roman/roman_01.do, retrieved on 4th January 2017.

Consonants

Plosive (stops)

ㄱ	ㄲ	ㅋ	ㄷ	ㄸ	ㅌ	ㅂ	ㅃ	ㅍ
k	kk	kh	t	tt	th	p	pp	ph
g,k*	k̬	k^h	d,t*	t̬	t^h	b,p*	p̬	p^h

Affricates

ㅈ	ㅉ	ㅊ
c	cc	ch
ʤ,ʨ*	ʨ̬	t^h

Fricatives

ㅅ	ㅆ	ㅎ
s	ss	h
s,ɕ	s̬,ɕ̬	h,ɦ

Nasals

ㄴ	ㅁ	ㅇ
n	m	ng
n	m	ŋ

Liquid

ㄹ
l
ɾ

Linguistic Glosses

Most abbreviations used in linguistic glosses follow the Leipzig glossing rules[2], updated on 31st May 2015. For morpheme glosses that are specific to the Korean language, we referred to Ho-Min Sohn's reference book on Korean Linguistics, *Korean* [Sohn, 2013]. They were marked with an asterisk in this list.

Glosses are used in linguistic examples which may come from the author's imagination, from the above-mentioned reference book by Sohn, from the *Korean Grammar for International Learners* by Ho-Bin Ihm, Kyung-Pyo Hong and Suk-In Chang [Im et al., 2012] or from the Sejong Corpus. The origin of the example is indicated in brackets:

- [Sohn_page] for Sohn's book,
- [KGIL_page] for Ihm et al.'s book
- the ID number of the sample for the Sejong Corpus. Samples from the spoken corpus start with a digit, while samples from the written corpus start with 'BR' (raw), 'BT' (POS-tagged), 'BS' (disambiguated POS-tagged) or 'BG' (syntactically parsed).

[2] Available at: https://www.eva.mpg.de/lingua/pdf/Glossing-Rules.pdf

Abbreviation	**Label**
ADV	adverbial
AH*	addressee honorific suffix
DECL	declarative
FQ*	frequentative
IND	indicative mood
INF	infinitive mood
INS	instrumental
LOC	locative
MD*	pre-nominal modifier
NMLZ	nominaliser
NOM	nominative
OBJ	object
POL*	polite speech level
PR*	propositive sentence-type suffix
PRS*	prospective
Q	question marker
QUOT	quotative
RQ*	requestive mood suffix
RT*	retrospective mood suffix
SH*	subject honorific
SUP*	suppositive mood suffix
TOP	topic

Chapter 1

Introduction

The work presented in this dissertation tackles the problem of seeking constructions that are syntactically similar to a given construction, in the context of language learning. While annotated corpora are ideal resources for such a search, access to them is still limited to specialists and their exploration is limited by a strict matching system. Our objective is to go beyond those limitations and to account for the use of syntactic similarity research in the acquisition of grammatical constructions.

We rely on knowledge from several fields, including Corpus Linguistics, Natural Language Processing and Language Acquisition, to propose a tool that contributes to the demystification of grammar, helps language learners in their apprehension of grammatical constructions and encourages the use of a wide range of resources in language learning and teaching.

1.1 Background

The topic of this doctoral dissertation was defined over weeks of discussions with my supervisors while I was still working on my master's project, a tool that automatically segments spoken French into macrosyntactic units. While the current research problem is completely different from the previous one, we note that they do share common elements: an interest in syntax, the use of corpora, and the construction of an automatic processing chain. The first element is thoroughly commented on in the following section, where we clarify the particular focus on

grammar of this work and we also define the grammar(s) that we refer to. As for the two remaining common elements, they are a direct consequence of my studies in Natural Language Processing.

The active research and community in Natural Language Processing show that this discipline carries as many challenges as offered by both the complexity of natural language and the development of technical means and methods. Among those challenges, what particularly caught my attention was the tremendous work that has been done and still is being done around corpora: from the collection of samples to their processing and annotation, through the widening of the variety of corpora. Despite the growing interest in Corpus Linguistics for decades, we are still under the impression that the use of corpora has no limit, be it in its extended applications to other disciplines or in the construction of linguistic resources and tools.

A good example of such possibilities is Linguee[1], a tool that I use frequently and that relies on the exploitation of parallel corpora, i.e., multilingual corpora that are aligned – on sentences in this case, to provide not only a usage-based bilingual dictionary, but also a KWIC (KeyWord In Context) display to see the search word(s) and the corresponding translations in context.

All of the studies and tools that I was confronted with, especially in lexicometry, as well as works that I have contributed to during my two internships,[2] have convinced me that linguistic studies should be usage-based. This work is therefore fully inscribed in a usage-based, and specifically corpus-based, approach.

The choice of application of a corpus-based approach to language learning is simply due to my own experience as a language learner. I grew up in an unbalanced bilingual environment as a child[3] and have since been lucky to find opportunities to

[1] http://www.linguee.com/

[2] My first intership focused on the linguistic specifications of the segmenter and parser SEM developed by Yoann Dupont, under the supervision of Isabelle Tellier and Iris Eshkol-Taravella. It is described on http://www.lattice.cnrs.fr/sites/itellier/SEM.html and has an online version on http://apps.lattice.cnrs.fr/sem/. My second internship resulted in the macrosyntactic segmenter that I mentioned at the beginning of this section, which is described in Wang et al. [2014].

[3] Such a linguistic background is explained in more detail in the "Language proficiency" para-

learn more languages. My linguistic background has made me a language learning enthusiast with a penchant for cross-linguistic observations, and maturity only brought more concern. Each grammar lesson came up as new challenge and internal struggle on *how* and *when* each new grammatical construction should be used. Grammar books and direct questions to teachers were often enough to satisfy my curiosity. In other cases, I used to do what most language learners do and simply occasionally tried to understand the constructions when I happened to see them in new contexts.

The studies I pursued provided me with awareness that resources such as annotated corpora can help me to answer my questions. Moreover, I had the chance to learn how to search for them, including using complex queries, and with the distance necessary to use them properly as I was trained to be critical with regard to the protocol of constitution and annotation of corpora.

Prompting language learners (and teachers) to use corpus exploration tools, as linguists do, is probably the best solution to allow them to be autonomous in their search. However, our hypothesis is that simplyfying the method of corpus exploration for the search of syntactic construction might be more beneficial to them, as they could focus their energy on language data instead.

This background section is meant to provide personal insights on my choices regarding this dissertation. In the following sections, I offer practical reasons for focusing on grammar as well as for why I switched from working on French to working on Korean – a language that I was highly eager to learn when I entered university. Indeed, I chose to attend Korean classes at another university (INALCO, briefly described below) while my own university offered classes in English, Spanish, Portuguese, Hungarian or Finnish[4] to name a few. This decision required me to have lunch in the metro and to run from Mairie de Clichy to Censier several times a week, in order to attend more Korean language classes than I could validate, so that I could keep up with the level of my classmates, whose schedule was fully dedicated to the study of the Korean language, literature and civilisation.

graph in Section 2.2.1.

[4] I also attended Finnish classes for two years as an auditor, thanks to the kindness of the Finnish lecturer and a fortunate coincidence with my schedule.

1.2 Focus on Grammar

All languages in the world have grammar. While words give shape to our world and substance to language, *grammar* is what makes languages more than just an arbitrary succession of words with no relations. Words gather in clauses or phrases, and phrases form utterances or gather in sentences that, in turn, gather in paragraphs and wider units. *Syntax* is the linguistic discipline that specifically accounts for these hierarchical relations, as well as precedence relations, commonly called word order. Given that syntax is a subset of grammar, we alternatively use "syntactic construction" and "grammatical construction" in this dissertation with no particular distinction. However, we may refer to different types of grammar, defined below.

What is grammar? For language learning enthusiasts, grammar is a source of endless means of expressing oneself, but also a dive into the intricate mechanisms of language. However, this is certainly not how grammar lessons are quite remembered by most people. Quite the contrary; Joan Bybee hints at a strong negative experience when she mentions that grammar has a "bad reputation among those who struggled with it in school" [Bybee, 2012].

Perhaps part of the reasons underlying this "bad reputation" is that a flaw in vocabulary is often interpreted as a simple weakness of the memory, either as something that we do not *recall* or something that we do not *know* (yet). Conversely, an error involving grammar is rather perceived as a true deficiency, as due to an incapacity to *understand* the use of a grammatical construction. Indeed, Carton [1995] states that whereas comprehension skills (listening and reading) depend more on the lexicon, grammar is fundamental for production skills (speaking and writting). Forgetting a grammar point or using it in the wrong context therefore entails frustration.

The "bad reputation" of grammar at school is also certainly linked to its prescriptive nature. The following excerpt from Marcellesi [1976, p.9] shows the two sides of grammar that we presented:

> "[...] s'agit-il d'enseigner la grammaire uniquement pour apprendre l'orthographe à l'enfant, pour lui apprendre à "bien" écrire, et sub-

> sidiairement à "bien" parler, ou pour le doter d'un instrument qu'il aura appris à faire fonctionner, qui lui permettra de s'exprimer en toutes occasions, en toutes situations, instrument de libération pour un individu inséré dans les luttes qui, dans notre société, opposent les classes entre elles."
> ("Is teaching grammar only about teaching spelling to children? To teach them to write "well", and subsidiarily to speak "well", or is it to provide them with a device to operate? A device which will allow them to communicate on all occasions, in all situations, a freeing device for an individual integrated in the struggles that oppose classes in our society.")

Contrary to a *descriptive* grammar, whose aim is to describe language structures and patterns of language use, *prescriptive* grammar (also called *normative* grammar) supports the (implicitly unique) proper use of language. Prescriptive grammar is based on a set of explicit rules, which are used as a common reference, a standard, a norm for all speakers of a given language. As its name suggests, from a prescriptive grammar perspective, all deviations from the established norm are considered as errors that should be corrected. The role of prescriptive grammar is to determine what *should be said* and what *must not*.

Incidentally, the norm used in prescriptive grammar is based on restricted samples of *written* productions, but its scope is wider than the genres that it originates from, i.e., either literature or newspapers.[5] As mentioned in the previous quotation from Christiane Marcellesi, prescriptive grammar equally rules written productions and spoken productions.[6]

Written corpora are also composed of books and articles from newspapers or magazines, but more diverse materials are being integrated: for instance, the written corpus of the British National Corpus is composed of published materials (books and periodicals), as well as non-published reports, correspondence and work, all of which were written for different audiences (mostly for adults but also

[5] That is the case of *Le Bon Usage*, "The Good Usage", a famous prescriptive grammar book for the French language.

[6] In this work, we hardly refer to a "grammar of speech" but we do believe that studies of spoken corpora are essential to draw a grammar specific to speech that is not just a deficient version of the grammar of the written language [Brazil, 1995]. As a matter of fact, we also performed some experiments on spoken corpora in Chapter 6.

children and teenagers).[7] Working on a limited number of genres is *not* a problem intrinsically, provided that users of these resources are aware of this limit.

In addition, the aim of the use of corpora is resolutely descriptive, and can be considered as *performance* grammar, as opposed to *competence* grammar taught in school. This dichotomy is borrowed from Noam Chomsky: competence is the knowledge that speakers have regarding their language, while performance is the actual usage of that competence. Competence is known to be greater than performance, since we do not make use of the entire knowledge we have and we do not produce every word that we know. Likewise, we may know grammatical rules, but we may not apply them for fear of making a mistake, or simply because we did not find a proper occasion to do so. Rules of prescriptive grammar thus fall within the realm of the competence of learners, while corpora are, by nature, a showcase for performance grammar.

How should grammar be learned? In his *Traité de stylistique française*, Charles Bally, one of the disciples of Ferdinand de Saussure, has written about the teaching of grammar:

> " Il faudrait substituer à la routine un esprit scientifique sans pédanterie, mis à la portée des jeunes: si on les habituait à beaucoup observer, à réfléchir sans parti pris sur les observations faites, puis à *décrire* au lieu de *généraliser* ou avant de généraliser, ils ne jureraient pas si volontiers par des règles toutes faites et incontrôlées."[8][Bally, 1921, p.27]
> ("We should substitute this routine [of using empirical rules to assimilate] for a scientific approach lacking in pedantry, that is accessible to young people. If we accustom them to observe as much as they can, to think over their observations without prejudice, and also to describe instead of generalising – or before generalising – then, they would not swear so readily by ready-made and uncontrolled rules.")

According to this excerpt, Bally goes a step further away from normative grammar. For him, grammar should not only be descriptive rather than normative, it

[7] http://www.natcorp.ox.ac.uk/docs/URG/BNCdes.html#body.1_div.1_div.4_div.1, retrieved on 16th August 2017.

[8] Italics are from the original text.

should not even be taught as such at all. Instead of predefined rules, grammar should be the fruit of *observations* made by learners themselves. In this view, the learner has therefore an *active role* in constructing their *own* knowledge through observations.

Likewise, Bybee [2006] advocates a usage-based view of grammar, which is also based on observations, which she calls *experience*, and adds a special attention to frequency:

> "A usage-based view takes grammar to be the cognitive organization of one's experience with language. Aspects of that experience, for instance, the frequency of use of certain constructions or particular instances of constructions, have an impact on representation that is evidenced in speaker knowledge of conventionalized phrases and in language variation and change."[9]

Joan Bybee does not explicitly refer to language learning, but while her view of grammar does focus on the speaker, our view of grammar teaching is focused on the learner. This advocacy of learners' active role in their own learning and the importance of exposure to language is compatible with Tim John's Data-Driven Learning approach, described in Chapter 2. However, this approach raises a question: to what extent and how are attested examples of a given grammatical construction (for example from corpora) accessible to language learners?

We will see that while current corpus exploration software applications are powerful tools in providing learners with examples of the usage of particular words, sequences of words, or even certain grammatical constructions, their search options for retrieving grammatical constructions using patterns are often limited or demand specific knowledge.

1.3 Application to Korean as a Foreign Language

Our system was initially designed to be applied to French as a Foreign Language (*Français Langue Étrangère* or FLE), for obvious reasons of localisation, funding

[9] Incidentally, in this dissertation, we use the expression "grammatical construction" but not in the sense understood in Construction Grammar. We may therefore use alternately both "grammatical (or syntactic) construction" and "grammatical (or syntactic) structure".

and linguistic facility. However, after a year working on side projects relating to *sinogrammic languages*[10], we chose to apply our system to Korean. Of course, this decision stemmed from personal interest for the Korean language, but that was not our only reason. It also represents a stimulating challenge for us, and at a favourable time, since Korean has lately received a growing interest internationally, and notably in France.

Inalco is a French institute for oriental languages and civilisations located in Paris.[11] Korean has been taught at Inalco since 1960 but had the least populated department among sinogrammic languages[12] a decade ago, as shown in Table 1.1: in 1996, only 144 students were enrolled in Korean studies (regardless of their grade).[13] However, figures from this table also show that despite a general decrease in the number of students in sinogrammic languages departments, the Korean department is the only one that seems to have gained interest. Indeed, the number of students studying Korean underwent a fivefold increase between 1996 and 2013, whereas the departments of Chinese, Japanese and Vietnamese have all seen their figures decrease gradually. Incidentally, after 2013, a fixed *numerus clausus* has been established in both Inalco and Université Paris Diderot (the only other university in France that delivers a national diploma in Korean language and civilisation studies).

In addition to those figures, we note that the number of language classes offered at the university has increased significantly. Besides Inalco and Université Paris Diderot, which both offer a diploma Korean language, literature and civil-

[10] Magistry et al. [2017, p.40] define sinogrammic languages as languages that share the same writing system as Mandarin Chinese, as well as an important part of their lexicon.

[11] Inalco stands for **I**nstitut **Na**tional des **L**angues et **C**ivilisations **O**rientales, and might be considered as the Parisian counterpart of the London-based SOAS, School of Oriental and African Studies.

[12] 漢字 – *kanji* in Japanese and *hanja* in Korean – are still very much used in their respective countries (although sino-korean words are commonly written in *hankul* nowadays). Sinograms have physically disappeared almost completely from the Vietnamese environment, but a thousand years of Chinese rule has left traces. Compare the different transcribed readings of the sinogram 方 'square': fang¬ (Mandarin Chinese), hō (Japanese), pang (Korean), phương (Vietnamese), and even hong (Taiwanese), fong¬ (Cantonese) and huang (Teochew).

[13] Figures in this table were kindly extracted from the administrative system APOGEE by Stéphane Faucher, head of the board of studies ("*direction des formations*") of Inalco, and brought to us by Yoann Goudin.

Year	Chinese	Korean	Japanese	Vietnamese	Total
1996	1551	144	1767	306	3768
1997	1701	128	1827	268	3924
1998	1597	112	1837	286	3832
1999	1579	111	2037	280	4007
2000	1709	105	1833	267	3914
2001	1559	46	1508	166	3279
2002	1743	52	1611	144	3550
2003	1806	71	1729	143	3749
2004	1618	119	1443	170	3350
2005	1590	172	1484	161	3407
2006	1431	190	1432	141	3194
2007	1217	152	1230	111	2710
2008	1321	211	1457	127	3116
2009	1374	275	1468	118	3235
2010	1088	321	1420	108	2937
2011	1094	529	1427	125	3175
2012	1222	601	1417	152	3392
2013	1244	674	1381	119	3418

Table 1.1: Number of students enrolled in sinogrammic language departments at Inalco

isation[14], we found two other universities that offer diplomas in applied foreign languages,[15] including both English and Korean (Université Jean Moulin in Lyon, and Université de La Rochelle), as well as five universities offering a state diploma in Korean language (Université Michel de Montaigne in Bordeaux, Université du Havre, Université de Nantes, Université de Provence in Aix-Marseille, and Université de Rouen) and one university that offers Korean language classes (Université de Technologie de Belfort Montbéliard).

Working on Korean is challenging not only because it is not my first language, but also because of its properties. Korean is an agglutinative language, which means that words in Korean are composed of multiple morphemes agglutinated together. In fact, as explained in Section A.2.2, teaching Korean grammar essen-

[14] In French, ***L**angues, **L**ittératures, **C**ivilisations, **E**trangères et **R**égionales*, commonly called LLCER.

[15] ***L**angues, **E**trangères **A**ppliquées*, or LEA.

tially means teaching to segment, identify and combine those morphemes.

From this observation, we may assume that it is easy to retrieve syntactically similar constructions by simply concordancing on the right morpheme(s). For example, the morpheme *-keyss-* -겠- is non-ambiguous because it has no homograph, and is used either as the presumptive suffix (which can be glossed as 'may') or the intentional modal suffix ('intend to', 'will').[16] In other words, using simple "겠" as a query in a concordance would allow all sentences containing *-keyss-* -겠- to be retrieved and provide the user with concrete examples of usage of the presumptive or the intentional modal suffix in Korean.

However, seeking Korean grammatical morphemes is not always as easy. The construction illustrated in Example 1 is commonly referred to as *-lcito moluta* -(으)ㄹ지도 모르다 and is used to indicate the speaker's strong uncertainty and is composed of the prospective suffix *-(u)l* -(으)ㄹ, the indirect question noun *-ci* -지, and the verb *moluta* 모르다 'not know' or 'ignore' [Sohn, 2013, p.350].

The first difficulty might seem trivial, but typing the full form of the construction (as shown above) in a concordancer will not match anything. When used on a verb stem ending in a vowel, the prospective suffix takes the form *-l* -ㄹ and is *directly* attached to the verb. Korean is written with an alphabet called *hankul* 한글 in blocks of syllables.[17] This means that while it is easy to isolate the suffix in the transliteration of Example 1 using the Latin alphabet (kule-**l**), it is *not* possible to isolate the letter "ㄹ" because it is integrated into the syllable *lel* 렐, which forms a single *character* computationally speaking. One of the possibilities is to type only the construction without the prospective suffix. However, as we shall see in the results of our experiments in Chapter 6, the prospective suffix is not the only morpheme that can be used with *cito moluta* "지도 모르다".

The second difficulty is due to morphological variations. We have seen that the prospective suffix *-(u)l* -(으)ㄹ takes the form *-l* -ㄹ when attached to a verb stem ending in a vowel. As a matter of fact, the suffix is allomorphic and has another form when attached to a verb stem ending with a consonant: *-ul* -을. Contrary to the previous form, this form stands as a full syllable and can therefore be retrieved.

[16] The different usages of the morpheme *-keyss-* -겠- are given in the above-mentioned section, in Examples 20.

[17] More details on *hankul* 한글 are given in the paragraph "Korean Characters (computing)" in Section 5.6.

For example, *mokul* 먹을 'which will eat' or 'to be eaten' can be segmented into the verb stem *mok* 먹 'eat' and prospective suffix *ul* 을. In addition, the verb *moluta* 모르다 underwent two morphophonological changes to become *molla* 몰라 in the example: first, the deletion of the stem's (*molu* 모르) final vowel *u* ㅡ because of the vowel *a* ㅏ of the infinitive suffix; and second, the compensatory doubling of the now final *l* ㄹ. The whole process can be summarised as: 모르 (stem) + 아 (infinitive suffix) = 모ㄹ + 아 = 몰ㄹ + 아 = 몰라. Consequently, in order to retrieve as many sentences as possible while taking into account the morphological variations of such construction, the query has to look like 을? 지도 (모르|몰)[18], which can be glossed as "a construction starting with *ul* 을 or not, followed by a space (or not)[19], *cito* 지도, another space and either *molu* 모르 or *mol* 몰".

The last but not least difficulty concerns the possibility of searching for non-contiguous morphemes. This is not the case for this construction, but other constructions, such as A*myen* A*swulok* B (A면 A(으)ㄹ수록 B) 'the more A, the more B', necessarily involve the verb A between the two suffixes because each is attached to a verb. As in the previous problem, this difficulty can be solved using a regular expression, such as 면 .*?수록, but the construction of this type of query is not within the average person's reach.

Incidentally, these properties (except for the non-contiguity of morphemes) were used to build Table A.4 as well as to select the grammar points for our experiments.

(1) 그럴 지도 몰라.
kule-l ci-to moll-a
be.like.this-PRS whether-too ignore-INF
'I have no idea whether or not this can happen.' (intimate speech level)

In the present work, we endeavour to solve this research problem by constructing a system that provides access to annotated corpora for language learners (and non-specialists in general) and is precisely what allows more attested examples of a given construction to be sought in those corpora, without prior knowledge in linguistics or on how to use a corpus exploration tool.

[18] This imaginary query is a regular expression, i.e., a pattern that uses a specific formalism and operator symbols used to match a string.
[19] See the note on Korean orthography rules in Section 3.5.2.

1.4 Outline of the Dissertation

Our work, and accordingly, this dissertation, can be considered as a journey through different fields of knowledge, and of practice, as well as of various traditions. The chapter order that we propose only reflects our own peregrination. Readers are therefore free to undertake the journey from their own field, according to their expertise, or satisfy their curiosity by exploring an unfamiliar field first. In other words, it is up to the reader to choose a winding route, a shortcut or safely stay on the straightforward one. Whatever their choice, readers may find useful the frequent cross-references and indexed notions (indicated in the margin) that we set with the aim of facilitating detours.

This dissertation is organised as follows:

The first three chapters following this introduction constitute the state-of-the-art part of our dissertation. Their common objective is to provide the reader with the necessary background from the various disciplines upon which this work is built: language learning, corpus linguistics and natural language processing, with an in-depth focus on the design of corpus exploration tools.

Chapter 2 describes the framework of our research problem and accounts for our proposition of using native corpora in language learning. In order to explain what is at stake in language learning, we start by discussing the definitions of "first language" and "second language" before comparing the role of linguistic input in their respective acquisitions. We then present a selection of initiatives using native corpora for language learning, either indirectly or directly, before focusing on Data-Driven Learning, the approach that inspired our work.

Chapter 3 is an in-depth exploration of the corpus as a linguistic resource: this chapter provides explanations about the reasons why data are collected and assembled into corpora, why some of them have to be preprocessed, what kind of annotations we may find, and how those enrichments are exploited by language specialists. As an illustration, we describe the Korean language reference corpus, also called the Sejong Corpus, which we used in our experiments.

Chapter 4 concludes this state-of-the-art part with a historical and practical

overview of corpus exploration tools. For this overview, we selected various tools with different purposes. Using illustrations of the uses of these tools, we endeavour to identify the wide range of functions and querying possibilities offered by these tools and how suitable or unsuitable they may be for non-specialist users, not only in terms of interface, but also in terms of accessibility of the query language.

The two following chapters present our contribution: the requirement specification of an original corpus exploration function and the preliminary experiments that serve as a proof of concept. Due to the fact that the second is the concrete implementation of the first, these two chapters should be read in their original order.

Chapter 5 is the core of our work. It contains an extensive general description of the whole system architecture that we designed, as well as an illustration of the processing, with an example of what is expected at each step. With regard to the objectives of our work, we account for the use of similarity measures (including edit distance) and show their advantages over strict matching, as in current corpus exploration tools.

Chapter 6 serves as the proof of concept for our system. First, we provide a detailed presentation of the resources that we used (samples from the Sejong Corpus and illustrations of grammar points from Korean language textbooks), as well as a description of the preprocessings that were necessary for our preliminary experiments. Then, we present the various options that were tested and their results compared to our expectations. Finally, we demonstrate that our system is not specific to Korean by showing the adaptation to English data.

Following the tradition, the final part of the dissertation is composed of the conclusions of our current work and a presentation of the perspectives that still await us in our undertaking of retrieving similar syntactic constructions.

Chapter 2

Linguistic Resources in Language Learning

2.1 Introduction

While this work is situated in the realm of Natural Language Processing, every decision was resolutely made considering its final application to language learning.

Learning a foreign language is something that humans have been doing from as far back as since they have needed to understand or to communicate with other people, whether for commercial purposes or to thwart the plans of the enemy in war times. Its systematic study is a much more recent phenomenon in comparison, but has become increasingly important in a more and more globalised world. As Ellis states in his introduction of *Second Language Acquisition*, stakes may have changed but remain crucial:

> "This has been a time of the 'global village' and the 'World Wide Web', when communication between people has expanded way beyond their local speech communities. As never before, people have had to learn a second language, not just as a pleasing pastime, but often as a means of obtaining an education or securing employment. At such a time, there is an obvious need to discover more about how second languages are learned." [Ellis, 1997, p.3]

The study of language acquisition can be historically viewed as a sub-discipline of applied linguistics, but inevitably involves other disciplines: the first that might come to mind is education, given that language acquisition still mostly occurs within the framework of an institution; the second, equally important but with a completely different view, is psychology; in particular, behavioural or cognitive psychology, whose opposite viewpoints are briefly described in 2.2.2. While the first discipline views things from the teacher's perspective, the second accounts for what happens in the learner's mind. We may also mention the acquisition/learning dichotomy and say that education studies are aimed at enhancing *language learning*, while psycholinguistics describes *language acquisition*.

These two fields are, however, not totally independent from each other. Research in language learning takes into account findings from language acquisition, such as the way in which the lexicon is stored in the learner's brain and how it differs if the learner is bilingual, or the stages of cognitive development and their consequences on the order in which certain notions have to be taught, as well as the differences between learners depending on their personality, their learning styles and strategies. Likewise, while our study clearly falls within the frame of language learning, it is rooted in one of the issues that any language acquisition theory has to address: the role of input.

This chapter focuses on the linguistic resources that are available to language learners, in the broadest sense of the term: first, linguistic resources are defined as the *linguistic input* that learners are exposed to in Section 2.2, whereas in Section 2.3, they refer to the *actual material* that learners may use for language learning. The last sections list the range of linguistic resources available in language acquisition and discuss the access to these resources by language learners, eventually focusing on resources for learners of Korean as a Foreign Language (KFL).

2.2 The Need for Linguistic Input in Language Acquisition

All theories, either in First or Second Language Acquisition, agree on the fact that there cannot be any sort of acquisition without linguistic input, i.e., without exposure to 'real language' resulting from an effective interaction with other human beings. Both conditions have to be fulfilled: infants watching videos of a person speaking not directly to them or infants interacting with a person who does not use any language with them (either signed or spoken) will not be able to acquire language, even though they all have this inner capacity. The former case was tested by Kuhl et al. [2003] on phonetic learning, and authors suggest that interpersonal social cues and referential information, such as joint visual attention, is significant for infants.

What differs between theories is the role that they allocate to linguistic input and its importance.

2.2.1 First or Second Language Acquisition?

Before we can address the topic of the role of linguistic input in first language acquisition, we must ask ourselves what a 'first language' (sometimes abbreviated as L1) is, and to what extent this denomination is related to other common expressions with which it is regularly used interchangeably, such as native language or mother tongue. Incidentally, those differences also exist in other languages; for instance, in French they are respectively named *langue première*, *langue natale* and *langue maternelle*, while in Korean, they are called *cey 1 ene* 제 1 언어 (literally 'first language'), *mokwuke* 모국어(母國語, 'motherland language' or 'homeland language') and *moe* 모어(母語, 'mother language').

Order of acquisition The adjective 'first' implies that the order of acquisition of languages is fundamental, and that the acquisition of a second (or third, fourth etc.) language is somehow different. Using this property, Leonard Bloomfield draws a link between a 'first language' and a 'native language' (he also defines native speakers, a concept that we look deeper into in Section 2.2.4) in the following

definition from *Language*:

> "The first language a human being learns to *speak* is his *native language*; he is a *native speaker* of this language".[Bloomfield, 1935, p.43]

native language

native speaker

As stated above, 'native language' is a commonly used expression referring to first language. This definition is a good start with regard to its simplicity, but using the order of acquisition as the sole criterion is not sufficient in some (special but not so rare) cases when two or more languages are acquired simultaneously or nearly. In the case of early bilingualism, if the two parents speak a different language to their child, which one is the first language? Naturally, waiting for the first word that a child raised in a bilingual environment utters to identify its first language is not relevant[1]: it does not mean that the child does not understand the other language, or even that the word uttered is part of a distinct lexicon yet, or instead part of the overlap between vocabularies (which can only be determined with more linguistic data, in particular translation equivalents of the same words [Lanvers, 1999; Pearson et al., 1995] cited by Yip and Matthews [2007]). We would, therefore, rather say that early bilinguals have two first languages (trilinguals have three languages and so on and so forth) which is not the same as being a monolingual native speaker of either of these languages (see discussion on 'native language' in Section 2.2.3).

In order to understand what distinguishes an actual second language from a second first language, we need to look at other criteria: the question of the *critical age* up until which it is possible to acquire a language, how and from whom the transmission proceeds, and what level of proficiency is required.

Age of acquisition Indeed, what is implied in the denomination 'first language' is not just the order of acquisition but more importantly its earliness. It appears

[1] As a matter of fact, language differenciation occurs before infants produce their first words [Yip and Matthews, 2007, p.34] and the mastering of two languages at a 50/50 rate is more of an ideal than a reality even for an early bilingual. Although there is little relevancy in the order of acquisition for early bilingualism, there is undoubtely a dominant language, even at such an early age.

that "there is a period during which language acquisition is easy and complete (i.e., native-speaker ability is achieved) and beyond which it is difficult and typically incomplete" [Ellis, 1997]. These two observations are characteristics of what is called in biology a 'critical period'. This phenomenon thus gave its name to the theory addressing this issue in language acquisition: the Critical Period Hypothesis (henceforth CPH). Singleton and Ryan [2004] give a thorough overview of the CPH and its implications both for first and for second language acquisition, as well as evidence of its existence and duration from various studies of two disciplines:

1. neurology, which sheds light on the loss of some language-related capacities, such as phonological discrimination invoking, in particular, the diminishing plasticity of the brain and its lateralisation, i.e., the specialisation of its areas, including the language areas;

2. language acquisition by children with impairments[2].

Thus, Singleton and Ryan focused on previous studies of language acquisition by deaf children, by feral children (namely, two well-known cases: that of 'Victor the Wild Boy of Aveyron' who lived in the 18th-century and was commonly known as '*Victor l'enfant sauvage*' or '*Victor de l'Aveyron*' in France, and also the case of a 20th-century girl from California best known by her pseudonym 'Genie') and, finally, language acquisition in subjects with Down syndrome, a genetic disorder causing learning disabilities, especially with regard to phonological acquisition.

From this cross-study comparison, Singleton and Ryan conclude that language acquisition is already "in process from birth onwards" but that there is no real consensus on the offset of the critical period. This might be due to the differences in approach and the great number of factors that are at stake in these studies (especially those of the feral children, who in most cases were also victims of severe abuse for years, but might also have not benefited from adequate help in recovering or developping language [McNeil et al., 1984]). They also found that there is "no clear ground that language acquisition cannot occur beyond puberty", which does not make it a 'critical period' in the sense used in the biological sciences. However, although there is no proof of the impossibility of acquiring a language after the

[2]Since experiments aimed at purposefully depriving children of language are absolutely socially and ethically unacceptable.

end of puberty, there is an agreement on difficulties and incompleteness, which validates Ellis' definition.

This is also what Nicolas Tournadre asserts in the introductory chapter of his book *Le Prisme des Langues* aimed at a general public, as he gives the definition of another near-synonym of first language, *mother tongues*, in these terms:

mother tongue

> "Les langues 'maternelles' ne sont pas des langues transmises par la mère, pas plus d'ailleurs que par le père, l'oncle ou la tante, mais sont des langues acquises 'parfaitement'[3] au cours de l'enfance ou de l'adolescence. [...] Elles sont *acquises* sans effort et non apprises selon un processus volontaire et conscient." [Tournadre, 2014, p.16]
> (" 'Mother' tongues are not languages transmitted by the mother, nor are they by the father, the uncle or the aunt, but are languages acquired 'perfectly' throughout childhood or adolescence. [...] They are *acquired* effortlessly and not learned according to a voluntary and conscious process.")

Tournadre does not take a clear stance on this issue and only indicates vague periods ("childhood" and "adolescence") but his definition gives us more interesting criteria for our discussion.

Transmission by whom? The first of these criteria is about who is involved in the transmission of a first language: the denomination itself suggests the mother, but Tournadre defends that this criterion is not relevant. This is in accordance with Bloomfield:

> "A child cries out at birth and would doubtless in any case after a time take to gurgling and babbling, but the particular language he learns is entirely a matter of environment. An infant that gets into a group as a foundling or by adoption, learns the language of the group exactly as does a child of native parentage; as he learns to speak, his language shows no trace of whatever language his parents may have spoken." [Bloomfield, 1935, p.43]

[3] Quotation marks are from the original text.

The adjective 'maternelle' (literally '*maternal*') does not actually refer to a language related to mothers in this case, but to a language related to nurture. What is important in the acquisition of a mother tongue is not the status of people that children are interacting with, but rather the interaction in itself. Whether it be with members of the biological family or not, children develop some kind of affection for their mother tongue(s), the language(s) of the people who nurtured them.

Language proficiency Secondly, we note that Tournadre expects native speakers to have acquired their mother tongue(s) not so 'perfectly', as he uses quotation marks. The word *perfection* may not have much sense with regard to language mastery.

Tournadre also specifies in a footnote that even 'true bilinguals' seldom have the same competence in both languages. In a 'global village' context, there are incidentally cases where native speakers may not even be considered as good speakers of their own mother tongue(s), either because they gradually lost their language abilities by not speaking their mother tongue(s) regularly or because they only speak their mother tongue(s) in certain contexts.

Typically, the former case illustrates multilingual societies, such as some areas of Kabylie, a region of northern Algeria. While the Kabyle speak a variety of Berber called Kabyle, Literary Arabic is used in teaching and administrative contexts. Furthermore, French is actually the dominant language, especially for the middle class, as the consequence of its predominance in the media (both in newspapers and television) and in a business context, as well as in formal situations [Chaker, 2004, p. 4057]. This leads the Kabyle people to be able to *speak* their native language to some extent, but not to write it.

This is also what happens to immigrants who choose to communicate exclusively in their "adopted language" at the expense of their first language for social integration purposes. This phenomenon is what Bloomfield identifies as a "shift of language". This loss of the first language in an environment where the second language is spoken is called language attrition.[4] Likewise, children of immigrants

[4] Seliger [1996, p.616] precisely defines language attrition as "the temporary or permanent loss of language ability as reflected in a speaker's performance or in his or her inability to make

might see the same shift and forget the language that they inherited from their family, and solely speak their "adult language"[5], i.e., the language of the country that they live in.

On the other hand, the second case describes the situation of children of immigrants who only speak their first language at home (or at least within the family circle) but outside this setting, on any other occasion, and therefore most of the time, they speak another language. Even though this language obviously comes second, perhaps even several years after the first exposure to the heritage language, it is also to be considered as their (second) native language. Interestingly, the second native language would soon become the language in which those children are the most fluent in, as a result of socialisation and school. In some cases, this asymmetrical relation may be even stronger. Indeed, it is also not rare that those children lose the ability to speak their heritage language fluently (yet not the ability to understand it), especially if they have an older sibling who already goes to school and who has brought home the language of the country that they live in. This phenomenon is commonly observed nowadays among the first generation of children born in the country of immigration (also called 'second generation', the 'first generation' being the one that immigrated), such as the Teochew community in France, in which I grew up.

In my case, as the eldest of my siblings, born in France and raised by several members of my family with variable proficiency in French who spoke to me in Teochew (潮州話 - a southern Min language) only before I attended preschool, I was indirectly exposed to input in French from birth but started interacting in French only from the age of three. Teochew is obviously my first language, but I consider both Teochew and French to be my native languages because from as far back as I can remember I could think in both languages and I do not recall having any trouble in acquiring French. However, I am aware that a transfer from Teochew to French does happen (and vice versa) and that there are apparently

grammaticality judgments that would be consistent with native speaker (NS) monolinguals of the same age and stage of language development."

[5]We believe that Bloomfield described here the case of children who emigrated along with their parents, since for the generations of children born in the country where their parents immigrated, there is no reason for this language to be linked to their adulthood as they must have learned it at least since they were sent to school.

well-known expressions that I do not understand, typically taken from French old slang, which is mainly transmitted to French children by their grandparents or great-grandparents. For obvious reasons, my cultural heritage is different. I do believe, however, that I also know French expressions that other natives do not know and that, as a matter of fact, it is virtually impossible for a native speaker of a language to have a sound knowledge of all varieties of that language.

Finally, Bloomfield also notes that there are also extreme cases where "[a foreign-language learner] becomes so proficient as to be indistinguishable from the native speakers round him", showing that language proficiency is definitely not a criterion defining a first or native language but is rather a common property, at least for monolinguals.

Nature of the Process Lastly and most importantly, the main difference between native language(s) and the other languages that one speaks is highlighted in the last part of Tournadre's quotation: it is precisely the very nature of the process of acquisition that makes it unique. Tournadre puts into perspective acquisition with learning, stating that what is 'acquired' (in italics in the original text) is *not* 'learned' (emphasis added this time). He also chose to write this definition at the beginning of a section entitled "*langue acquise versus langue apprise*" (acquired language *versus* learned language). This brings us back to the dichotomy mentioned in the introduction to this section, when we opposed psycholinguistics to education. We remarked that the latter is automatically linked to the particular context of language learning and teaching, in other words, to the framework of an institution, with a teacher as the main 'deliverer' of language and the classroom as the setting of 'delivery' or transmission of knowledge.

Moreover, according to Tournadre, learning a language is a process that is necessarily conscious, voluntary, and, by opposition to acquiring a language, learning costs a conscious effort. Indeed, students attending language classes know why they are seated in a classroom and listening to the teacher, taking notes, doing exercises, being evaluated, trying to memorise words and perhaps struggling in doing so, while we can picture children in preschool (interestingly also called 'nursery school' or '*école maternelle*' in French) playing and interacting with other children or adults, actually receiving linguistic input (caretaker talk) and feedback on what

they are saying, but never acting like they are conscious that they are learning a language.

This dichotomy is one of the five main hypotheses of the language learning model in Stephen Krashen's major work, summarised in Table 2.1. According to this table, the acquisition of a language is initially prompted by the will to communicate, while language learning may be due to various goals. Indeed, the motivation behind learning a language could be due to very practical reasons, such as getting a diploma or a job, or being socially integrated in the case of immigrants for example, but it could also be due to a keen interest in the language or culture.

	Acquisition	**Learning**
Process	subconscious	conscious
	implicit	explicit
	grammatical 'feel'	grammatical 'rules'
Situation	informal	formal
Perception	natural	artificial
	personal	technical
Language Exposure	massive	limited
Base	practice	theory
	language in use	language analysis
Method	inductive coaching	deductive teaching
	rule discovery	rule-driven,
	bottom-up	top-down
Goal	communication	various

Table 2.1: Selection of properties opposing the processes of acquisition and learning from Krashen [1981a]

Another important feature is however missing in this account: native speakers of a language have a unique affective attachment towards their native language(s) and might consider them as part of their own identity. Qualifying the idea that the acquisition of a first or a second language is mostly similar, Wolfgang Klein gives as his first argument that native languages constitute one aspect of the cognitive

and social development whereas this development is supposed to be finished when one is learning foreign languages:

> "L'ALM [Aquisition Langue Maternelle] et l'ALE [Acquisition Langue Étrangère] se distinguent entre autres par le fait que la première constitue un aspect du développement cognitif et social global, alors que dans la seconde, ce développement est achevé (ou presque)." [Klein, 1989, p.39]

To conclude this discussion we would like to mention the extreme case of adults who are Native Koreans born in Korea and adopted in France by French families between the ages of 3 and 8 and who have not been exposed to Korean since adoption. Pallier et al. [2003]'s neuroimaging study shows that the eight individuals do not perform differently from those of the control group of native French speakers on given tasks, despite Korean being their 'native language'. This conclusion suggests that they could not benefit from the exposure that they had in infancy or childhood because for some reasons Korean was seemingly "erased" from their brain. Moreover, in Pallier [2007], Christophe Pallier gives more precision about this group of adoptees. He adds that further experiments were conducted on their level of proficiency in French and that those experiments demonstrate that, again, their performance on given tasks was no different from that of French natives, but different from that of Korean natives who learned French as a second language and have lived in France for several years. This confirms his previous hypothesis that:

> "Any child, if placed in the unusual situation of having to learn a new language between 3 and 8 years of life[6], can succeed to a high degree, and that they do so using the same brain areas as are recruited for first-language acquisition" [Pallier et al., 2003, p.158]

From this conclusion and our discussion, we would say that for those adoptees, Korean is indeed their first native language, but instead of referring to French with

[6]We note that this neuroimaging study might give more precision on the period of the CPH, but having worked with subjects who were adopted before the age of 10, Pallier et al. only shows that language acquisition is still possible before 10 but not that it is impossible after 10.

the 'L2' acronym (which stands for 'second language') used by Pallier et al. we would rather say that French is their 'second native language' as we did for children of immigrants: they *acquired* it at the *early age* and there is little doubt about the fact that the cognitive and social identity of the adoptees was still developping when they were adopted.

2.2.2 Input in First Language Acquisition

The first linguistic resource that we are considering in this work is linguistic *input*. The word 'input' is commonly used in Natural Language Processing to refer to the data given to a programme to be processed in a processing chain. The data that results from this process is then called an 'output' by opposition. In this chapter on language acquisition, what we call 'input' is the linguistic data to which acquirers or learners are exposed. For children acquiring their first language, it mainly consists of oral samples of language that are available to them when they interact with adults. For language learners, linguistic input usually consists of both oral and written samples of languages but the nature of these samples is very different from what is found in first language acquisition (see Section 2.2.3 for the role of input in Second Language Acquisition). In both cases, the role of input is undeniably crucial. Incidentally, one of the conclusions of the studies of feral children that we mentioned when we questioned the notion of 'critical period' is that language acquisition cannot occur without human interaction using language at an early age, in other words, the lack of linguistic input given through meaningful interaction.

One of the obligations of theories of language to be viable is to account for the nature of language, as well as its development and acquisition. What is particularly interesting for us is to understand how each of them integrated input into their model.

Heike Behrens has worked on the relation between input and output in first language acquisition to understand to what extent an input language gives concrete evidence of language and to what extent children's language relates to the input language. As we have seen in Section 2.2.1, children acquiring their first language(s) are actually not aware of any process happening in their mind and seem

to effortlessly manage to not only pick up phonemes in their native language(s) among all of the sounds that they are able to discriminate but also to infer implicit grammatical rules and other subtleties of language by interacting informally with adults. In order to explain the apparent 'miracle' of the "the acquisition of a highly complex language very fast and seemingly without effort", Behrens invokes and opposes major theories on first language acquisition:

> "In the nativist tradition it is assumed that innate linguistic representations, Universal Grammar, help children to identify and acquire the linguistic rules which are relevant in their target language [...]. In constructivist and emergentist approaches, no specifically linguistic innate representations are assumed. Instead, it is argued that children are very efficient pattern and intention recognisers so that they can induce linguistic structure based on the language they hear." [Behrens, 2006, p.3]

What is important to understand is that in one case, children already have innate properties of language *encoded in their brain*, while in the second case children do have innate capabilities related to language but really have to *induce language properties from the input* that they are receiving. These opposite viewpoints are still competing in the nature versus nurture debate in their modern form and it would take much more than a feeble subsection to account for it. We are therefore only briefly introducing the part of language acquisition theories that focuses on what is relevant for our purposes: the role of input.

Behaviourism One of the first schools of thought that tried to explain the role of linguistic input in language acquisition is the behaviourist theory of verbal behaviour (an extension of Skinner's general theory of learning) based on 'operant conditioning'. For this theory, input is a set of empirical stimuli to which children respond by emitting responses or 'operants' (i.e., a sentence or utterance). Stimuli are not necessarily observable but they are essential to trigger reactions, which implies that the control of stimuli is important to enable the acquisition of language as a system:

> "A child acquires verbal behavior when relatively unpatterned vocalizations, selectively reinforced, gradually assume forms which produce appropriate consequences in a given verbal community." [Skinner, 1957, p.31]

As this quotation explicitly says, operant conditioning is based on reinforcement, which is called positive "if a desirable event or stimulus is presented as a consequence of a behavior and the behavior increases", or negative "when the rate of a behavior increases because an aversive event or stimulus is removed or prevented from happening" [Flora, 2004]. Both entail an increase in the frequency of a behaviour but differently: for example, a child asking for water politely is said to have received positive reinforcement if a compliment along with the water is given as a reward for using a polite question, while the same event is said to be negatively reinforced if the child does so to escape being scolded by adults.

Behaviourist theories are best known for their research on animals more than on human beings, the most popular case being the experiment Pavlov, conducted on a dog to which he successfully taught to salivate (the operant) at the sound of a bell (the stimuli) previously combined with food several times. The conclusion of this experiment is that the dog has learned to automatically associate the bell sound to food.

Today, behaviourism remains a fundamental school of thought with major contributions in psychology and interesting methods based strictly on empirical observations, but this rigourous methodology and dedication to the directly observable is not considered sufficient for language acquisition, as it does not account for all of the cases in which stimuli are not enough to trigger a good response and does not take into account important factors, such as the developmental stage of children.

Nativism Nativists also see linguistic input as a trigger, but instead of triggering an operant (as in the behaviourist view) from children, input activates innate properties of language according to the properties found in the input. One of the most important notions from nativism, which is also mentioned by Behrens, is "Universal Grammar", implying that the properties encoded in every child's brain are of the same nature, and work with any existing natural language. What is

invariable is called a 'principle' whereas what varies across languages is a 'parameter' [Piattelli-Palmarini, 1980]. Brown [2006] gives the example of the principle of assigning meaning to word order. Which parameter applies depends on the specific language in question: if children are exposed to subject-object-verb input for instance, they will activate this parameter and inhibit the subjet-verb-object parameter and vice-versa. Brown also mentions another fundamental notion in generative language acquisition theories, what Chomsky [1965] presents as the language acquisition device (often abbreviated as LAD), a metaphorical 'little black box' in the brain that embodies this innate knowledge.

We also note that among nativist theories, some do take into account the developmental stage of the child. For instance, Krashen's Input Hypothesis stipulates that "if an acquirer is at stage or level i, the input that he or she understands should contain $i + 1$" [Krashen, 1981a, p.100].

Constructivism The most recent school of thought among the three presented here, constructivism, benefitted from advances in psychology in building their model. For constructivists, linguistic input is not used as a trigger. Still according to Brown [2006], the emphasis is rather on "[the construction of] meaning out of available linguistic input [...] in creating a new linguistic system" and precisely on "the importance of individual learners constructing their own representation of reality" for cognitive constructivists, and on "the importance of social interaction and cooperative learning in constructing both cognitive and emotional images of reality" for social constructivists. Providing any linguistic input is not enough for language acquisition: if children are not able to comprehend the input that they are given, then they will not construct any meaning out of it.

On the one hand, one of the most essential concepts in social constructivism is Lev Vygotsky's *Zone of Proximal Development* (ZPD), a concept that Brown defines as "the distance between learners' existing developmental state and their potential development [or the description of] tasks that a learner has not yet learned but is capable of learning with appropriate stimuli." That "appropriate stimuli" is the *comprehensible input* children need to accomplish a task that "[they] cannot yet do alone but could do with the assistance of more competent peers or adults" [Slavin, 2005, p.44] cited by Brown. It has to be comprehensible but also just the

amount of help that the children need and not more to truly "allow [them] to take on increasing responsibility as soon as she or he is able"[Rosenshine and Meister, 1992].

On the other hand, Jean Piaget's works on cognitive constructivism insist on learners building on prior learning experiences and therefore stress on "the importance of individual cognitive development as a relatively solitary act" [Brown, 2006, p.13]. This belief is based on Jean Piaget's biological timetables and cognitive stages of development. For him, social interaction is less central but still "triggers development at the right moment in time".

2.2.3 Input in Second Language Acquisition (SLA)

Now all of those theories give their own account of how first language acquisition occurs according to their model, but the focus of this chapter is the role of input in language *learning*. For the reasons stated above, learning a (second or foreign) language cannot be the same as acquiring a first language, at least because it means that learners already 'know' one language, and that their cognitive and social development is achieved (or nearly, depending on the age). However, it does not mean that learning a second language is in any way easier than acquiring the first (it is, as a matter of fact, rather felt as much more difficult and 'unnatural' (see Table 2.1 in Section 2.2.1), but it is obviously different. Yet studies in language acquisition have implications for language teaching and have sometimes even directly inspired teaching methods for pedagogical innovations.

Nativist theories, such as Krashen's Input Hypothesis, suggest that exposure to the language is sufficient to acquire it, instruction is unnecessary. Furthermore, in his own words, Krashen [1981b, p.62] asserts that "comprehensible input is the only causative variable in second language acquisition.". This quotation is cited by Brown, who reproaches Krashen for putting too much responsibility of acquisition on the input and for leaving the learner "at the mercy of the input that others offer". Instead, we can consider that the learner has a more active role to play in constructing meaning out of the input that he or she is given, or in other words, in transforming input into *intake*, which we can define as the part of

intake

saliency

input that the learner actually takes from, remembers and learns from. Similarly, instruction might also help the learner noticing some input by giving *saliency* to some elements. We constantly receive linguistic input and it is difficult to contrast data and to discriminate what is relevant, i.e., what we do not know but could learn from the context, especially while communicating because the communication is more focused on intercomprehension.

Constructivism has inspired Community Language Learning, as well as the Silent Way, two methods advocating the central role of the learners' construction of language.

2.2.4 Target Language Data in Second Language Learning

Most of the time in second language learning, learners are exposed to what we call *simple codes*, that is to say, the adaptation of one's level of language to that of the interlocutor who is assumed to be at a lower level by giving *comprehensible input*. This phenomenon is not restrained to second language learning contexts. We previously mentioned it for first language acquisition as it occurs with caretaker talk for instance.

Krashen [1981a] lists three types of simple codes in second language learning:

1. *teacher-talk*, "the classroom language that accompanies exercises, the language of explanations in second language and in some foreign language classrooms, and the language of classroom management";
2. *foreigner talk* occurring outside the classroom and with native speakers;
3. and finally *interlanguage talk*, "the speech of other second language acquirers, often that of the foreign student peer group".

The most representative of simple codes is *teacher-talk* and for some learners, teacher-talk is even the only exposure that they have to the target language if the teacher does not make sure that learners interact with each other in the classroom. On the other hand, learners who are acquainted with native speakers may be exposed to *foreigner talk*, which happens, for instance, when friends who are aware of

the difficulties of the learner try to make sure that they communicate successfully.

In the 'global village' context, a learner of a language has also access to unsimplified authentic samples in the target language. These data have not been altered to be comprehensible, and are usually uttered by native speakers for other native speakers.

Authentic language? Authentic language materials are opposed to non-authentic materials, such as the written dialogues (or their audio counterparts) found at the beginning of each lesson of most of modern textbooks. Those are usually made up by teachers and researchers in second language acquisition to illustrate a *speech act* and to teach a communicative competence by giving appropriate language samples (in terms of vocabulary and grammatical constructions) for a given situation (typically, how to order at a restaurant or how to borrow a book at the library). In class, the analysis of these dialogues would typically be preceded by a discussion on how learners actually act in a similar situation, and followed by another discussion on more extra-linguistic and pragmatic related issues such as the appropriate gesture to call a waiter, the appropriate posture when addressing him, and the differences there might be if it is a waitress, or what to do in the event that we cannot find our library card but need to borrow books. This scheme is designed to anticipate learners' needs and corresponds to the *communicative approach*, which is the dominant approach nowadays to language teaching and learning.

speech act

communicative approach

The data described here is purposely not authentic but gives the saliency that we briefly mentioned in Section 2.2.3 to the features that are important for the speech act. In this sense, non-authentic data are complementary to authentic data and our tool – which is aimed at providing more authentic samples based on syntactic similarity – might as well be used along with non-authentic materials and explicit instruction.

Another type of authentic data has always been available for learners but also benefitted from globalisation and the development of peer-to-peer networking or online platforms such as YouTube: books, films, series and other cultural media. There are two main reasons why all of them are more easily found in their original

version today: first, more people have learned foreign languages and the mastery of a foreign language is not reserved to an elite anymore, and second, human immigration has created new needs. When a community of immigrants settles in a city, it is not rare to see specialised bookstores or CD/film stores opening in this area and it is not surprising that a cosmopolitan capital like Paris has this type of stores for each of its minority communities. While books in a language other than the official language(s) of a country are still grouped together in specialised libraries or shelves, in the case of films, national release in their original version is widespread and not just for films in English. In some cases where a film is only released in its subtitled original version, we might think that this choice has been made for econominal reasons, but in the cases where the film is released in both subbed original version and dubbed, the cost is more important. This new policy shows that it is assumed that people are increasingly interested in original versions. We can also note that in the case of English, a more drastic shift has been made in some countries (Scandinavian countries in Europe for instance), where films with English as their original language are not even subbed, although English is not one of the official languages, nor a national language.

These data are indeed not suitable for language learners but are still useful in the acquisition of the target language. Most teachers even advise their students to watch films or series in their original versions (subtitled if needed, and preferably in the original language) instead of a dubbed version, in order for them to train their ears to the sounds of the target language but probably also to notice salient input and learn from it, and maybe even to start forgetting that this is their target language, and just leisurely enjoy exposure to it. The strenght of this sort of data is that learning is *not* their purpose. Ours is different due to the fact that in our case, we intend to provide data that answers questions from the learner who is *conscious* of the process of learning and is even *active* in it.

Native language? There is an interesting discussion in SLA that we cannot not mention in this dissertation: the discussion around the native speaker as a model for L2 learners. According to Vivian Cook's works, the native speaker is the implicit model of most theories in language teaching and SLA, a "ghost-like presence" to which we are always ultimately comparing the L2 user. Cook distinguishes the L2

learner who is still in the process of learning the L2 from the L2 *user*, which refers to someone who is *using* the L2. There is no definite boundary between the two and it seems that a learner becomes a user whenever he steps out of the classroom and uses his L2, but this terminology has the merit of acknowledging L2 users as actual speakers of the L2 in their own right.

For Cook [1999], a native speaker is not the ideal speaker that Chomsky describes and has no undisputable characteristics but two : (1) "a person is a native speaker of a language learnt first" and (2) "native speakers are not necessarily aware of their knowledge in a formal sense", a property that Cook compares with riding a bicycle and not being able to explain how. These properties are in accordance with those mentioned in our own analysis of a native language in Section 2.2.2; the first is related to the earliness of acquisition, and the second to the very nature of acquisition. Given that being *multicompetent* (in the sense of being a user of several languages) has consequences on how the brain functions and for other obvious reasons, L2 users will never be native speakers unless they were born again, thus making the monolingual native speaker[7] as a model an unattainable goal.

Instead, Cook argues that in language teaching, the model should be the L2 user. One of her arguments is that the knowledge that the L2 user has in mind as a speaker has its own characteristics (see Selinker [1972] on the well-known notion of *interlanguage*, which we are not exploiting in this work).

> "In a sense, whatever the native speaker does is right—subject, of course, to the vagaries of performance and the like. Multicompetence is intended to be a similarly neutral term for the knowledge of more than one language, free from evaluation against an outside standard." [Cook, 1999, p.190]

Since our work is aimed at providing authentic data from native speakers, we are aware that our work might be perceived as a suggestion that the native speaker is the ultimate model that learners should copy to be 'native-like'. Rather than this, we believe that providing native speaker data that were *not* specifically collected for language learning nor selected for being conventional (data used by

[7] Cook also distinguishes the *monolingual* native speaker from the *bilingual* native speakers.

linguists to see how interaction happens in a natural setting for instance) does help learners in acquiring certain structures that they find difficult to use. Obviously, data from newspapers or published books are written samples that were formerly approved by the editor and thoroughly reviewed by a certain number of experts, but then conversational samples feature more spontaneity and creativity as they were not revised to stick to a normative standard[8]. In both cases, these samples are authentic data in the sense that L2 users would also have to go through a thorough review of their writings to be published, and are likely to also display as much disfluency, such as hesitations, as the native samples. We thus argue that providing L2 users with the strategies used by native speakers to gain momentum would not do them any harm and could actually be helpful to them. As a matter of fact, native speakers are also frequently 'deviant' of their native language norms.

Our objective is to provide a complementary linguistic resource for L2 learners and L2 users who might want to see how their target language is used by native speakers, even though it does involve a comparison between their own use and that of native speakers, we do not encourage judgments. The data shown in native corpora using our tool are examples of what *may* be encountered in real life, what *can* be said but not necessarily what *should* be.

There are also more pragmatic reasons for our proposition of using native corpora in language learning. We believe that resources used by linguists to observe and describe language in use and the analyses resulting from those observations might help learners in different ways for different types of learners, from those who need more explicit rules to those who need more input because 'they are more intuitive'.

We also believe that this kind of help is not already provided by the wide range of pedagogical materials or other linguistic resources available to language learners such as:

1. lexicon (monolingual, bilingual, specialised, technical);

2. dictionary (monolingual, bilingual), useful for the definition (meaning) of a word, its pronounciation, its use, its synonyms and/or antonyms, and the

[8]We do believe, however, that the spoken variety has its own standards.

collocations where it is typically found;

3. conjugation books, for verbal inflections and constructions;

4. grammars for grammatical rules and syntactic constructions;

5. textbook and their exercise book;

6. and of course, the Internet with its 'unlimited' textual data and audio as well as videos, useful for exposure but overwhelming to answer a question regarding language learning.

2.3 The Use of Corpora in Language Learning

2.3.1 Indirect Use: Statistics and Examples

Researchers have shown a growing interest in the use of native speaker corpora in language teaching and learning for more than a century. In the first half of the twentieth century, word lists were published, which were derived from native corpora for teaching purposes. A good illustration of this early interest is the work of the American psychologist Edward Lee Thorndike, who wrote a series of books addressing the needs of language teachers. These books were published one decade apart from one another and each of them provides ten thousand words, reaching up to 30,000 in the last book[9].

Figure 2.1 was extracted from the first part of the book and displays a detailed account of words occurring at least once per 1,000,000 words. We can see that Thorndike made individual counts for each type of source that he had separated into different columns, where T stands for the Thorndike general count of 1931, L for the Lorge magazine count, J for the Thorndike count of 120 juvenile books and S for the Lorge-Thorndike semantic count. Column G states the occurrences

[9] *The Teacher's Word Book* published in 1921, *A Teacher's Word Book of the Twenty Thousand Words Found Most Frequently and Widely in General Reading for Children and Young People* published in 1932, and finally *The Teacher's Word Book of 30,000 Words* published in 1944 and this time co-written with a colleague psychologist, Irving Lorge.

per million words in the whole corpus [Thorndike and Lorge, 1944]. This presentation is useful in that teachers who are only interested in the frequency of words in children's literature because they are teaching young children might only look at the figures in Column J, while those interested in adult reading would rather focus on Column L. By order of importance, words marked AA are more frequent than those marked M, and the greater the number in each column, the higher the frequency of the word is in the corpus.[10] Most of the time, inflected forms are counted under the 'basal word' (or *lemma*, as we may call it) but some might be counted separately, as is the case in this example, with *knowing* and *known* appearing beside *know*.

	G	T	L	J	S
know	AA	M	M	M	M
knowing (adj.)	20*	?	194	?	?
knowledge	AA	400	465	400*	640
known	AA	M	M	?	600*
Knox	3	6	15	3	40
Knoxville	1	2	8	2	14
knuckle	5	18	38	16	21

Figure 2.1: Small excerpt of the frequency word list from Thorndike and Lorge [1944]

Another way of using this word book is to jump to the last part of the book, entitled "List of the 500 Words Occurring Most Frequently and of the 500 Words Occurring Next Most Frequently". This section is divided into two parts, each containing a list of words in alphabetical order, the first for the most frequent words ranked 1 to 500 and the second for words ranked 501 to 1000. Among the words appearing in Figure 2.1, only *know* and *known* appear, respectively, in the first list and in the second.

[10] These values are not just mere counts, but the results of a calculation based on 'credits'; Thorndike gave words taking into account their proportion in the original source. The psychologist also ensured that words appearing a certain number of times in numerous sources that he used for the Thorndike general count of 1931 had more credits than those appearing the same amount of times but in a single source. For more details on this calculation, see Appendix A in Thorndike et al. [1932].

Fries and Traver [1940] report that before their application to education, word counts were made for stenographers, essential to a time with no recording or dictation machines. Word lists then helped stenographers to determine the importance of abbreviating certain words based on their frequency of occurrence.

The efficiency of this method needs no further proof and frequency of words and collocations is still used in this way in second language pedagogy nowadays. Some of the most notable applications of these methods are the Collins COBUILD monolingual dictionary series and the Japanese TV programme "100Go de Start Eikaiwa" (*Let's start English with 100 keywords*).

In COBUILD dictionaries, illustrative examples were taken from authentic native corpora and definitions are ranked using word frequencies (described in Sinclair et al. [1987]).

Taking a step further, *100-go de Start! Eikaiwa* "100語でスタト!英話" ("Let's start English with 100 Words!") is a Japanese TV programme teaching conversational English through the introduction of keywords and their most important collocates according to the British National Corpus, [11] from which examples of use in context were also given. This program was the first research-based projet to have reached a target as large as the general public and to have achieved such a successful impact, as it was broadcast for three years on NHK, one of the major national channels in Japan [Tono, 2011].

The works mentioned above all rely on statistics to give relevant information on the most prominent usage of words based on word counts. Next to those works on isolated words, Palmer [1933]'s pioneer work on the key role of collocations in mastering a foreign language was also to be found in the same period.

[11] A-100-million-word corpus of both the written and spoken language, which is known to be the most representative large corpus of British English of the late twentieth century. The corpus is freely available from: http://www.natcorp.ox.ac.uk/.

However, these utilisations of native corpora are still limited in some way to one dimension of native speaker corpora if we consider all of the possibilities that could be offered by such a resource: they only focus on the acquisition of vocabulary and are mainly helpful to teachers for preparing relevant pedagogical materials. In the above-mentioned works, language learners were rather indirect users, as there was always an intermediate between them and that informative primary material.

2.3.2 Direct Exposure

Native corpora started to be fully used as resources of authentic data only around half a century later. Beyond the creation of lists of preselected words and phrases, native corpora display actual use of a language, which learners can benefit from by being more directly exposed to what is often called real language. Indeed, this observation phase of the target language allows learners, firstly, to observe variations and, secondly, to notice regularities or patterns in a language [Holec, 1990] and therefore to develop hypotheses on an object whose inner workings have hitherto always been given to them and taken for granted.

In recent years, the most wide-spread use of corpora in language learning and teaching is done through concordancers, whether directly using exploratory tools or indirectly with print-out KWIC[12]. This direct confrontation to authentic data enhances active learning, as learners explore by themselves what is really in use and can induce and grasp morphological and syntactic, as well as pragmatic features that are used by native speakers of their target language. Even phonological and prosodic features could be integrated if audio files were provided along with transcriptions of an oral corpus. Furthermore, in one of his experiments, Boulton [2009] found that learners with low-level language ability could also benefit from KWIC presentations, meaning that this approach could be applied to a wider range of learners than originally assumed.

[12]KeyWords In Context, the result of a query on a keyword which is a list of contexts where this keyword can be found in a given corpus. This notion is thoroughly explained in the section below.

2.3.3 Data-Driven Learning

Leading learners to be like researchers, in other words enabling them to be capable of making hypotheses, observing and processing data and actively seeking proofs to confirm or to invalidate their first hypotheses, and eventually making new ones if needed, is what Tim Johns was working towards when he developed the Data-Driven Learning approach. Indeed, in this approach, language learners are seen as "research worker[s] whose learning needs to be driven by access to linguistic data" [Johns, 1991, p.2]. Being actors of the construction of their own knowledge could help learners to develop reflexivity [Kettemann and Marko, 2011], as well as linguistic skills and learning strategies, thus making them more autonomous [Albero, 2000b]. Autonomy is crucial for anyone, but especially language learners. As described in Albero [2000a], the notions of autonomy and self-education have a variety of definitions, from the key to existencial and social emancipation, to the objective of any language learner to be able to autonomously interact in the environment of their target language. In the latter case, autonomy can be considered as the set of skills needed to manage one's own learning, the means to adapt to any situation and the condition for the ultimate successful achievement of training.

Using corpora to work towards autonomy does not consequently mean that the language teacher ceases to be of importance. The role of the teacher is still important if not essential to help and guide learners in their discovery of language through direct access to native corpora [Kettemann and Marko, 2011]. More guidance and less autonomy could even be better for learners in institutions, as discussed by Ciekanski [2014] in her careful study of the real perspectives opened by corpora as a resource for a more autonomous language learning. Yet despite the research papers about the benefit of the use of corpora in language learning and teaching, we cannot help but notice that corpora are still not integrated into most curricula and are not used in class. One of the reasons behind the discrepancy between theory and practice might be precisely the fact that if teachers want to integrate corpora in their classrooms, they need to be familiar with both this kind of data and the query tools, which is not self-evident. On the one hand, corpus data can be overwhelming, and on the other hand, the interface of current query tools

might look too complex and often require training to be used accurately [Boulton, 2012].

Chapters 4 and 5 present systems that address this dual complexity. Chapter 4 gives an overview of corpus exploration tools and describes their effort to adapt to non-specialist users such as language teachers and language learners, notably with regard to the interface. In Chapter 5, we describe the system we built with the idea of facilitating the familiarisation with the exploration of authentic corpora.

Chapter 3

The Corpus as a Linguistic Resource

3.1 Introduction

Theoretically, any collection of more than one document can be called a "corpus". In French high schools, for example, it is common to study and write essays on a *corpus de textes* ('corpus of texts') composed of usually 3 or 4 excerpts from different works, in French Language and Literature class. However, the common definition used in modern linguistics implies specific criteria:

> "A corpus is a collection of pieces of language text in electronic form, selected according to external criteria to represent, as far as possible, a language or language variety as a source of data for linguistic research." [Sinclair, 2005, p.16]

corpus

According to this definition, a corpus cannot be dissociated from its original purpose: the corpus is first and foremost a resource built to serve a *specific purpose* and is therefore somehow *representative* of a *language variety*. The language variety, the size of the corpus, and its format depend directly on the purpose.

One could wonder why a corpus necessarily has to be stored electronically, given that corpora were used before the advent of computers, and that the Bible was manually quantatively studied in multiple languages before concordancers were invented. The readers must keep in mind that there is not a single conventional definition of a corpus, and that the one that we present in this dissertation is used

within the framework of *modern* linguistics. As a matter of fact, the format of the corpus implies not only that its size is big enough to be unbiased and representative of a language variety but also that it can explored qualitatively and quantitatively using corpus exploration tools (which we present in Chapter 4), both of which ensures that it is relevant for linguistic research.

O'Keeffe et al. [2007, pp.1-2] share the same view of a corpus, as they list three major properties for a corpus: "principled", "usually stored on a computer" and "available for qualitative and quantitative analysis".

With the expansion of corpus linguistics studies, the use of corpora may sometimes be presented as something self-evident and almost natural in linguistics.[1]. However, historically, other methods were obviously used before any corpus was investigated, and those methods are still methodologically viable for some purposes. Incidentally, no corpus is suitable for all purposes, and all purposes cannot be satisfied with the help of corpora.

In this chapter, we present the constitution and the use of the corpus as a linguistic resource. Section 3.2 introduces the context in which a change of perspective occurred in linguistics, while Section 3.3 gives an overview of the different existing types of corpora. As we have seen, a corpus is always constituted with a specific objective, and we will see in Section 3.4 that each decision, each step from the preprocessings to the actual processings and annotations of the corpus, serves this objective. Finally, in Section 3.5, we illustrate the processings of a corpus by presenting the one that we are using in our experiments, the Sejong Corpus.

3.2 The Need for Attested Data

The most famous notorious and influential linguistic dichotomy in Europe is that of *langue* and *parole* from Ferdinand de Saussure, considered as the founder of modern linguistics in Europe. Saussure distinguishes the language as "a system of linguistic signs considered in and of itself and shared by the members of a linguistic community" which he refers to as *langue*, and language as "the virtually

[1] And if we have insinuated such a thing as well, it was not our intention.

infinite set of written or spoken utterances produced by the individuals of such a community", which he calls *parole*, literally, speech.[2] This dichotomy exists in generative grammar under the competence/performance distinction. Competence is not related to an individual, nor to a language community as a whole, but rather to an *ideal* speaker-listener. Conversely, performance varies with the speaker's history, with the situation of communication, and other extralinguistic factors.

While performance can be studied with corpus-based observations, competence is better approached by introspection, if the language is one's own, and by questioning native speakers.

Corbin [1980, p.155] (cited in Jacques [2005]) considers that:

> "[ce qui fait l'intérêt de l'introspection, c'est] la possibilité d'envisager d'autres énoncés que ceux qui sont attestés. *L'introspection peut alors être conçue comme l'instrument privilégié d'une recherche sur les limites ultimes du possible prédictible à partir des observables.*"[3]
> ("[the interest of introspection is] the possibility to consider utterances, other than those that are attested. *Introspection can thus be conceived as the priviledged instrument of a research on the ultimate limits of what is possible and predictable from observable data.*")

Indeed, corpora are collections of texts, of authentic and attested data, but cannot account for what is not attested. However, what is not attested is not necessarily wrong, or odd, especially within the limits of a given corpus that is *representative* and necessarily *limited to* a variety of language. Competence is larger than performance in this sense: it is impossible to produce or observe all of the possibilities offered by competence. The limits of corpora are overcome by introspection but the contrary is indeed also true: the limits of introspection are exactly where corpora start to be useful.

The main critics that are addressed to introspection are that introspective methods rely exclusively on judgments that are not always reliable, but primarily that it cannot account for variation. For a certain number of disciplines related to

[2] These definitions were extracted the "Competence/Performance" entry written by Anne Abeillé for Houdé et al. [2004].

[3] Italics are from the original text.

linguistics, including sociolinguistics, variation is their very heart, and even at the heart of competence: can we consider that competence is really identical for all of the native speakers of a language, regardless of their backgrounds?

Conversely, corpora are the most reliable resources in order to account for language variation. Word frequency rates – and textual statistics in general – allow to have a quantitative and an objective approach on language (as we have seen in 2.3.1 with the indirect use of corpora in language teaching). There is not a single way of using corpora: sometimes, corpus-based studies is a term that encompasses all studies involving the use of corpora, and sometimes they are opposed to corpus-driven studies.

In this respect, **corpus-based studies** are often called *top-down* approaches: they are used to verify or illustrate a theory, and start from an intuition, an idea or a hypothesis which is then confronted to authentic data, and is, finally, either validated or discarded according to the observations made. Conversely, **corpus-driven studies** are said to be *bottom-up* approaches: they start directly with observations of authentic data, from which a hypothesis is formulated. McEnery and Hardie [2012, p.151] are not convinced by this opposition, which they rather consider as a "sliding scale", and argue that the distinction between the two schools relies on their stances on the status of corpus and corpus linguistics: the former considers corpus linguistics as having a theoretical status, while the latter considers it as a linguistic methodology.

Our work claims to be inspired by the "Data-Driven Learning" approach, developed by Johns and briefly described in 2.3.3. However, with regard to this corpus-based *vs.* corpus-driven issue, we believe that our system can be used in both types of studies: while the "learner as researcher" method clearly identifies as a top-down method, the fact that our system is based on similarity and not strict matching, especially in the distributional analysis search mode, allows for more serendipitous findings.[4] Either way, access to authentic data is at the heart of our system.

[4]See Chapters 5 and 6, especially the description in Section 5.3.4 and the results of experiments in Section 6.3.

3.3 Types of Corpora

Corpora are not a resource used only within the realm of corpus linguistics, but were exported to many fields related to language. There are therefore as many types of corpora as needed in different types of studies.

For McEnery and Hardie [McEnery and Hardie, 2012], the different types of studies in corpus linguistics are defined by the following features:

- Mode of communication: spoken language *vs.* written language *vs.* sign language;
- Corpus-based *vs.* corpus-driven linguistics (see the discussion in Section 3.2);
- Data collection regime: monitor corpus approach (where the corpus continually expands) *vs.* balanced/sample corpus approach;
- The use of annotated *vs.* unannotated corpora;
- Total accountability *vs.* data selection;
- Multilingual[5] *vs.* monolingual corpora.

For more detail on each feature and presentations of different corpora displaying different combinations of these features, we invite the readers to refer to McEnery and Hardie [2012].

Like the selection of the corpus, the type of study depends on its purpose. For instance, our study on Korean:

- uses written samples, but only due to lack of time to analyse the results from the experiments conducted on the spoken samples extracted from the Sejong Corpus.
- is not specifically corpus-based or corpus-driven, but allows both types of studies, as explained earlier.
- uses a balanced corpus known as the Sejong Corpus, or the Korean National Corpus.

[5]Multilingual corpora are called comparable or parallel corpora depending on the degree of alignment.

- uses the morphosyntactically annotated samples, given that our objective is to seek syntactic constructions.
- is based on randomised samples to keep the total accountability of the corpus, as our objective is not to seek specific examples.
- uses the monolingual part of the Sejong Corpus.

The last feature results from a deliberate choice. Indeed, the Sejong Project produced large monolingual corpora, as well as smaller multilingual corpora (Korean-English, and Korean-Japanese). The use of multilingual corpora (in the broad sense encompassing bilingual compora) in a language learning application would be highly beneficial for language learners, especially beginner learners who might be troubled by unknown words or expressions. Multilingual corpora are called *parallel corpora*, if they contain original texts that are translated in different languages, and often *aligned*[6], or *comparable corpora*, if the different texts are similar in genre, topic, and register but are not strictly translations of one another. Linguee[7] is a search engine that allows to search for expressions in parallel corpora in as many as 25 languages, mostly European. It also integrates a dictionary statistically based on these corpora. Despite the interesting possibilities offered by multilingual corpora, we chose to focus on monolingual corpora for many reasons: first, multilingual corpora are usually smaller and rare, which would limit the scope of our system; second, we believe that to be confronted to monolingual data may help learners to focus on the constructions of the target language, instead of relying on translations.

parallel corpora

comparable corpora

3.4 Corpus Processing

3.4.1 General Overview

Whatever the purpose of the study, a corpus needs to be prepared and undergo a certain number of transformations.

[6] Data are said to be aligned if pairs of translated items are identified. Alignment can occur on paragraphs, sentences, phrases or words. The smaller the unit the easier it is to retrieve precisely how an expression was translated.

[7] http://www.linguee.com

The **core of the processings** is determined by the objective of the study: the **segmentation** defines the units and therefore sets the granularity of the study while the **annotation(s)** directly depend on the nature of the study. It would indeed be impossible to study the use of a syntactic phenomenon relying solely on the wordforms. In Section 4.3.3, we give the example of the progressive verbal form in English *V-ing* which would be unnecessarily complex to study without any syntactic annotation. The morpheme *-ing* is specific to the progressive in English but as a string of characters *ing* appears frequently in words such as *thing* that would be noisy for this study.

Corpus processings also include what is called *pre*-processings (before the main processings) and *post*-processings (after). Both usually pertain to the formatting issue related to the use of segmentation and annotation tools. In addition to that, some preprocessings are rather determined by the nature of the corpus.

Figure 3.1 is an example of processing chain for a corpus, starting on the left with optional preprocessings (transcription, normalisation, data cleaning) followed by the first processing, the segmentation (here, tokenisation) and then by a series of annotations (POS-tagging, parsing, lemmatisation, semantic analysis) performed in a certain order. Examples of parsing and semantic analyses are given in gray next to the corresponding box.

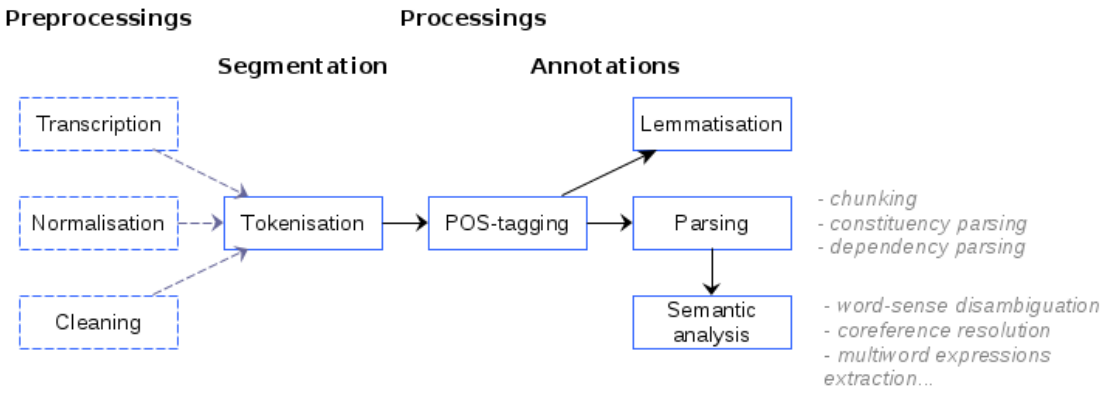


Figure 3.1: Example of processing chain for a corpus

In this section, we briefly describe the processings introduced in the flowchart while insisting on issues that we consider crucial for any study based on corpus linguistics.

3.4.2 Preprocessing

Different types of corpora were described in Section 3.3 and we have seen that the methods of data collection may differ highly depending on the nature of the data: for instance, while written data are most of the time simply copied and sometimes scanned, multimedia data are collected through audio or video recordings. At this stage, we end up with data as various as textual data, either raw in `.txt` files or *noisy*, audio files and video files.

As the linguistic part of corpus processing is based on tools which take as input a stream of textual data, before any processing can be performed on the collected data, in some cases, preprocessings are necessary. In this section, we chose to focus on three types of preprocessings.

Transcription Any spoken corpus has been transcribed, either phonetically or orthographically, or both. Orthographic transcription is still mostly manually done, usually by naive native speakers (e.g students in Linguistics) under the supervision of a linguist. Using a speech processing tool such as Praat or Transcriber[8], they listen to the signal and transcribe what they hear simultaneously. This method is efficient but has drawbacks: human transcription is a tedious work that not only has a cost but also tends to be "subjective and unreliable" [Goddijn and Binnenpoorte, 2003, p.1361].

By contrast, automatic transcription would help having a more objective and reliable output but has serious obstacles still standing in the way of an accurate result. Those obstacles range from practical reasons to inherent difficulties. On the one hand, the quality of the signal depends on both the quality of the recording tool and on the conditions of recording: the poor quality of a recorder, a noisy background or the overlapping speech of multiple speakers may all lead to a hardly

[8]Both are free tools available at `http://www.fon.hum.uva.nl/praat/` for Praat and `http://trans.sourceforge.net` for Transcriber, both consulted on 9th June 2017.

comprehensible signal. On the other hand, as a matter of fact, spoken language can be ambiguous because of homophones (similarly to homographs for written data), as well as because the word boundaries are inevitably blurred.

Normalisation Data collected from the web (forums, messages from social networks or comments in review websites for example; see Baranes [2015] for French) as well as from text messaging (see Han and Baldwin [2011]) are prone to having non-standard forms. In order to use processing tools that are meant to be used on standard forms, one must perform a round of normalisation.

In those *noisy* corpora, common phenomena include typing, spelling and grammatical mistakes, as well as *ad hoc* abbreviations (e.g *ily* for "I love you") and reductions such as consonant contractions (e.g in French *tkt*[9] for "t'inquiète" (*don't worry*), in Spanish *mñn* for "mañana" (*morning*, *tomorrow*), in Italian *scs* (*sorry*) for "scusa" [Panckhurst, 2010] or in English *ppl* for "people").

Cleaning In addition to the normalisation process, collecting data from the web also involves another preprocessing: cleaning non-textual or undesirable data. Most webpages come in an HTML (HyperText Markup Language) format, which, as its name suggests, is not composed of raw textual data but is a structured format based on a *markup system* using *tags*. Those tags are elements written in angle brackets such as `<tag/>` but most of them work in pairs surrounding a content, either textual or another tag: `<tag attribute=''att_value''> text </tag>`. The whole document can be represented as a tree of tag nodes, following recommendations from the World Wide Web Consortium, better known as W3C[10].

Collecting data from the web therefore means extracting a specific content among other content and structural information. In order to do so, we have two possibilities: the first consists in targetting specific content based on a textual cue and clearing whatever surrounds; the second consists in going through the docu-

[9] In this case the letters "qu" have been replaced by their homophone "k" but the abbreviation *tqt* does exist as well.

[10] To be valid, an HTML page has to abide by certain basic rules. However, apart from these, web developers are rather free to structure their document as they will, although it is highly advised to comply by the W3C recommandations. See https://www.w3.org/

ment tree and extracting the desired textual nodes' content. The second method is recommended as it makes good use of the structural nature of HTML documents, but this implies that the document is well-formed and consistent, which is not always the case.

Figure 3.2 is an example of navigation throught an HTML tree. As a high number of written corpora include articles from online newspapers, we chose to use the online version of the British daily newspaper *The Guardian*[11] to illustrate the second method. Mozilla Firefox gives the possibility to 'inspect an element' from a page by right-clicking on it, in this case, the headline of an article. This feature opens the frame that we see on the bottom part of this capture, showing the complexity of the webpage structure: in the source code, the actual targetted text (in black) only appears after a succession of nested elements (each indentation represents a deeper level of nesting). Tree parsing tools allow to bypass this apparent complexity by enabling the direct targetting of the headline node based on attributes for example, in this case, either `class=''content__headline''` or `itemprop=''headline''` which are both unique within the page.

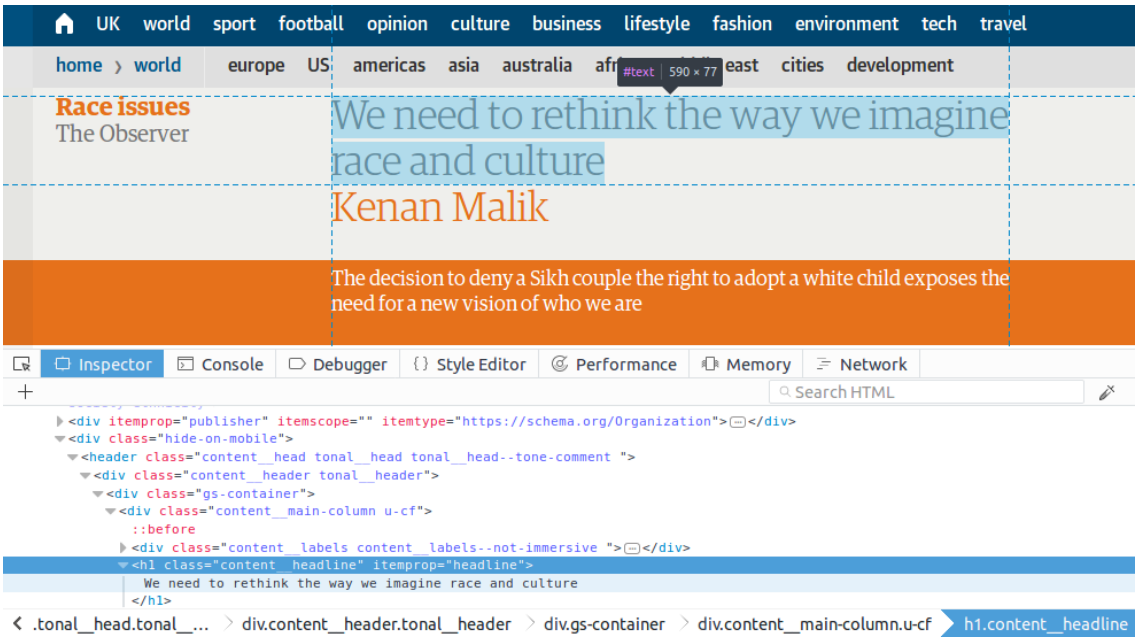


Figure 3.2: Screenshot of an article from *The Guardian* and its source code

[11] Accessible on `www.theguardian.com`, last consulted on 3rd July 2017

3.4.3 Segmentation

Once preprocessings are completed, the data can be processed and the very first step consists in segmenting the stream of characters into minimal units. Those units may correspond to words, but we will see in this section that segmentation in NLP is often called *tokenisation* because the notion of *token* is preferred, mainly for practical reasons. Defining the minimal units for a corpus is crucial as segmentation serves as a basis; any error of segmentation will affect the rest of the processing chain.

3.4.3.1 The "Word" Issue

Words in general Beyond the popularisation of search engines and keywords, *words* have always been an intuitive unit of speech. Indeed, *word* is a metalinguistic term that is commonly used outside of the linguistics sphere, which makes it comprehensible by a vast majority of people, unlike *morphemes* or *lexemes* for instance. It would not be surprising indeed to hear an adult correcting a child by saying "You are using the wrong *word*.", meaning "You are using the wrong form to refer to this object". The Oxford English Dictionary gives the following definition for a word as an "element or unit of speech, language, etc."[12]:

> "Any of the sequences of one or more sounds or morphemes (*intuitively* recognized by *native speakers* as) constituting the *basic* units of *meaningful speech* used in forming a sentence or utterance in a language (and *in most writing systems normally separated by spaces*); a lexical unit other than a phrase or affix; an item of vocabulary, a vocable."[13]

This definition confirms what we said about words being an intuitive notion, and that no specific instruction in language studies is needed to comprehend it: just being a native speaker seems to be enough. It also gives important properties of words: first, words have a **specific meaning** and a **specific form** in a given language; secondly, they are opposed to *phrases*, units that are composed by several

[12] Interestingly, this is only the third acceptation of 'word', the first two being the word in Christian Church ("singular, mostly with possessive or *the*; often in fuller forms as the *word of God*, *God's word*"), and the word as "speech, utterance, verbal expression".

[13] Italics are added.

words, and to *affixes*, morphemes that are non-autonomous components of a word; and thirdly, words are **part of the vocabulary**, i.e., the "sum or stock of words employed by a language, group, individual, or work or in a field of knowledge"[14]. Another interesting point is the fact that words are said to be separated by spaces in most writing systems, which implies that some writing systems do not use spaces to separate words, and indeed, **word segmentation** in languages such as Chinese, Thai or Vietnamese[15] is not a trivial task but a real challenge (see Magistry [2013]'s work on wordhood in Chinese for example). This typographical or orthographic definition of word seems simple but is in fact difficult to comprehend for people who have not learned to read and write, let alone speakers of an oral language.

Words in linguistics In linguistics, the notion of word is much more complex and controversial: the term 'word' covers different concepts and refers to different entities depending on the level of analysis – whether we think of words as syntactic or semantic units for example. Bloomfield [1935, p.170] gives two definitions of word; a general definition acknowledging the word as "the smallest unit of speech [for the purposes of ordinary life]" and that we assume would be quite close to the definition from the dictionary, and a more linguistics-oriented definition:

> "A free form which is not a phrase is a word [...] in brief, a word is a minimum free form".

Free forms are forms which "occur as sentences" and "can be isolated in actual speech", as opposed to **bound forms**. Bloomfield gives as an example the *'s* [z] form which can alternatively be a free form, i.e., a word, when used as the verb "to be" in *John's ready* and a bound form when used as the possessive morpheme in *John's hat*. Yet, the verbal form "is" seldom appears alone. To justify the fact that the verbal form "is" should indeed be considered as a a free form, Bloomfield argues that "the linguist cannot wait indefinitely for the chance of hearing a given form used as a sentence" and gives the following fictive dialogue instead: *Is? – No; was*. He also acknowledges that "[i]n the case of many languages [...] it is impossible

[14] Definition 2a of *vocabulary* retrieved on the Merriam-Webster website on 5th April 2017.

[15] In Chinese, spaces only delimit sentences along with punctuation; in Thai, phrases and sentences are delimited but not words; and in Vietnamese, spaces are boundaries to syllables, not to words.

to distinguish consistently, on the one hand, between phrases and words and, on the other hand, between words and bound forms" [Bloomfield, 1935, p.179].

As a matter of fact, this syntactic property is not the only criterion as Bloomfield considers grammatical units like *the* in English or conjunct forms *me* or *te* in French[16] as words although they are rarely used alone. He justifies by saying that they "play much the same part in [their respective] language" as forms like *this* or *that* for *the*, and *moi* or *toi* for the French pronouns.

Words in NLP What is considered as a word for a corpus exploration tool depends highly both on the language studied and on the parameters set by the software developer. In the case of *'s*, a concordancer might consider the apostrophe ' as a *word delimiter* or *word boundary* and therefore cut systematically between *John* and *'s*. This is the case for AntConc, as shown in Figure 3.3, where we can observe a concordance of the 'word' *s* taken from the first chapter of *A Study in Scarlet* by Arthur Conan Doyle.

word boundary

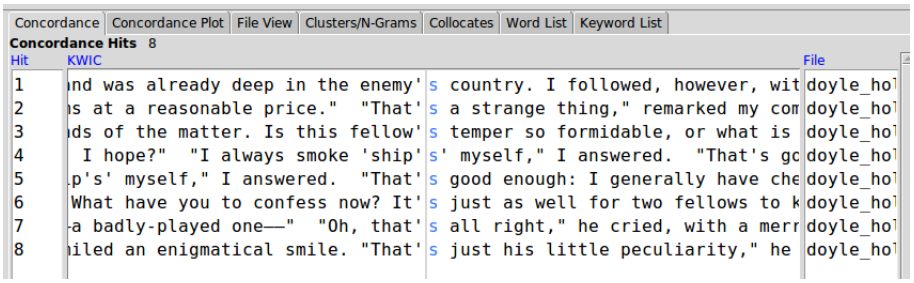


Figure 3.3: Concordance of the 'word' *s* using AntConc

The concordancer finds eight occurrences of *s* as a word, three times as the possessive (lines 1, 3 and 4 as part of apparently a tobacco's name) and five times as the verb form "is" (lines 2, 5, 6, 7 and 8). According to Bloomfield's definition, this automatic segmentation is then wrong 3 out of 8 times. Furthermore, if we look at Figure 3.4, we note that the contraction of the negation in the form *n't* has also been considered as a word, but only the *t*, seperated from the forms *don*, *didn*

[16] In French, personal pronouns (called disjunct or *non-clitic* pronouns) have a conjunct or *clitic* counterpart, that is to say a form with the syntactic characteristics of a word, but which depend phonologically on another word or phrase.

and *mustn*. Because this error is systematic it can be easily avoided using either of these methods:

- the first solution would be to **preprocess** the corpus and segment only when *'s* is a verb (i.e., to separate *that* from *'s* "is" but to keep *don't* or *enemy's* as words). Also, to ensure that the apostrophe ' is not used as a word boundary, it should be added in the list of all the characters that are considered part of a word (or rather *token*, a notion described in the following subsection) in the settings of the tool[17];
- the second solution would be to **normalise** the corpus, i.e., to replace *altered* or *ill-formed* words by their standard form. In our case, this process would split the form *that's* into the two separate forms *that* and *is*.

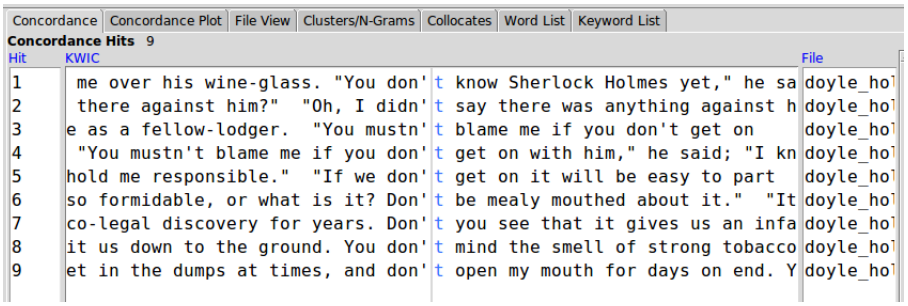


Figure 3.4: Concordance of the 'word' *t* using AntConc

Incidentally, the second solution could be extended to **lemmatisation**. This text processing task consists in giving each *token* a label with its corresponding *lemma*. The form *that's* would then be labelled as such: *that_ THAT 's_ BE*, with the syntax *token_ LEMMA*. The processes of normalisation and lemmatisation are described in 3.4.4.

[17]Default settings often consider that words can only contain alphabetical (small or capital) letters; conversely, any other character is automatically considered as a word boundary. Incidentally, AntConc also has specific classes for number tokens, punctuation tokens, symbol tokens, mark tokens as well as a fully user-defined token class. Apart from the last class, all classes are pre-defined and users only need to check the boxes of the categories they are interested in. For example, if the "math" box or the "currency" box in the symbol class is checked, the program will interprete symbols like € or £ as words in their own rights, just like *euro(s)* or *pound(s)*.

3.4.3.2 Tokenisation

token

Words have multiple definitions according to the use or to the linguistic background. In NLP, the denomination of '*token*' is often preferred to that of 'word', for words have a specific meaning in computer sciences which has nothing to do with linguistics apart from the fact that they are also minimal units, but comprehended by the hardware of a processor. We have seen indeed that the minimal unit in AntConc is the *token*, but other NLP tools might still use *word*, especially if they are aimed at non-specialists. As for specialists, they find more convenient to define a technically adequate notion which would have the benefit of being "at once **linguistically significant** and **methodologically useful**"[18] [Webster and Kit, 1992, p.1106]. A second definition from Webster and Kit helps us understand these two properties:

> "A token will not be broken down into smaller parts. In other words, for the purpose of computational processing, it can be treated as an *atom*[19]." [Webster and Kit, 1992, p.1109]

The fact that a token is the smallest meaninful unit according to Webster and Kit does not mean that it *cannot* be broken down but rather that it *should* not: what we consider as a token defines the **granularity of the analysis**.

A token is therefore not always a word nor a sequence of characters surrounded by word boundaries. Interestingly, we can note that using the form *don't* as an input in the online corpus exploration interface corpus.byu.edu[20] raises the following error: "In nearly all cases, the tagger separates words that have an apostrophe (e.g. we're or don't) or which are a contraction of two separate words (e.g. gonna or gotta). These need to be entered as two separate words".

The correct input in this case is *do n't*, a query composed of two tokens that are indeed both "linguistically significant" (in contrast to the previously mentioned forms *don* and *t* if the segmentation is to happen on the apostrophe) and "method-

[18] Emphasis was added.
[19] Bold emphasis in the original article.
[20] A popular portal hosted by Brigham Young University and allowing access to multiple reference corpora, notably the CoCA (the Corpus of Contemporary American English) and the BNC (the British National Corpus). The portal as a corpus exploration interface is described in 4.4.

ologically useful" as this segmentation allows to explore either each unit individually or together as a bigram (see the n-gram section below) and the token *n't* enables queries on the contracted form exclusively.

Types Eventually, the notion of token is often paired with that of *type*, another fundamental notion in computational linguistics. In the glossary of the MOOC[21] he designed, Tony McEnery defines tokens as "examples of the same *type*" and a *type* as:

type

> "[...] a single particular wordform [and] any difference of form (e.g spelling) makes a word a different type."

In other words, a corpus is segmented into tokens, but each token is counted under a single type. The same distinction exists in French linguistics, but with a slightly different terminology:

> "A series of non-delimiters whose bounds at both ends are delimiters is an occurrence. Two identical series of non-delimiters constitute two occurrences (tokens) of the same word (type)" [Lebart et al., 1997, p.23]

Following this definition, what is called a *forme* or *forme graphique* in French is a wordform, in other words, a type. Conversely, a token is called an *occurrence*.

The ratio resulting from the comparison between types and tokens is a well-known measure of **vocabulary diversity** in a corpus. Again according to McEnery's glossary, the ratio "equals to the total number of types divided by the total number of tokens. The closer the ratio is to 1 (or 100%), the more varied the vocabulary is."

3.4.4 Annotations

As seen in Figure 3.1, once the corpus is segmented, a range of annotation processings are applicable. In this section, we briefly describe the annotation processes that are related to our work.

[21] The Massive Open Online Course "Corpus Linguistics: Method, Analysis, Interpretation" is available on Futurelearn since 2013 and is open to "anyone who has an interest in the study of language", according to the introduction of the course: https://www.futurelearn.com/courses/corpus-linguistics, accessed on April 21st.

3.4.4.1 Morphosyntactic Tagging

Also commonly called POS-tagging (which stands for Part-Of-Speech tagging), morphosyntactic tagging is an important step in corpus processing on which other layers of annotations often rely on, as shown in Figure 3.1.

part-of-speech

However, parts-of-speech are not a notion of Natural Language Processing, but a notion of linguistics that dates back to centuries BCE, with the Indian grammarian Panini in Asia, and Plato in Europe. POS are also commonly known as "lexical categories" or "word classes", and they are indeed classes to which all of the words of any language are assigned. The nature and the number of POSs depends not only on the language, but also on the linguistic background: while it is not surprising that two typologically distant languages such as French and Korean do not share the same POSs, we also see in Section A.2 that linguists do not quite agree on the number of POSs in Korean. This situation is probably not unique, as linguists over centuries have used different criteria for the classification of POSs. Among them, we note:

- semantic criteria: e.g. adjectives describe quality
- morphological criteria: e.g. adverbs are invariable
- syntactic criteria: e.g. determiners appear before nouns

Since POS are not semantic classes but morphosyntactic, semantic criteria are not reliable for the classification of POSs. Conversely, it is possible to determine the POS of a word using morphological criteria (namely, *inflectional* and *derivational* criteria) and syntactic criteria, among which, the *distribution*. The distribution of a word is the range of grammatical positions in which it may appear.

distribution

For example, the word 'unfortunately' can be classified as an adverb for different reasons. First, we can use the inflectional criterion and note that 'unfortunately' is invariable in form. It therefore does not inflect for number (contrary to a noun like 'noun' (singular) *vs.* 'nouns' (plural)), neither does it for tense (contrary to a verb like 'like' (present) *vs.* 'liked' (past)) nor for comparison (contrary to an adjective like 'good' (positive) *vs.* 'best' (superlative)). Second, we can use the derivational criterion and note that 'unfortunately' is composed of the suffix *-ly*,

commonly used in English to derive adverbs. And finally, we can use the distributional criterion and note that 'unfortunately' may appear (2) at the beginning a sentence, (3) at the end of a sentence, (4) between a verb and its subject and perhaps, even (5) in isolation.

(2) Unfortunately, she did not come.

(3) She did not come, unfortunately.

(4) She unfortunately did not come.

(5) ? – She did not come. – Unfortunately!

3.4.4.2 Lemmatisation

Inflection is the use of *morphemes* which are added to a *base word* or *root word* to express different grammatical functions such as the number (singular, plural, dual etc.) or the gender (feminine, masculine, neuter) for nouns, or the tense (present, past, future), the mood (indicative, imperative, optative etc.), the aspect (perfective, imperfective, progressive etc.) and the modality or illocutionary force for verbs. Inflection does not affect the meaning or the category of the root word. Consequently, it is more suitable to group all inflected forms under a single form, the lemma, when studying the use of the verb "to be" as in the preceding section.

Lemmatisation is a processing particularly important for inflected languages such as English, French, or Romance languages in general. As for agglutinative languages such as German, Finnish or Korean, where each affix represents a single function, lemmatisation amounts to isolating the root from the affixes, in other words, segmenting words into morphemes.

Multiword Expressions One of the key problems to Natural Language Processing is the recognition of compound tokens called *multiword expressions* (hereafter MWE) and defined as:

multiword expressions

> "idiosyncratic interpretations that cross word boundaries (or spaces)"
> [Sag et al., 2002, p.2]

Indeed, MWE comprise units of various size ranging from compound of two words to expressions as large as idioms: *bus stop* (compound noun), *come down*

to (phrasal verb), *in the face of* (compound preposition), *Sir Arthur Conan Doyle* (named entities), or *to have more holes than a Swiss cheese* (idiomatic expression). All those expressions vary not only in size and category but also in behaviour. While some of them are lexicalised and have at least partially idiosyncratic syntax or semantics, others are institutionalised ("occur with high frequency in a given context") and are syntactically and semantically compositional to different degrees [Sag et al., 2002, p.3]. For a discussion on the classification and issues on the use of MWE in NLP, see for example Constant [2012].

What they have in common and what we want to stress is that all of them constitute a unit to some extent. An expression such as "pomme de terre" in French refers to a single entity (a *potato*) although it is composed of three distinct words "pomme" (*apple*), "de" (here, *from*) and "terre" (*earth*). MWE could be analysed as atomic units or represented as an additional layer of annotation on several tokens, depending on the purpose for which the corpus was built. For instance, the compound proper name "Sir Arthur Conan Doyle" could be treated as a single token and represented with non-boundary characters instead of spaces (for example, `Sir_Arthur_Conan_Doyle_NP0`[22]) or it could be segmented into four tokens based on spaces but considered as a single MWE referring to a single person (`[Sir_NP0 Arthur_NP0 Conan_NP0 Doyle_NP0]_person`).

Corpora using the CLAWS7 tagset such as the BNC or the COCA make use in theory[23] of *ditto tags* to group "a sequence of similar tags, representing a sequence of words which for grammatical purposes are treated as a single unit"[24]. MWE can therefore be retrieved by analysing POS tags: if a compound is composed of three tokens, a pair of numbers is added at the right end, the first being the position of the given token in the compound word and the second being the total number of tokens concerned. The presentation of the tagset include the example of the MWE *in terms of*, analysed: `in_II31 terms_II32 of_II33` (where `II` is the tag for "general preposition").

[22] NP0 is the tag for neutral proper names (neither singular nor plural) in the CLAWS7 tagset.

[23] In the XML version of the BNC for example, multiword expressions are grouped in `<mw>` tags so that a single tag is given to the whole MWE. In addition, proper nouns are actually never considered as MWE, as mentioned in the tagging guide of the BNC (http://www.natcorp.ox.ac.uk/docs/URG.xml?ID=posGuide, consulted on June 30th, 2017).

[24] Retrieved from the UCREL CLAWS7 tagset webpage: http://ucrel.lancs.ac.uk/claws7tags.html, consulted on 30th June 2017.

The identification of MWE is crucial in disambiguating sequences of words with different usages such as *a little*, considered in the BNC as a multiword adverb in *They are all a little drunk* but as seperate words when used as a quantifier meaning 'a small amount' in *You couldn't let me have a little milk?*.

MWE across languages The identification of MWE is crucial for tasks involving several languages such as comparative linguistics or translation, especially for idiomatic expressions which can be conveyed in totally different ways depending on the culture. While the British people cry "that takes the biscuit!" when a person or a situation becomes extremely annoying, Americans would be more gourmand and say "that takes the cake!". As for the French, they would rather ironically say "c'est le bouquet!" (litt. *that's the bouquet*) which is closer to the German idiom "Das ist die Gipfel!" (litt. *that's the top*).

But the size is not all that matters as smaller MWE may be equally tricky. MWE in a given language might be a single word in another. If we observe the translations of 'bus stop' in Example 6, we can see that a compound word in English can be translated by either a single word (see 6b 'bussipysäkki' in Finnish), by another MWE with a similar number of words (see 6c '*besu cenglyucang*' '버스 정류장' in Korean) or by another MWE again but this time with a different number of words (see 6a 'arrêt de bus' in French).

From the example of 'blackcurrant bush', we observe that the difference does not depend on the morphological typology. Both Finnish and Korean are agglutinative languages while French is a fusional language, and yet all of them may produce one-word compound words as in Example 7 (again, French is 7a, Finnish 7b and Korean 7c). It rather depends on the productivity of patterns and morphemes in the creation of new words: in the case of French, 'NOM de NOM' ('NOUN of NOUN') and the morpheme *-ier* are both very productive. As a matter of fact, the latter is extensively used to derive names of fruit trees and bushes, among other functions.

(6) 'Bus stop'

a. **arrêt de bus**
stop of bus

b. **bussipysäkki**
bussi-pysäkki
bus-stop

c. **버스 정류장**
besu cenglyucang
bus stop

(7) 'Blackcurrant bush'

a. **cassissier**
cassiss-ier
blackcurrant-NOM

b. **mustaherukkapuu**
musta-herukka-puu
black-currant-tree

c. **까막까치밥나무**
kkamak-kkachipap-namwu
black-currant-tree

Working with more than one language necessarily involves dealing with an uneven number of words for at least some compounds, let alone for a whole paragraph or a whole text. Such a situation may require an extra step in the processing of the corpus. If tokenisation is applied and MWE are not grouped together, the data in the two languages need to be *aligned* or explicitly linked.

3.5 Illustration: the Sejong Corpus

3.5.1 Presentation

The 21st Century Sejong Project (in Korean, *21seyki seycongkyeyhoek* 21세기 세종계획) is a ten-year long project launched in 1998. Several Korean universities, including Seoul National University and POSTECH (Pohang University of Science and Technology), collaborated with the ambition of providing the Korean language with a large reference corpus for both written and spoken varieties and an electronic dictionary based on the corpus. An exploration tool for each of the two resources was also developed: the kkma online concordancer on the one hand, and on the other hand, a dictionary manager on the DVD of the project.

Sejong Corpus

The reference corpus is named 'Sejong Corpus' after the project, but is also called the Korean National Corpus (sometimes abbreviated KNC). This corpus actually encompasses two corpora: a primary corpus for contemporary (South) Korean and a specialised corpus. The primary corpus is composed of a raw corpus of 59,635,608 *ecel* 어절 (58,829,962 for the written part, 805,646 for the spoken part) which was partly declined into different versions:

- a POS-tagged corpus of 13,302,421 *ecel* 어절 (12,496,775 for the written part, the whole spoken corpus – 805,646 for the spoken part);
- the written POS-tagged corpus was also partly enriched with disambiguated sense (11,443,305 *ecel* 어절);
- the parsed corpus of 677,349 *ecel* 어절 was also based on the POS-tagged corpus.

3.5.2 Segmentation

A specific unit for Korean: *ecel* 어절 The Sejong Corpus is segmented into *ecel* 어절, chunks of text separated by spaces which can be a single morpheme or a combination of several morphemes [Choi et al., 2016]. This unit is specific to Korean and different from the notion of word as understood in languages such as English or French and described in Section 3.4.3.

In English, a distinction is made between *free forms* and *bound forms*, i.e., forms that never occur as sentences and are never isolated in actual speech (according to Bloomfield [1935], see the "Words in linguistics" paragraph in Section 3.4.2). However, unlike English but similar to French, a bound form does appear between word boundaries (spaces) as shown in Example 8 with the bound noun *swu* 수, which conveys the meaning of possibility/impossibility when used in the construction *-(u)l swu issta/epsta* -(으)ㄹ 수 있다/없다. Korean bound nouns also appear with the same postpositions as regular nouns. In Example 9, *swu* 수 appears with the particle *-to* -도 meaning 'also' or 'either', a particle commonly found attached to regular nouns. Both examples are taken from a sample of the Sejong spoken corpus. [25]

(8) 운동을 할 **수** 있을지 모르겠다 [5CT_0013]
wuntong-ul ha-l **swu** iss-ul-ji molu-keyss-ta
sport-OBJ do-PRS way exist-PRS-whether ignore-may-DECL
'I don't know if I can engage in sports activities.'

(9) 기억할 **수**도 없어? [5CT_0013]
kiekha-l **swu**-to eps-e?
remember-PRS way-also lack-INF

[25] In the literature, bound nouns are also called dependent nouns or defective nouns but we hold on Bloomfield's terminology throughout this dissertation.

'Can't you even remember?'

Except for the raw corpus where the texts are only segmented in sentences, the annotated versions of the Sejong Corpus are all segmented into *ecel* 어절. A sentence in the raw corpus such as the one from Example 10 appears as such, and only paragraph tags (<p>) separating all sentences are added. However, we can note that tokens are easily retrieved as *ecel* 어절 are units strictly separated by spaces, at least in a normalised text following the official spacing rules of Korean (한글 맞춤법, which translates as 'Korean orthography rules' but actually affect both spelling *and* spacing). [Kim, 2013]. However, it is noteworthy that the application of these rules is considered "extremely complicated" and that according to a study on public school teachers' language use (based on their written productions) mentioned in Kim [2013, pp.71-72], the achievement rate on spelling and spacing is as low as 73.6% for teachers specialising in Korean language, and 55% for government employees. These scores question the use of the *ecel* 어절 as a linguistic unit, all the more so as, in practice, Korean is sometimes written without any spaces.[26] Furthermore, the minimal unit in syntactic annotations of Korean is not the *ecel* 어절 but the morpheme (described below).

(10) 기상청은 7일에는 전국적으로 눈이나 비가
kisangcheng-un chilil-ey-nun cenkwukcek-ulo nwun-ina pi-ka
weather.centre-TOP 7.day-LOC-TOP national-ADV snow-or rain-NOM
내릴 것이라고 말했다. [BRAA0163]
nayli-l kes-i-la-ko malhay-ss-ta.
fall-PRS thing-is-DECL-QUOT say-PST-DECL
'The weather center said that it will snow or rain on the whole country for 7 days.'

The morpheme as the smallest unit The annotated version of the Sejong Corpus is segmented in *ecel* 어절, with one *ecel* 어절 per line, as illustrated in Figure 3.5. However, each *ecel* 어절 is actually decomposed into morphemes, since morphosyntactic tagging is performed on morphemes in Korean, not on *ecel* 어절.

Strictly speaking, the smallest unit – or *token* – of a Korean corpus is therefore not the *ecel* 어절 but the morpheme, if we refer to the definition we gave in Section

[26] This practice happens often in informal written situations such as texting friends, or communicating on forums or social media.

3.4.3. Given that the *ecel* 어절 can be segmented into smaller parts, it is not an atom, and not a token.

As a matter of fact, in the experiments we conducted in Chapter 6, the sentences we used were represented as strings of morphemes without any particular *ecel* 어절 boundary. For example, if we use the sentence from Example 3.5, the sentence would be: `기상청/NNG 은/JX 7/SN 일/NNB 에/JKB 는/JX 전국/NNG 적/XSN 으로/JKB 눈/NNG 이나/JC 비/NNG 가/JKS 내리/VV ㄹ/ETM 것/NNB 이/VCP 라고/EC 말/NNG 하/XSV 았/EP 다/EF ./SF`.

This representation is not ambiguous at all, since the POS indicates the nature of the morpheme (whether it is independent, or if it is a prefix and should be attached to the following word, or a suffix and should be attached to the preceding word). This observation, along with the fact that Korean people do write without spaces in certain situations, as mentioned previously, shows that the segmentation in *ecel* 어절 is somehow superficial.

Multiword Expressions MWE are not identified as such in the Sejong Corpus. However, idioms were extracted in the SELK (Sejong Electronic Lexicon of Korean) which was built along with the Sejong Corpus. The SELK is composed of sub-dictionaries based on word categories [Shin, 2008] and among them, an idiom category. In addition, we can note that there are several studies on the extraction of other types of MWE such as light verb constructions [Kim et al., 2004] or verbal collocations [Lee et al., 2015] that were based at least partly on the Sejong Corpus.

In his description of Korean, Sohn [2013, p.416] states that the most productive type of compound nouns is the pattern `NOUN+NOUN`, with two general subtypes: modifier-head and head-head (appositive). He then gives examples of compound nouns illustrating this pattern, classified by their origin (native Korean, Sino-Korean, mixed of different origins or loanwords).

One of them is the native Korean word *kecis-mal* 거짓말 which is composed of *kecis* 거짓 ('false') and *mal* 말 ('word') and has the meaning of 'lie'. In the Sejong Corpus, *kecis-mal* 거짓말 is analysed as a single word because there is no space between the two nouns. However, this is not always the case as Korean compounds can span several *ecel* 어절. In this case, the compound word has to

be retrieved with a post-processing. Kim and Choi [1999, p.1085] implemented an incremental algorithm to extract compound nouns based on the following rules: "add to a single noun also the subsequent component if it is tagged (1) common noun, (2) foreign character sequence, (3) dash or noun derivative suffix only if a further noun follows". This post-processing would be necessary for compounds such as *kecis nwunmwul* 거짓 눈물 ('false' + 'tears', 'crocodile tears') *besu cenglyucang* 버스 정류장 ('bus stop') seen in Example 6c.

3.5.3 Annotation

```
BTAA0163-00000952 기상청은      기상청/NNG + 은/JX
BTAA0163-00000953 7일에는       7/SN + 일/NNB + 에/JKB + 는/JX
BTAA0163-00000954 전국적으로    전국/NNG + 적/XSN + 으로/JKB
BTAA0163-00000955 눈이나        눈/NNG + 이나/JC
BTAA0163-00000956 비가          비/NNG + 가/JKS
BTAA0163-00000957 내릴          내리/VV + ㄹ/ETM
BTAA0163-00000958 것이라고      것/NNB + 이/VCP + 라고/EC
BTAA0163-00000959 말했다.       말/NNG + 하/XSV + 았/EP + 다/EF + ./SF
```

Figure 3.5: Example of POS-tagged sentence from the Sejong written Corpus [BTAA0163]

```
BSAA0163-00000952 기상청은      기상청/NNG + 은/JX
BSAA0163-00000953 7일에는       7/SN + 일/NNB + 에/JKB + 는/JX
BSAA0163-00000954 전국적으로    전국__03/NNG + 적/XSN + 으로/JKB
BSAA0163-00000955 눈이나        눈__04/NNG + 이나/JC
BSAA0163-00000956 비가          비__01/NNG + 가/JKS
BSAA0163-00000957 내릴          내리/VV +ㄹ/ETM
BSAA0163-00000958 것이라고      것/NNB + 이/VCP + 라고/EC
BSAA0163-00000959 말했다.       말__01/NNG + 하/XSV + 았/EP + 다/EF + ./SF
```

Figure 3.6: Example of morphologically tagged and disambiguated sentence from the Sejong Morph Sense Tagged written Corpus [BSAA0163]

In Figures 3.7 and 3.8, constituents have been coloured solely to enhance the readability of the imbrications of constituents both in the parsed sentence and in its tree. Items in blue are the leaves of the tree, i.e., the last nodes corresponding

to tokens, while orange items are intermediate constituents and the red item is the maximal constituent that is smaller than the sentence.

(11) 그 애제자는 이번에 모 음대에
ku ayceyca-nun iben-ey mo umday-ey
this favourite-TOP this.time-LOC X College.of.Music-LOC
들어갔다. [BGAA0164]
tuleka-ss-ta.
enter-PST-DECL
'This time, the favourite student entered the College of Music of University X.'

```
(S    (NP_SBJ    (DP 그/MM)
                 (NP_SBJ 애/NNG + 제자/NNG + 는/JX))
      (VP   (NP_AJT 이번/NNG + 에/JKB)
            (VP   (NP_AJT     (DP 모/MM)
                              (NP_AJT 음대/NNG + 에/JKB))
                  (VP 들어가/VV + 았/EP + 다/EF + ./SF))))
```

Figure 3.7: Example of parsed sentence from the Sejong written Corpus

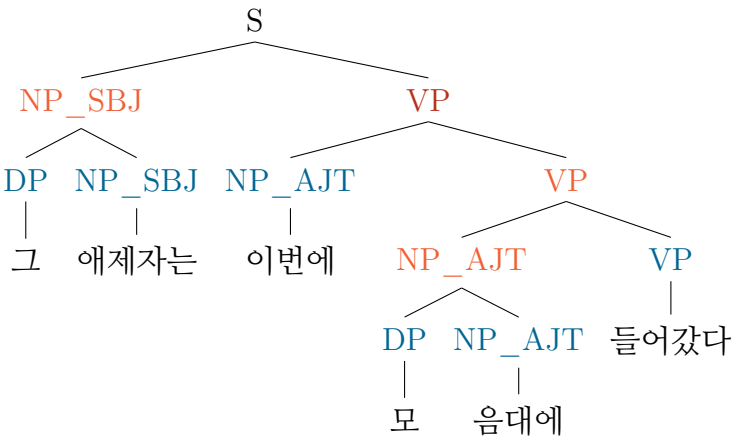


Figure 3.8: Example of parsed sentence from the Sejong written Corpus

3.6 Conclusion

A corpus is not simply a collection of raw texts, but a resource that can be linguistically enriched with different layers of annotations. Indeed, a corpus is rarely

constituted to remain raw, and multiple preprocessings and processings are often considered as integral parts of the constitution of the corpus, and not just as additional steps.

We have seen that those layers may be of different natures (morphological, morphosyntactic, syntactic, semantic, pragmatic, prosodic etc.) depending on the purpose of the constitution of the corpus and its uses. Given that a corpus may have as many layers of annotations as desired, the possibilities to study the correlations between the different annotations are considerable. Some corpora, such as Rhapsodie[27], are constituted specifically for that purpose: Rhapsodie is a syntactic and prosodic treebank of spoken French, which objectives are "to define rich, explicit, and reproducible schemes for the annotation of prosody and syntax in different genres [...] in order to study the prosody/syntax/discourse interface in spoken French, and their roles in the segmentation of speech into discourse units" [Lacheret et al., 2014, p.295].

One of the main contributions of corpus in linguistics and applied linguistics is the study of patterns of words which co-occur significantly:

> As is often the case in linguistics, different terminology has been used over the years to describe the phenomena of multi-word vocabulary or chunks. Labels include 'lexical phrases' (Nattinger and DeCarrico 1992), 'prefabricated patterns' (Hakuta 1974) 'routine formulae' (Coulmas 1979), 'formulaic sequences' (Wray 2000, 2002; Schmitt 2004), 'lexicalized stems' (Pawley and Syder 1983), 'chunks' (De Cock 2000), as well as the more conventionally understood labels such as '(restricted) collocations', 'fixed expressions', 'multi-word units/expressions', 'idioms', etc. Whatever the terminology, all seem to agree that multi-word phenomena are a fundamental feature of language use. [O'Donnell et al., 2012, p.63]

The study of such phenomena is typically facilitated by corpus exploration tools, which often integrate a function that automatically computes co-occurrences. Given that corpora are not investigated manually anymore, due to their size and

[27] http://www.projet-rhapsodie.fr/

to the fact that it is a tedious task, nowadays, the possibilities offered by annotated corpora are directly linked to the functions that are implemented in corpus exploration tools. An overview of these possibilities and functions are described in Chapter 4.

Chapter 4

Overview of Corpus Exploration Tools

4.1 Introduction

A corpus is a collection of texts (either from written resources or from transcribed speech data) considered as a representative sample of a given variety of language or genre. We demonstrated in Chapter 3 that whether the investigator adopts a corpus-based approach, testing preformed hypotheses against authentic data, or a corpus-driven approach, inducing hypotheses from observed regularities or exceptions, corpora are an invaluable resource from which examples of *real language* use can be extracted not only to support but also to refute linguistic arguments.

Corpus as a tool Incidentally, a corpus is also often considered as a *tool* or an *instrument* in itself. Anthony [2013] points out this peculiar paradox by contrasting quotations from well-known authors such as the two following quotations from Susan Hunston stating that

> "corpora have been likened to the invention of telescopes in the history of astronomy" [Hunston, 2002, p.20]

but also that

> "[strictly speaking,] a corpus by itself can do nothing at all, being nothing more than a store of used language" [Hunston, 2002, p.3]

and argues that a distinction should be made between the corpus as a linguistic *resource* and the *tools* that are essential to explore corpora. Extending the comparison with astronomy, Laurence Anthony mentions that observing a planet through the human eye or through a reflector telescope gives very different perspectives on the same object, and argues that in the same manner, researchers should be aware that corpora are not unchanging resources and that what can be inferred from them strongly depends on the possibilities that the tool offers.

Indeed it is theoretically possible to explore a corpus just by looking through it manually as it had been done before corpora were electronically stored. However, considering that the scale has changed and that even 'small corpora' contain at least several thousand words (precisely between 20,000 and 200,000 words according to Aston [1997]), exploring a corpus manually is a tedious and lengthy work, not to mention that humans are neither infallible nor inexhaustible.

Tools as more than tools While corpora cannot be explored without a proper tool, corpus exploration tools obviously cannot be dissociated from their object and purpose either. Even though corpora came first, we are currently in a situation of natural mutual dependency rather than a hierarchical relation and we should not consider corpora as simple tools that could be used interchangeably. For anyone working in corpus linguistics and in particular for software developers, a corpus exploration tool is not simply an observation tool: the development of such tool necessarily involves implicit theoretical linguistic choices.

As a matter of fact, analyses resulting from the output may vary highly depending on which tool was used, or more precisely, depending on both the underlying linguistic choices (e.g what is considered as a *word*?) and the features (does the tool allow counting words, tokens and/or lemmas?). Anthony therefore insists that refocusing on corpus exploration tools could solve many problems originally thought to be linked to corpora, for instance by choosing carefully which tool to use according to the purpose of the study, to the background of the tool and to its parameters. A striking example is the utility of annotations in a corpus: even if annotations are of no use for a study, it could still be perfectly suitable to use an annotated corpus. For this particular study it would indeed be relevant to choose a tool with an option hiding the implemented tags.

In this chapter, we will first provide a brief historical overview of software tools originated from Corpus Linguistics before introducing a range of corpus exploration tools currently used in different fields for different purposes, and highlighting the specificity of each. While the first sections give a general overview of the tools and functions made by and for specialists, in Section 4.4, we focus on adaptations made with the aim of opening corpus exploration tools to a broader public, including non-specialists such as language teachers or students.

4.2 Corpus Exploration Tools through History

Tony McEnery and Andrew Hardie identified four different generations of corpus exploration tools from the onset of corpus linguistics in the mid-20th century to recent years, and described their respective strengths and weaknesses in McEnery and Hardie [2012, pp.37-48]. We summarise this thorough description as follows:

The first generation (1960's-1970's) could only run on mainframe computers (i.e large general-purpose computers supporting numerous peripherals or subordinate computer[1]) and was limited to the processing of corpora of rather small size, exclusively in English, and using the ASCII[2] character set. Because ASCII was originally based on the English alphabet, an obvious limitation to the use of ASCII is the processing of non-English texts, as ASCII does not allow to display any letter with diacritics such as the French accents (acute and grave accents in *éphémère* 'ephemeral' or the circumflex in *île* 'island'), vowels with umlaut from German (as in *Bürger* 'citizen'), not to mention the display of completely different writing systems from Arabic, Hebrew, Chinese, Korean or the Khmer script

[1] Definition of a 'mainframe' retrieved on the Oxford English Dictionary (http://www.oed.com) on 23rd March 2017.

[2] American Standard Code for Information Interchange, created in 1963 by what was formerly known as the American Standard Association (but today called American National Standards Institute). The original ASCII character set comprises lowercase and uppercase letters of the English alphabet, numbers from 0 to 9, punctuation symbols, and control characters, i.e non-printable characters originated from typewriter systems, such as the 'carriage return' which moves the cursor (formerly the carriage) to the left-hand side of the paper and which together with the line feed, allows to type on a new line. Characters from other alphabets appear only in extensions developed from 1981.

to name a few. In the case of a few non-available characters in an alphabet, the problem could be solved by simply replacing them with characters from ASCII or by adding an extra character if the closest character in ASCII does exist already in the language (for instance, adding an <e> after vowels with umlaut to differentiate *schon* 'already' from *schön* 'beautiful' by writing schoen).

According to McEnery and Hardie, the main drawback of tools from the first generation is the fact that "replicability was difficult to achieve" for several reasons: tools would actually come in separated packages with a unique function, they would only run on one type of mainframe computer and not others, tools for manipulating corpora were not shared resulting in a waste, as many tools would 'reinvent the wheel'. As for corpora, they would also display many *ad hoc* conventions (especially in annotations) because standards were not yet clearly developed.

The second generation (1970's-1990's) is marked by the development of personal computers which led to the spread of new *concordancers*[3] which were made specifically to be compatible with personal computers and widely distributed. However, personal computers are significantly less powerful than mainframe computers. This inevitably resulted in an overall diminished performance of corpus exploration tools. Firstly, the new generation of concordancers had less functions than those from the first generation (mostly limited to KWIC (KeyWord In Context) concordancing, sorting the left and right contexts alphabetically, and basic descriptive statistics) and secondly, they could not search through corpora as large as before. McEnery and Hardie give the example of the Longman Mini-Concordancer [Chandler and Tribble, 1989] which would run out of memory when searching through a few tens of thousands of words, while in the same period, "tools running on mainframes were able to deal with corpora of a million words or more". Moreover, standards had still not found a satisfying ground so it was still complicated not only to properly read corpora with special characters but also to deal with the various conventions of annotations inherited from previous *ad hoc* conventions in different corpora.

Despite these drawbacks, the second generation had a positive impact on corpus linguistics in that this "democratising effect" allowed much more studies (even a

[3]The concepts of concordancers and concordancing are defined below.

"boom" from the late 1980s), and ended the waste of energy in reinventing the wheel that we mentioned previously, so that this energy could ideally be redirected into the production of better tools.

The third generation (late 1990's up to now) are the most common tools available today. This category includes widely used WordSmith [Scott, 2017] developed by Mike Scott since 1996[4] and freeware toolkit AntConc [Anthony, 2014] built by Laurence Anthony.[5] Contrary to tools from the previous generation, this new generation is characterised by:

- multi-language support thanks to the advent of *Unicode* – a new character encoding specifically created to be "universal (addressing the needs of world languages), uniform (fixed-width codes for efficient access), and unique ([a given] bit sequence has only one interpretation into character codes)"[6];
- more user-friendly interfaces – a significant characteristic considering the increasing number and variety of users, both linguists and non-specialists of language (see Section 4.4);
- and the possibility to work on larger corpora.

Anthony [2013, p.152] qualifies this third and last point as he considers that "the biggest limitation with third generation tools is that they struggle to handle very large corpora of over 100 million words" while McEnery and Hardie are more concerned with the striking similarities between tools from this generation in terms of functions and the lack of innovative functions that expand the possibilities of corpus searches:

> "[...] if the toolset does not expand, then neither will the range of research questions that may reasonably be addressed using a corpus." [McEnery and Hardie, 2012, p.42]

[4]The latest version can be found at http://www.lexically.net/wordsmith/ (version 7 in 2016)

[5]Freely available on http://www.laurenceanthony.net/software/antconc/ (release 3.4.3, 2014)

[6]Those are the explicit goals fixed by the Unicode Consortium in the summary narrative of their history, on http://www.unicode.org/history/summary.html retrieved on 1st April 2017

Indeed, all of them seem to revolve around four main functions (concordance, frequency lists, collocations, keyword analysis, see description in 3.2) with very little specialisation. But authors are also aware that for an innovative functionality to be added, two elements are necessary: an agreement on its utility and a large audience to meet. Otherwise, given that most of the corpus exploration tools are mainly developed by one contributor, it is not worthwhile for the latter to invest the time and effort to implement this new functionality.

The fourth generation (from the mid 2000's up to now) tackles problems met by the third generation: they can handle very large corpora (over 100 millions words) with fast processing and they solve the rising copyright problem. These improvements are enabled by the new architecture of the tool, based on a pre-indexing system hosted on an external server. For third generation tools, the speed of exploration is limited by the power of personal computers. On the other hand, for fourth generation tools, both data and calculations are localised on an external server, which is sometimes as powerful as mainframes. The copyright problem refers to the fact that some authors are reluctant to share their corpora freely, considering the potentially staggering amount of time and effort they dedicated into the construction, the edition, the annotation(s), as well as the correction of their corpora in case of an (semi-)automatically annotated corpus. This architecture also allows only restricted access to the licensed corpus in the form of snippets[7] (although processing is still done on the whole corpus) instead of providing a link to download the corpus. However, the fact that the corpus needs to be processed, pre-indexed and then stored on an external server is also what gives rise to the fourth generation tools' limitations. Also, McEnery and Hardie [2012, pp.59-60] add that

> "inevitably, a web-based concordancer will never allow the full range of analyses that a technically savvy researcher could accomplish with a copy of the corpus on their own computer"

inferring that fourth generation tools might be more interesting for novice users than for specialists who would be limited by not having the possibility to manip-

[7] Small portions of texts, similar to what Google shows on the results page.

ulate the corpus as they like.

Considering the fact that third and fourth generation tools have complementary strenghts and weaknesses, it is not surprising that they still coexist and that they both are popular among researchers. As a natural consequence, some of the tools that we use or cite in this dissertation are from the third (e.g AntConc, Le Trameur, Lexico) or the fourth generation (e.g The Lexicoscope, the corpus.byu.edu web interface, or the KKMA web interface for Korean) of corpus exploration tools.

concordancing

All tools mentioned above in this historical account are based on *concordancing*. In other words, their main function is to display words of a text in their immediate context, as defined by McEnery and Hardie [2012, p.35]:

> "A concordancer allows us to search a corpus and retrieve from it a specific sequence of characters of any length – perhaps a word, part of a word, or a phrase. This is then displayed, typically in one-example-per-line format, as an output where the context before and after each example can be clearly seen."

In addition, apart from certain first-generation tools, they would also provide other functions such as computing statistical information on *keywords* or the possibility to search *n-grams*, a notion that we define in Section 4.3.2.

In parallel with the development of concordancers in the 'anglosphere', French researchers focused on statistical analysis. Differences in objectives and approaches resulting in the development of different sets of features are not discussed in this dissertation. We invite the reader to refer to Poudat and Landragin [2017] (in French) where these differences are mentioned.

4.3 Querying Possibilities

When we think of a way to search through a corpus, we automatically and intuitively think of using *words*. Such queries seem to be reliable enough for millions

of people to use *keywords* on a daily basis when using a web search engine. In this case however, users want to retrieve documents matching their query, that is to say documents containing the keywords they typed in the search field (usually a single-line search box also called a *search bar*) usually in no particular order[8] and wherever those keywords occur in the document. Documents are therefore said to be 'bags of words' as if a text was a mere bag where words have been cut and mixed. In this case, there is no way to retrieve sentences or even phrases. Words appear to be independent and are simply counted as such.

In this section, we first introduce query systems that allow to retrieve *whole documents* based on *metadata* before describing query systems matching *occurrences* of a query *within* a corpus. We will see that to do so, corpora necessarily underwent a certain number of processings, according to the type of corpora, the complexity of annotation desired and the purpose for which they were built. Those processings are determining because query possibilities entirely depend on them: for example, it would be impossible to make a query on a syntactic construction without at least a morphosyntactic annotation layer (see 4.3.3). The deeper we get into layers of annotations, the more complex the query inevitably becomes and the more specialised the user has to be.

4.3.1 Metadata-based Queries

Before texts were searchable, the indexation of documents was solely based on metadata. On certain systems such as libraries' query systems, searching through the text is still not possible: typing the word "`corpus`" in the search bar does not retrieve every document with at least one occurrence of the word "corpus" in its textual content but only documents containing this word in the metadata, i.e in the title, the authors, the editors, the publisher, the publication date, the collection, the language in which the document is written, and the topics in the form of keywords to name the most common metadata in book indexation.

Every corpus exploration tool providing access to a corpus is bound to give the possibility to access the metadata as well, at least for the corpus as a whole if not

[8]Web search engines generally allow a more precise search looking for the keywords in the strict order they were typed in, either in the advanced search options or using a particular syntax like double-quotes for Google. Such queries are called *n-gram* (see below).

for each sample. Contrary to libraries' query systems, corpus exploration systems do not allow the metadata to be searched directly – only the textual content of the document is. When the metadata are considered important for the exploration of the corpus, they are usually separated from the search within the corpus as in Frantext, a "textual database" for French comprising both a corpus and an online corpus exploration tool. Frantext is not the largest corpus of French but has a unique characteristic as it is a diachronic corpus of more than 5000 texts ranging from the tenth to the twenty-first century and almost 300 million words.[9]

As shown in the following step-by-step procedure to use Frantext, we are first prompted to do a search within the metadata before being asked to type in the keywords that we are interested in within the corpus:

1. Selection of the text(s) to explore;

2. Search in the metadata of selected texts[10] – precisely either in the name of the author, the title of the document, the literary genre, the publication date or the shelf number – which, if used, narrows down the number of texts to explore and thus helps refine the selection of the text(s);

3. Search in the texts using either keywords or a powerful query language specific to Frantext;

4. Configuration of the visualisation parameters of the results, notably by choosing the sorting option "sort by ascending/descending order chronologically" instead of "sort by ascending/descending alphabetical order";

5. Result analysis.

This highlight on metadata with a search in the metadata as a separate step is a specificity to Frantext due to its diachronic nature. Nowadays, most corpus exploration tools use metadata as options to narrow the scope of the query to certain samples. Widely-used metadata are:

[9] Figures are quoted from Frantext's official website's main page: http://www.frantext.fr/

[10] Called "Recherche dans un élément bibliographique" in French, literally "search in a bibliographic item"; the interface has no English version.

- the date of publication, particularly exploited in Frantext, as well as in Google Ngram Viewer, for example (see Section 4.3.2));
- the genre or subgenre of the sample: based on the medium of communication, either "written", "spoken" or "signed", or on more specific genres such as "fiction", "magazine", "newspaper" and "academic" for the COCA (see Figure 4.4), "discourse" and "ritual/religious texts" for Rhapsodie (described in 3.6);
- other metadata specific to the corpus: "interactivity", "social context", "event structure", "channel", "planning type", "quality" for Rhapsodie because it was specifically built to study the interaction between prosody and syntax in spoken French, or in a completely difference perspective, "year of birth", "study of French (in months)", "stay in France (in months)", "sex" (of the speaker(s)) to retrieve speech samples of learners of French in IPFC (InterPhonologie du Français Contemporain).

4.3.2 Word-based Queries

The access to word-based queries is granted by the segmentation into words, or rather into *tokens*, a process thus called *tokenisation*. This process is far from trivial and should not be overlooked as tokens serve as a foundation for every additional annotation layer. For more information on this topic, see the discussion on the difference between words and tokens, as well as on the role of tokens in NLP in Section 3.4.3.

As for this section, we describe gradually more complex types of queries, all of them using combination of words but in different ways: n-grams, skipgrams and phraseological units.

N-grams Contiguous sequences of n items (in this case, words or lemmas) are called *n-grams*. Common n-grams are bi-grams, tri-grams, up to 5-grams and allow users to look for word clusters, possibly including *collocations*, words that occur regularly with some words in a statistically significant manner.

A notable implementation of n-grams is Google's Ngram Viewer, an online search engine allowing any user to compare the frequency of several "phrases" (n-grams) in a corpus of books within a span specified by the user – from a single

particular year up to two hundred years, between 1800 and 2008. Ngram Viewer is available for eight languages, namely Chinese (simplified), French, German, Hebrew, Italian, Russian, Spanish and English (with distinct corpora for American English, British English and English Fiction, as well as a general corpus including the three variants of English) and relies on the Google Books Ngram Corpus, which second edition contains over 8 million books, or "6% of all books ever published" [Lin et al., 2012, p.170][11] and over 500 billion[12] words [Michel et al., 2011, p.177]. Critics mainly focus on the representativeness of the corpus (scientific vs. popular books for instance in Pechenick et al. [2015]) and on the quality of the corpora: given the large amount of data, each corpus had to be processed automatically from OCR[13] to syntactic annotations, resulting in many potential inaccuracies [Hamamura and Xu, 2015]. Despite these drawbacks, Google Ngram Viewer is an interesting tool for the unique overview it gives of not just historical variations but also cultural and social changes in language use (in books), providing that users are given caveats about the limitations of the corpus.

N-grams of words or lemmas are relatively easy to use for novice or non-specialists users, as searching through a corpus using n-grams only implies to type the desired chunk of words into a textual field commonly called a search box and then looking through the concordance to pick up recurrent words used with this chunk (collocation) or the context(s) in which it is mostly used. This simplicity has a limit: studying only strictly contiguous sequences of words is rich because they allow to retrieve interesting collocations or word clusters (see definitions in Chapter 3) but it fails to take into account certain constructions, as simple as associated words with modifiers that are not at their edges but in between. Cheng

[11] From over 10,000 publishers and authors from more than 100 countries participating and counts among its partners seven international libraries (Oxford University (UK), University of Complutense of Madrid (Spain), the National Library of Catalonia (Spain), University Library of Lausanne (Switzerland), Ghent University (Belgium) and Keio University (Japan)), cf. https://books.google.com/intl/en/googlebooks/about/history.html

[12] Precisely, 361 billion words for English, around 45 billion words for French and Spanish, around 37 billion words for German, around 35 billion words for Russian, around 13 billion words for Chinese and finally around 2 billion words for Hebrew.

[13] Optical Character Recognition, the automatic digitisation or conversion of images of typed, handwritten or printed text into machine-encoded text. The quality of the output highly depends on the quality of the image (resolution, contrast between characters and the background, sharpness of font, lisibility of handwriting etc.).

et al. [2006] illustrate this limitation by giving the example of two sequences, "a lot of *local* people" and "a lot of *different* people", which would not be retrieved along with "a lot of people" if the latter was used as a query. The difference is only materialised by a single modifier and one might consider the three sequences to be instances of the same pattern (despite the minor difference in surface), and therefore be interested in retrieving all of them.

Skip-grams To tackle this problem and handle what Cheng calls "constituency variation" efficiently, another type of pattern has been developed in the early 2000s to retrieve not only contiguous sequences of words – or strict n-grams – (`AB`) but also non-contiguous sequences (`ACB`). Those sequences where undesired items (`C`) between the main components (`A` and `B`) are *skipped* are called *skip-grams* or *gapped n-grams*, as well as "long distance n-grams" in language modeling [Huang et al., 1993].

It is interesting to note that, as a matter of fact, skip-grams are easily represented with a proper query containing a 'wildcard', usually marked by the symbol `*` as in the CPQ Syntax (used in Google Ngram Viewer or in TXM) or `.*?` in regular expressions. For instance, the query `A .*? B` allows to skip a high number of items between `A` and `B`.[14] The use of wildcards, let alone of regular expressions, is not self-evident but mastering it allows much more possibilities and complexity than skip-grams. For this reason and also because making wildcards or regular expressions available actually costs less effort than implementing skip-grams as an alternative option to n-grams, the majority of concordancers do not allow skip-grams, strictly speaking, but usually have users learn how to use the powerful wildcards instead.

Phraseological units The size of clusters retrieved by skip-grams are usually limited to a frame of 11 words (up to four words skipped on both sides of a given word due to computational cost as well as to the assumption that this frame is sufficient for relevant sequences (see Wilks [2005] cited by Greaves and Warren [2007]). However, even rare, associated words separated by a high number of words

[14] Precisely, all items may be skipped except for a carriage return character, which means that all items are skipped within the same line.

could still be relevant and interesting. Some researchers went a little further than skip-grams models by building tools allowing to look at **larger word clusters**. That is the case of Chris Greaves who built the "phraseological search engine" ConcGram© [Greaves, 2009] and Olivier Kraif who developed The Lexicoscope [Kraif and Diwersy, 2012]. In their case, they even allow to take into account **constituency variation**, in other words, they allow to match words from different constituents[15].

Both systems take as inputs several words[16] called *pivots*, either directly input by the user or selected through iterative associations. In the latter case, the tool takes a first pivot (or the first two for ConcGram©) and searches for words with which it has the strongest co-occurrence rates; these words are then used in turn as pivots and so forth, up to four additional pivots for both tools (see Greaves and Warren [2007, p.291] for a description of the automatic construction of concgrams and Kraif and Diwersy [2012, p.405] for the French tool).

On top of that, the two phraseological search engines also consider **positional variation** in that they both match `AB` and `BA` (e.g "speaking English" and "English speaking") when retrieving n-grams of words given as inputs by users. Both constituency variation and positional variation have to be handled to retrieve sequences like "world city of Asia" and "Asia's world city" with a single query. According to Cheng et al. [2006, p.416], using ConcGram©, those sequences can indeed be retrieved together with the 3-word pattern `asia world city`. Similarly, The Lexicoscope allows to skip words and does not take into account the order of pivots (although it does in the traditional concordancing mode): for example, the pattern `like woman` retrieves both the sequences "ravished like the Sabine women" and "like those of the women who play cellos". In this case, we can also note that pivot word `woman` was used as a lemma and not a wordform for it also retrieved the plural form "women".

[15] In both cases, the corpora tied to the exploration tool were preprocessed with a constituency parser, see 3 for more information on this processing. Constituency variation is the only property that needs annotations and cannot rely solely on words. However, we chose to describe phraseological units in word-based queries as the queries themselves do not contain any constituency tag.

[16] By *words*, we are referring to inflected forms of a word, as well as to the corresponding lemma. It is up to the user to choose whether morphological variations should be considered or not.

Eventually, ConcGram© also has a major advantage for beginners in corpus linguistics as it automatically conducts searches of statistically relevant word associations of 2 to 5 words called concgrams within a frame of any size for a given corpus, sparing users the tricky choice of a relevant frame. This **fully automated search** also allows a **corpus-driven approach** in that it enables users to find new phraseological patterns, and not simply "a more extensive description of known patterns of collocation and their meanings" [Greaves and Warren, 2007, p.290], a quality that we also value and display for our own system, described in Chapter 5. We did not have the opportunity to test this particular function but concgrams have been used to investigate a certain number of texts in the literature and establish their "phraseological profiles" (see for example [Hou, 2016]).

This overview of current tools used in corpus linguistics gives a wide range of the possibilities offered to explore the luxuriant yet intricate lexical network of a language, from bi-grams to phraseological units or collocations. The studies of word combinations that we mentioned, and especially that of phraseological units, give a good glance at certain constructions. For example, we could compare the two genitive constructions in English (one with the preposition "of" as is "word city of Asia" and the other with the "'s" as in "Asia's world city") using the 3-word patterns on ConcGram©. However, this method does not allow users to study the use of the genitive construction in general but only in a particular case where the words specified appear significantly in the same context. We could then simply look directly at concordances of the two morphemes that we are interested in: `of` and `'s`. Without any morphosyntactic information, the latter can be obviously mistaken for the contraction of the verbal forms "is" and "has" and for the former, a quick search in the COCA reveals different relations introduced by the genitive case (among others, possession in "the ability *of* individuals", origination or reference in 'University *of* Central Florida", composition in "a group *of* low-income parents"; see Rappaport [2004] and Jensen and Vikner [2004] cited in Kardkovács and Tikk [2007] for the description of the relations), as well as possible other functions of the preposition in "possible because *of* opportunities" where "of" could be analysed as a part of the component "because of". Incidentally, the query itself relies on a

combination of wordforms (or lemmas), not on syntax. To use a syntactic pattern directly as a query, one needs to rely at least partly on syntactic annotations.

4.3.3 Annotation-based Queries

The need for annotation In all the above-mentioned cases, queries are solely based on words. This is sufficient for a large number of queries but not for studying the use of certain syntactic morphemes or constructions, even with complex textual queries usually involving regular expressions. For example, the study of the **progressive verbal form in English** seems easy as this construction relies on a particular morpheme: *-ing*. However, a query like `*ing` would retrieve not only the progressive forms *going*, *being* and *doing* but also words like *something*, *during* or *according*. In the case of the present perfect progressive, a query such as `been *ing` would be efficient enough, but for the present progressive, `is *ing` retrieves the forms *is something* and *is nothing*.[17] Such errors are easily avoided if we rely to some extent on syntactic (or at least morphosyntactic) annotations for queries on syntactic constructions; just like queries on synonyms (e.g *hate* and *detest*), near-synonyms (e.g verbs *hate* and *loathe*, or nouns *hate* and *aversion*) or from a similar class of word (e.g words conveying negative emotions) cannot be used if the corpus is not enriched with semantic annotations.

Comparison of two constructions We can consider for example that "speaking English" is an instance of the pattern `V-ing_ADJ NOUN`[18] and that "English speaking" is an instance of `ADJ V-ing_VERB`. Working with these patterns that make use of parts of speech – instead of a query with lexical words only – allows to look at the contexts and **differences of usage between the inflected form `V-ing` when used as a verb and as an adjective**. Such queries are allowed in the advanced research mode of The Lexicoscope among other tools.

[17] Tested on the COCA using the corpus.byu.edu interface (henceforth referred to as the BYU Corpora interface): *is something* is the 3rd most frequent match of `is *ing` while *is nothing* is the sixth.

[18] We use `V-ing` to represent an inflected verb formed by the verb stem with the suffix *-ing* for the sake of the example. However, please note that this inflection is not necessarily encoded in this way in corpus annotations, and not even necessarily encoded at all.

Figures 4.1 and 4.2 give us interesting examples of output retrieved using morphosyntactic tags as well as an interesting insight of the possibilities offered by The Lexicoscope. In both cases, we used a query involving two 'pivots': first, the form *speaking* either annotated as an adjective or as a verb, and second, either a noun as the object of *speaking* (as in "speaking English") or a co-occurring adjective (as in "English speaking").[19]

Left context	Pivot	Right context
Hiram was allowed to go on	speaking	his **mind** [...]
careful to avoid	speaking	her **name**.
it was the English who were incapable of	speaking	foreign **languages**.

Figure 4.1: Example of output using The Lexicoscope with "speaking" used as a verb with a noun as object

Figure 4.2 first reminds us that The Lexicoscope allows both position variation and constituency variation. The second example here matches the adjective *familiar* that is not only in another constituent but also belongs to the right context of *speaking* while the two other examples match an adjective that is on the left.

Left context	Pivot	Right context
the biggest director in the whole **English**	speaking	theatre [...]
It was a woman	speaking	– someone **familiar**, someone she knew.
it would be good to have a local **English-**	speaking	person with you [...]

Figure 4.2: Example of output using The Lexicoscope with "speaking" used as an adjective with a noun adjective

While certain forms such as the progressive form can be systematically identified and retrieved with a specific morpheme, others are not as transparent. In those

[19]These queries were performed on the English Literature Corpus available on The Lexicoscope's platform and which covers 273 texts with a total of almost 38 million words.

latter cases, it is necessary to go beyond wordforms. It is impossible for example to study the **preterit in English** without using at least POS (part-of-speech) tags (see the definition in Section 3.4.4). Even so, a query on a token with *-ed* as ending and tagged as a verb could retrieve not all preterit forms but only regular ones. To retrieve irregular verbs, the corpus needs to have a specific annotation for the preterit, such as `VVD` for past tense of lexical verbs in the CLAWS7 tagset, used in most reference corpora for English, including the British National Corpus and the COCA.

The necessity to use tags is not only bound to specific morphosyntactic forms. To describe and retrieve syntactic constructions, there is no other possibility than to resort to morphosyntactic tags directly as well. As explained in Wang [2016], the matching of two segments such as "*the person who is sleeping*" and "*the jury which was locked up*" which have no lexical units in common but which share the same syntactic structure can only be achieved with a pattern like "`DET NOUN WH-PRO AUX VERB`".[20] This type of query is commonly used in linguistics, but producing such patterns requires users not only to know the tagset of the corpus but also, and maybe more importantly, to be able to associate a word with the right part-of-speech. Regular expressions are a good means to broaden the range of query possibilities but at the cost of a more advanced training and the adaptation to a whole other level of abstraction.

4.3.4 In Information Retrieval

Computing the similarity between two objects is a common part of numerous tasks related to Information Retrieval.

Figure 4.3, extracted from Amini and Gaussier [2013].

A document can be representated by *term vectors*:

$$D = (t_i, t_j, ..., t_p)$$

[20] These part-of-speech tags do not belong to any specific tagset. They are purposely generic and we decided to use the tag `VERB` for the sake of illustrating the fact that the two segments are different in terms of grammatical categories (auxiliary and -ing verb on the one hand, verb and preposition on the other hand) but are *similar* in the sense that they are both verbal phrases.

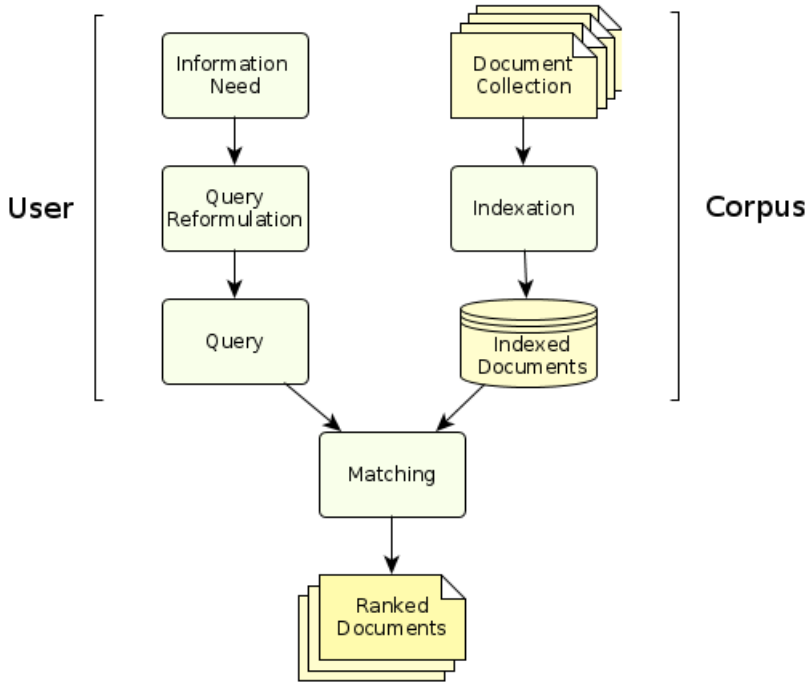


Figure 4.3: Flowchart of the different steps of Information Retrieval

where each t_k is a term[21] in document D. To search this document, the query also has to be represented in vector form:

$$Q = (q_a, q_b, ..., q_r)$$

or in the form of a Boolean statement:

$$Q = (q_a \ and \ q_b) \ or \ (q_c \ and \ q_d \ and \ ...) \ or \ ...$$

These representations in vectors are typically "bag-of-words" representations, briefly described in Section 5.2.

Query-document similarity Considering the fact that the main task of IR is to retrieve a relevant document based on a query (exactly what search engines are

[21] In the sense of 'token', i.e., a segmentation unit that does not necessarily correspond to a word. See Section 3.4.3.

made for), it is not surprising to find that computing query-document similarity has been, and is still, a major research problem.

Query-query similarity With the rising popularity of Social Networking Services (SNS) such as Twitter, and the growing importance of opinion mining, similarity measures have to adapt to a new format of text: short-text messages. Due to the small size of both the query and the candidates, new strategies were required.

Query reformulation Query reformulation is used both in query-document and in query-query similarity searches when the original query is not sufficient to retrieve all relevant documents or to match relevant queries. It may consists in substituting words from the original query or in expanding (i.e., adding new words to) the query, with alternative words like synonyms. In the case of misspellings, query substitution might be more suitable but expansion is useful safer.

4.4 Current Effort to Adapt to Non-Specialists

Annotated corpora are rich resources that have attracted the attention of many specialists (of other disciplines, or of linguists who had never been trained in corpus linguistics) as well as non-specialists (in our case, language learners but anyone interested in language is potentially concerned). A solid evidence of this phenomenon lies in the increasing number of training programmes proposed by consortiums[22], or as part of workshops and summer schools both in conjunction with international conferences and independently[23]. Even though those courses are offered to non-specialists, not all of them have the time, the money or the will to receive a full training in corpus linguistics in order to access corpora. This section addresses the crux of the matter of the accessibility of corpus exploration tools to non-specialists

[22]See https://groupes.renater.fr/wiki/txm-users/public/ateliers_txm for an updated calendar of TXM training workshops for example.

[23]See Corpus Linguistics Summer School, organised with the Corpus Linguistics Conference (CL2017) by the University of Birmingham at http://www.birmingham.ac.uk/research/activity/corpus/events/2017/summer-school-2017.aspx and the annual three-day Summer School in English Corpus Linguistics, organised for the fifth time at University College London: http://www.ucl.ac.uk/english-usage/summer-school/

by introducing two types of adaptation: the simplification of tools' interface and the simplification of the language used for the query.

4.4.1 Simplification of the Interface

We have mentioned in Wang [2016] that the complexity of the interface was a major cause of concern from non-specialist users such as language learners, notably quoting from Boulton [2012]. Regarding this growing issue, efforts are currently made towards the simplification of interfaces. In all those initiatives, we identify three main ideas:

1. interfaces must be user-friendly;
2. interfaces must be minimalist;
3. the number of steps involving the user must be kept to a minimum.

In Section 4.2, we have shown that corpus exploration tools first came with textual interfaces only, but graphical interfaces were rapidly implemented. Nowadays, apart from linguists who have been trained in Natural Language Processing and are used to command-lines interfaces, most users would prefer a graphical interface, thought to be more *user-friendly* because the possibilities of interaction are shown.

The concept of affordance The first two ideas of user-friendliness and minimalism combined together contribute to facilitate the perception of the actions that users can actually do with the tool. This has long been explored in psychology in the concept of *affordance*. The word "affordance" was coined by psychologist James J. Gibson who defined it as follows: "The affordance of anything is a specific combination of the properties of its substance and its surfaces taken with reference to an animal." [Gibson, 1977]. This definition shows the importance of both the 'thing' and the operator[24] (or animal in the definition), as what is perceived from all the properties of the 'thing' depends completely on the 'operator': Gibson gives

affordance

[24]The term 'operator' used here is meant to be neutral. An 'operator' is simply an entity that is capable of acting on something, being an animal or a human being.

the example of a non-rigid surface such as a stream, which has the intrinsic physical properties independent of any operator but *affords* swimming only to those who are equipped and able to do so.

This concept taken from ecological psychology was then extended and popularised by Donald Arthur Norman in his well-known book *The Design of Everyday Things*[25]. For Norman, affordances of an object are

> "the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used." [Norman, 1988, p.9]

Following Norman's definition and examples, an affordance is commonly illustrated by the handles on cups, offering an obvious affordance for holding or by different types of door handles: a simple plate affords pushing, a bar affords grasping and pulling while a knob affords turning. This concept is fundamental for the design of physical objects but Norman also applied it to the design of digital objects, and notes that

> "many existing programs for user applications are too abstract, requiring actions that make sense for the demands of the computer and to the computer professional but that are not cohesive, sensible, necessary, or understandable to the everyday user." [Norman, 1988, p.178]

In our case, making the affordances of corpus exploration tools **perceptible** is crucial to effectively lead the beginner users to the functions available. Throughout the chapter, the words *affordance* and *afford* are thus used in this acceptation; see Blin [2016] for a clarification on the use of the concept of affordance in language-related domains, especially in Computer Assisted Language Learning (CALL).

Minimalism: the example of Google The interface of Google is interesting in this matter. We believe that Google's success as a search engine relies partly on its very minimalist interface, comprising only the company's name (the only

[25]The first edition of the book was entitled *The Psychology of Everyday Things* but was changed in the edition of 2002 for a title that was "more meaningful and better conveyed the contents of the book" (preface of the 2002 edition).

colourful element on a plain white background), a single textual search field and two buttons, `Google Search` and `I'm Feeling Lucky`. Interestingly, the advanced research link used to be visible on the main page but is now a hidden option in the footer. We also note that the second button's original goal is to fully trust the algorithm and redirect the user directly to the website that ranks first.[26] Rose and Levinson [2004] consider this type of research has a "navigational search" because in this case the user implicitly want to "navigate to a specific website". The main and mostly-used button, `Google Search`, helps achieve the two other types of goals, informational and resource goals. Unlike corpus exploration tools, search engines retrieve whole documents, but we can consider the exploration of corpora as a directed (learning something about a particular topic) or undirected (learning anything/everything about a particular topic) informational type of search.

Using keywords in a simple search field seems like an obvious method to search for information, but if the user does not have something specific in mind, another type of search might be useful as well: the search by categories. This method provides the user a means of refining their search starting from a broad category first. For example, Yahoo.com's portal displays categories such as 'News', 'Finance' and 'Sports', which are refined when selected – clicking on the News tab gives access to sub-categories 'US', 'World', 'Politics', 'Tech' etc.

The two methods are implemented on commercial websites: in order to buy new calligraphy supplies, one can search for a specific nib on Amazon by typing the brand along with the model's name, or search for an undefined calligraphy tool and click on "Arts, Crafts & Sewing", "Painting, Drawing and Art Supplies", "Drawing" and finally "Nibs". From this point, the user can select a nib or a related object such as an ink bottle, or choose to refine the query once again by specifying a brand or price range. The second method requires a more complex interface and inevitably takes more time but saves users the frustration of not being able to define a query.

[26] Since Google has implemented Google Instant, this functionality is not accessible unless users manually turn off Google Instant in the Search Settings. Instead, either the user types in keywords and this redirects to the normal search page automatically, or the user hovers over the button and tests one of Google's services. For example, `I'm Feeling Generous` redirects to https://onetoday.google.com, a page presenting featured nonprofit project supported by Google while `I'm Feeling Curious` opens a box in the usual search page results with an interesting trivia.

We will see in Chapter 5 that searching starting from a broad query and then refining step by step can be less overwhelming if efforts are consistent with the three above-mentioned ideas.

Simplification: the BYU interface Among corpus exploration tools, a widely-used one is particularly interesting to comment due to a recent change: the popular BYU website has been upgraded to a cleaner and clearer version in May 2016.

The old query interface for the COCA was "a bit overwhelming" as acknowledged by authors on the new BYU Corpora page.[27]

In order to ensure coherence and allow a comparison of different interfaces, Figures 4.4, 4.5 and 4.6 have all been annotated with the following tags:

- ① the main functions or features of the tool;
- ② the textual field, i.e the main interaction with the user;
- ③ a part-of-speech (POS) support to help the user insert POS by browsing through a list;
- ④ options of the tool that can be skipped.

If we compare Figures 4.4 and 4.5, we notice four major improvements to make the interface clearer and seemingly simpler:

1. the different main functions (see ①) appear as different tabs in the new version, instead of options with radio buttons in the old version. This modification does not change anything in the interaction with the interface (clicking on the KWIC radio button and selecting the KWIC tab *does* switch to the KWIC options the same way) but considering that tabs are commonly used in browsers nowadays because they give the impression of separate windows, we consider this interface as a more user-friendly display in that the action of switching is better *afforded*;

2. the steps mandatory for using the tool have been kept to the minimum: a single textual search field (see ②) and two self-explanatory buttons. Besides, the instructions for use are given in the right panel along with examples and

[27] http://corpus.byu.edu/help/updates2016.asp, consulted on 3rd April 2017

links to more detailed explanations of each important notion for the LIST display. This help panel already existed in the old version of the interface but could not be hidden;

3. as another example of perceptible affordance, the POS list (see ③) is the same as before, but this time the use of a box for the list of POS[28] and its position (next to the textual field instead of being beneath it) shows more clearly that the list *may* be used *several times* to insert a POS in the query, whereas from the old interface, users might get the impression that they *have to* select *only one* POS *after* typing the query;

4. finally, the most striking difference between the two figures remains the **reduction of the size and load of the interface**. This is mainly due to minor functionalities and advanced options giving access to metadata (namely "sections", "sorting and limits", and "options"; see the red ④ on both figures) being completely **hidden**. Despite being optional to the search, they previously took 3/4 of the old interface but now appear as a single line in the bottom part of the interface. Furthermore, the fact that they are displayed in a lighter grey[29] than the shade used in the two buttons suggests that they are indeed optional;

Third generation tool interface AntConc which is also designed to be used by non-specialists also has a fairly light interface. As shown in Figure 4.6, advanced options are hidden behind an `Advanced` button. If the user clicks on it, they are offered the possibility to load a file containing the search terms, or to add "context words" (i.e one or several words that have to appear in the same context as the search term(s) and within a "context horizon" that they can redefine (default settings range from 5 words before to 5 words after the search term(s)).

Unlike the BYU Corpora interface, AntConc does **include options** that are not considered advanced (see ④) **directly in the interface**: check boxes to perform a case sensitive search and/or to allow the use of regular expressions (the `Regex` box), sorting options (`Kwic sort` subsection) as well as an option to reduce

[28] This box actually appears when the user clicks on `[POS]` written in light grey. This zone then transforms into the box shown in Figure 4.5.

[29] When an option is selected, its colour turns into black.

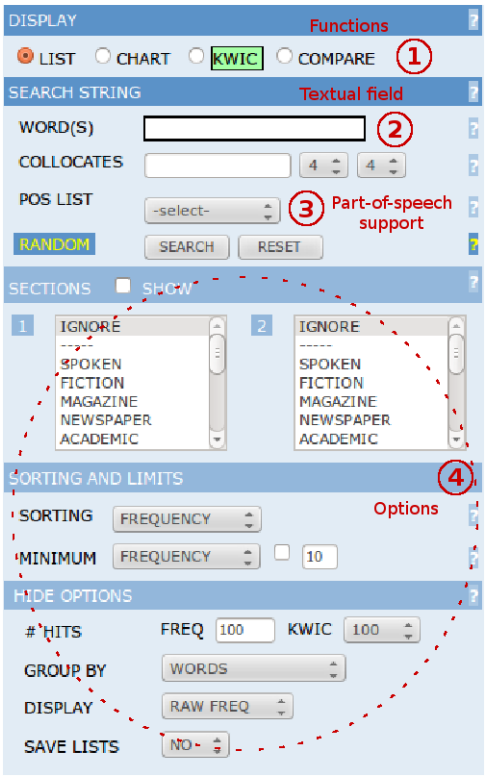


Figure 4.4: Old interface to explore the COCA (before May 2016)

Figure 4.5: New BYU interface to explore the COCA (from May 2016)

or increase the number of characters considered as the context surrounding the search term(s) (`Search Windows Size`). On the other hand, AntConc does not have any support to include parts-of-speech (see the absence of a ③), but this is

intrinsic to all third generation tools. In Section 4.2, we mentioned that second and third generation tools were not specific to particular corpora but are meant to be used with any. On the contrary, fourth generation tools can only be used with pre-indexed corpora and therefore the annotations for each of them must comply to a certain standard. This standard is not necessarily rigid and strictly consistent, especially in the case of the BYU Corpora which hosts corpora in languages other than English such as Spanish and Portuguese. As a matter of fact, the morphology of verbs in those two languages are richer than in English. This adds more complexity in the POS list (specific annotations have been added for the imperfect tense or the conditional mood).

However, we note that the **default settings** of all of these displayed options are adequate for a basic search. The user does not need to modify them. But making options more apparent than in the COCA query interface might not only facilitate the use of the tool by confirmed users, who are more likely to use third generation tools for their flexibility and wide range of functions, but also **arouse the curiosity** of beginner users and **encourage more experimentation** from them. Despite the apparent complexity, this method is advocated by Norman who considers that it is possible to "[make] systems easier to learn and to use [by making] them explorable, [by encouraging] the user to experiment and learn the possibilities through active exploration" [Norman, 1988, p.183]. The only conditions that must be met are that both the activation of a function and its effects be visible to the user, and that the interaction is cost-free or undoable (in the sense that it can be 'undone').

4.4.2 Simplification of the Query Language

Learning a computer language is not so different from learning a natural language in that it relies on the use of a **specific vocabulary** and a more or less strict **syntax**. A complex query language could then be described as comprising a vocabulary that is either difficult to assimilate or obscure and technical, and a strict and complex syntax, difficult to use and to interpret. In other words, simplifying a query language comes down to **relying on as little external knowledge as possible** and keeping the syntax as close as possible to what non-specialists

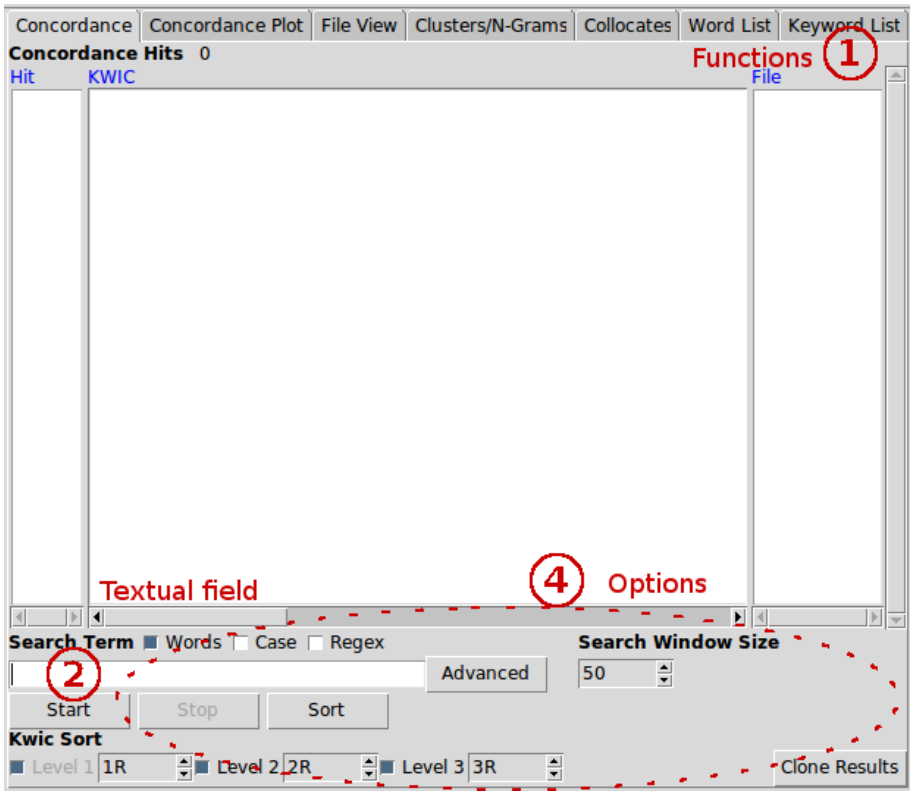


Figure 4.6: AntConc's Concordance interface with default settings

commonly use or at least see.

Different strategies can be implemented. We singled out four of them:

1. **reducing the apparent technicality** by avoiding specialised terminology, or at least by providing a definition or an illustration comprehensible by ordinary users;

2. **replacing rarely used characters** by more commonly used ones to prevent mistyping. Regarding very uncommon characters such as the vertical bar (actually common in both mathematics and computing where it is best known as a 'pipe' and used as the disjunction symbol), this strategy could even prevent a cumbersome waste of time and energy in looking for them on the keyboard or on the internet[30];

[30] As a matter of fact, with the keywords "keyboard vertical line", most of the links that Google returns immediately are about how to type this character (precisely, 8 out of 10 on the first page,

3. **clearing the syntax** by getting as close as possible to natural language, that is to say use special characters as little as possible and consider possible errors, such as forgetting a space, which is not something ordinary users are used to mind;

4. **hiding the query language aspect** by providing support so that the user can select what they like instead of typing it. This support helps in two ways: it provides a list of what is possible to enter into the system to the user and also takes points (2) and (3) further by preventing mistyping the query.

We can observe the three first strategies in Table 4.1. This table was retrieved from the update page of the new BYU Corpora website[31] and features "unnecessarily complex" CQP syntax (second column), the BYU's old and deprecated syntax (third column) as well as its new simplified syntax (fourth column).

From left to right, we can note significant simplification of the query language, following the three first strategies that we introduced:

1. full category names were restored and are used instead of abbreviations (here, "noun" instead of "nn")[32],

2. for lemmas, brackets were replaced by capital letters and

3. querying on a particular lemma restricted to a specific category is possible using the syntax `LEMMA_POS`, which is quite common in corpus linguistics compared to `[[=LEMMA]].[POS*]`.

We can also add that for users who do know what parts-of-speech are and how to use them but not how they are *encoded* in the COCA, there is a support implemented that we already mentioned in Section 4.4.1 (see ③ in Figure 4.5). This support matches the fourth mentioned strategy.

In a similar way, TXM has developed a support called the "query assistant" in the form of a pop-up box. The query assistant gives users the possibility to

the remaining ones are a description of the vertical line from Wikipedia and from theasciicode.com website)

[31] Still http://corpus.byu.edu/help/updates2016.asp, consulted on 3rd April 2017

[32] All abbreviations were not replaced; in fact, it seems like conventional abbreviations such as "adj" for *adjective* or "adv" for *adverbs* were kept.

adjust the number of words they want to work with and select for each of them the properties they are interested in (wordform, lemma, part-of-speech) *through a list* and how they want them to be (starts/ends with or equals to).

Type of search	**CQP syntax**	**Previous BYU syntax**	**New BYU syntax**	**Example**
Word	`[word = ''nooks'']`	`nooks`	`nooks`	nooks and crannies
Lemma (forms of word)	`[lemma = ''decide'']`	`[decide]`	`DECIDE`	DECIDE that it
Part of speech	`[tag = ''NN.'']`	`[nn*]`	`NOUN`	fast NOUN
Synonyms	Not possible	`[=soft]`	`=soft`	soft, smooth, quiet
Customized word lists	Not possible	`[emailAddress@clothes]`	`@clothes`	dress, shoe, sock
Combinations of preceding	`[lemma = ''end'' & pos = ''VV.'']`	`[end].[v*]`	`END_v`	end, ends, ended, ending
	`[lemma = ''eat''] [tag = ''NN.'']`	`[eat] * [nn*]`	`EAT * NOUN`	ate the bananas, eat some cake
	Not possible	`[[emailAddress@clothes]]`	`@CLOTHES`	dress, dresses, shoe, shoes
	Not possible	`[[=clean]].[v*]`	`=CLEAN_v`	cleans, scoured, washing
	Not possible	`[wear] * [=nice] [email@clothes]`	`WEAR * =nice @clothes`	wore some good-looking pants

Table 4.1: Illustration of different type of search in different syntaxes (from complex to simple) and examples of possible output, retrieved from the BYU Corpora page

The new BYU syntax has the benefit of being simpler indeed, yet as rich as the old syntax as it preserves all the possibilities of complex queries previously

allowed. However, it still has to be learned by *all* users including those who were already familiar with the previous syntax, and this necessity is enough to keep certain reluctant users away from corpus exploration tools.

In Sections 4.4.3 and 4.4.4, we present tools that were designed for those reluctant users: complex queries are integrated into the system, and users simply have to click on the desired pattern or provide an example of the desired pattern in natural language.

Hiding the query language Before moving on to those sections, we would like to introduce another corpus exploration system for which the creators chose to make the underlying query language fully invisible to users. Frédérique Mélanie-Becquet and Catherine Fuchs have worked on the elaboration of a database query system entirely based on *formulaires* [Mélanie-Becquet and Fuchs, 2011]. The database allows to search for examples of comparison structures in French and was created using Microsoft Access but was exported to MySQL, which makes it possible to use SQL (Structured Query Language), a widely used query language specifically designed for relational database systems.[33] As these formulaires are not solely intended to be used by linguists, there is **no technical requirement** for users, no need to know any query language: SQL queries are generated automatically according to what is selected by users. Anyone interested in comparison structures in French only interacts with the formulaires, which are in fact web pages with a certain number of lists. This idea is very much similar to that of the first interface of the BYU Corpora page, and thus also has the drawback of being possibly overwhelming for the beginner user. Indeed, novices might be reluctant to go through the high number of options and objects in the lists despite efforts towards making the interface as clean and unified (in colours, fonts and page structure) as possible. However, there is concrete evidence of adaptation to non-specialists in this project that we need to mention: first, the **terminology** used is meant to be **comprehensible** by a large number of people[34] and indeed

[33]Simply put, a relational database is a collection of tables with each table consisting of a set of rows (unique objects) and columns (attributes).

[34]"Le métalangage utilisé est simple. Il ne reprend pas une terminologie complexe qui ne serait intelligible que pour un nombre limité de personnes." [Mélanie-Becquet and Fuchs, 2011, p.281]

notably there are no technical acronyms for instance; second, users who do not want to go through any option at all can simply skip this step and visualise the whole database.

Those two possibilities are allowed by the fact that the database is very specialised and that it only focuses on comparison subordinate clauses in French, which is much more targeted than a multi-purpose corpus like the COCA has to offer. We can therefore assume that even beginner users already know what they are looking for when using this system.

4.4.3 Example-based Queries

In this section, we have described corpus exploration systems that have implemented a means to help users in elaborating queries, especially those with no background in a field related to corpus exploration. Example-based query systems are particularly efficient in so far that they do not require from users to know how to use a query language at all. As their name suggest, those tools **only need *examples* in *natural language*** – as opposed to *constructed* languages like query languages – to perform a query. The example input by the user is then **automatically processed** by the tool and **transformed into a query** so that it may be interpretable by the program and match segments of the corpus.

Let us take the example of the GrETEL project[35], a CLARIN project from the University of Leuven which achieves the feat of **simplifying the query of treebanks**, notably known to be more complex than queries relying on parts-of-speech due to the complexity of the representation in trees. We performed a simple test using Poly-GrETEL [Augustinus et al., 2016], an online corpus exploration tool which allows syntactic queries on a Dutch-English parallel corpus and explicitly aimed at specialists such as translators or linguists carrying out comparative studies between Dutch and English, as well as non-specialists such as language learners. Poly-GrETEL is an extension of GrETEL [Augustinus et al., 2012] but as the latter only works for Dutch at the moment, we chose to illustrate the mechanism

[35]GrETEL stands for Greedy Extraction of Trees for Empirical Linguistics and is defined as a "user-friendly search engine for the exploitation of treebanks". Tools and documents on the project are available at http://gretel.ccl.kuleuven.be

using screenshots of Poly-GrETEL's monolingual search, for the sake of having examples in English.

Poly-GrETEL can be used following this step-by-step description:

- **Step 1 – Give an example**: as expected, we can see in Figure 4.7 that the first step requires the user to input "a sentence containing the (syntactic) characteristics [we] are looking for". In our case, we chose to use the basic monolingual search in English with the simple sentence given by default: "This is a sentence.". When this step is done, the tool proceeds with the syntactic analysis of the input performed by the integrated syntactic parser.
- **Step 2 – Input Parse**: the output of the parsing is then shown in the second step (Figure 4.8), the validation of the input example now represented as a top-down syntactic tree. The tree must be read from the top, starting from its root. The second line is the segmentation into phrases with the dependency relation of the phrase in red (e.g `nsubj` for nominal subject) and the category in black (e.g `NP` for noun phrase). Finally, the last line is the word node with four lines of information: (1) the dependency relation again, (2) the part-of-speech, (3) the lemma and (4) the word (or token) in grey. This step is important principally to specialists who can check if the parsing contains errors while non-specialists are unlikely to understand how to interpret the tree.
- **Step 3 – Select relevant parts**: once again, in this step, specialists might use their knowledge and refine their query but non-specialists can also give it a try and use their intuition to select how relevant they think each token is, and if they should appear as wordforms, lemmas, or simply the "word classes", i.e parts-of-speech. We can note that for this step, help is provided for non-specialist users with default settings and guidelines on the terminology used along with examples for each term (Figure 4.9).
- **Step 4 – Select a treebank**: Poly-GReTEL allows users to perform queries on the Europarl corpus from either 2000 or 2001, or on both together.
- **Step 5 – Query Overview**: like Step 2, this step shows a parse tree. This

time, the tree is simplified as it only keeps information that the user has selected in Step 3. In our case, because we left the default settings, all tokens were to be replaced by their corresponding word class only in the query so only the dependency relation and the part-of-speech appear in the query overview tree.

- **Step 6 – Results**: this last step obviously shows the matching sentences, in the form of a 3-column table: (1) Sentence ID (metadata to identify the sentence and the corpus from which it was taken), (2) the matching sentences in English (first line) where the matched segment in bold font as well as in Dutch (second line, purple colour) and (3) the number of hits. This table is preceded by a summary of the query with the total number of hits, of matching sentences and of sentences in the corpus in total so that the user can have an idea of the distribution of the syntactic structure in the corpus. Interestingly, the XPath query is also shown. This is useful for specialists who would like to check the query structure or modify it directly.[36] Our simple sentence happened to be transformed into the following query: `//node[@cat=''S'' and node[@rel=''nsubj'' and @cat=''NP'' and node[@rel=''hd'' and @pos=''DT'']] and node[@rel=''hd'' and @cat=''VP'' and node[@rel=''cop'' and @pos=''VBZ''] and node[@rel =''hd'' and @cat=''NP'' and node[@rel=''det'' and @pos=''DT''] and node [@rel=''hd'' and @pos=''NN'']]]]`

Back to Step 2, we can consider the fact that the input given by the user has to be automatically parsed is a serious drawback to example-based system. The quality of the syntactic analysis then relies entirely on the quality of the parser. Furthermore, there is no means of correcting the analysis for specialists (in case of dissatisfaction, Poly-GrETEL suggests to input another sentence), and non-specialists could be misled by an error that remains unnoticed, due to insufficient knowledge.

[36] GrETEL allows specialists to type an XPath query directly instead of going through the example-based system.

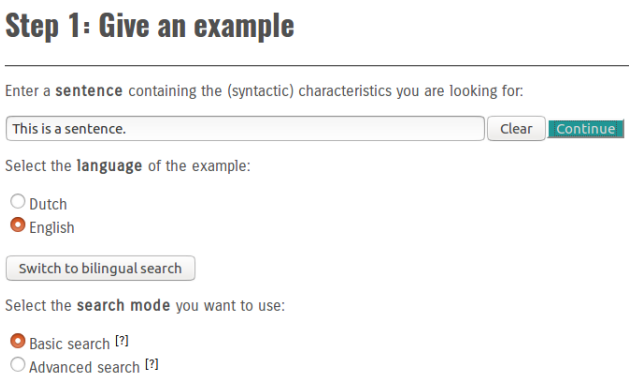


Figure 4.7: GrETEL's refining system for non-specialists: Step 1

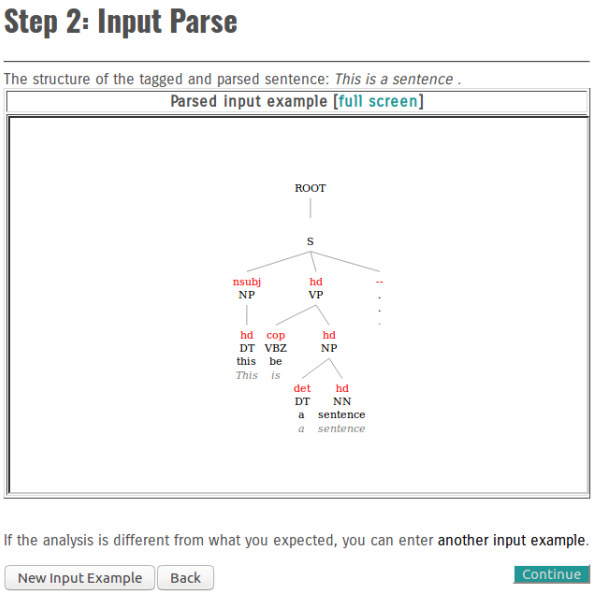


Figure 4.8: GrETEL's refining system for non-specialists: Step 2

Errors in Corpus Linguistics Dealing with errors is frequent in corpus linguistics.Indeed, nothing, not even reference corpora, is free of errors. We can find the following disclaimer in the editorial indications of the British National Corpus website[37]:

> "Despite the best efforts of its creators, any corpus as large as the BNC will inevitably contain many errors, both in transcription and

[37] http://www.natcorp.ox.ac.uk/docs/URG.xml?splitLevel=-1, consulted on 15th May 2017.

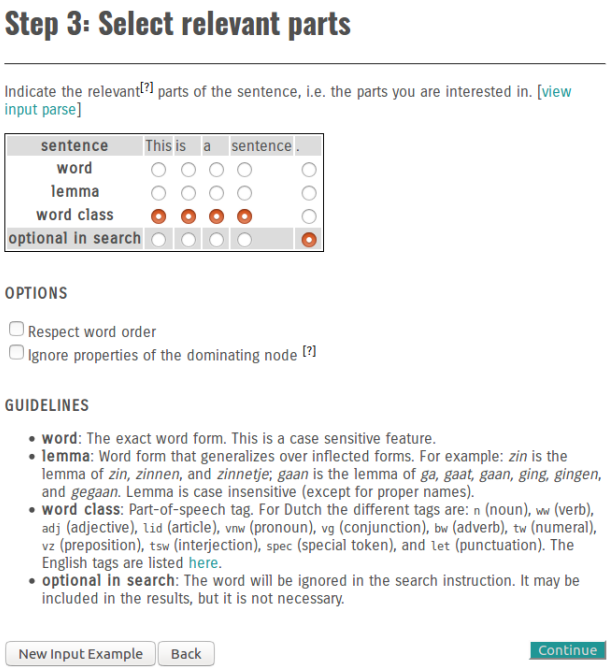


Figure 4.9: GrETEL's refining system for non-specialists: Step 3

encoding. Every attempt has been made to reduce the incidence of such errors to an acceptable level, using a number of automatic and semi-automatic validation and correction procedures, but exhaustive proof-reading of a corpus of this size remains economically [un]feasible[38]."

The first factor of errors is **human**: natural language is full of **ambiguity**, sometimes tricky to analyse even for specialists. Any **disagreement** on the processing of ambiguous cases or any unclear point in the annotation protocol inevitably leads to annotation inconsistencies, hence the use of inter-rater agreement measures such as Cohen's kappa[39]. Incidentally, even with a crystal clear annotation protocol, humans are neither infallible nor inexhaustible but rather prone to making mistakes. The advantage of errors made by a computer program is that, unlike human errors, they are systematic, which means that they are more **usable in terms of computation**. However, it is more difficult for a machine to remove a natural language ambiguity and because computer programs also follow a set of

[38] We believe that there was a typo in the currently available text...
[39] A strong agreement is said to be reached is kappa is above 0.6.

instructions given by their creator(s), they highly depend on human decisions as well. Eventually, in some cases, ambiguities cannot be removed even by specialists, due for example to a lack of contextual information. In fact, ambiguities may not even be meant to be removed in case of wordplays.

Errors in example-based systems Errors of annotation in a corpus could definitely lead to a wrong analysis of a phenomenon. However, in the case of matching an automatically parsed sentence with sentences from corpora which were also **parsed with the same tool**, errors may not be such a weighty problem. Meurers and Müller [2009, p.4] note that:

> "this setup has the interesting property that errors made by the parser do not have to be a problem given that both the initial instance of the search pattern and the corpus are processed with the same tool; the purpose of the parser is not to provide the ultimate linguistic analysis but to provide a link from the instance used to create the search pattern to other instances of that pattern in the corpus."

This comment was made to describe the Linguist's Search Engine[40], another tool that "was designed to provide the broadest possible range of users with an intuitive, linguistically sophisticated but user-friendly way to search the Web for naturally occurring data" [Resnik and Elkiss, 2005]. Philip Resnik and Aaron Elkiss also started from the observation that even specialists as linguists may not be willing to learn a query language, thus proposing an example-based tool to simplify the access of authentic data – in this case, the Web.

Likewise, we believe that parsing errors may be an advantage for our experiments. The benefit of reannotating the corpus is described in the perspectives of our work in Chapter 7.

4.4.4 Predefined Queries

Example-based query systems considerably reduce the amount of knowledge required from users to explore a corpus syntactically. Corpus exploration tools based

[40] http://wse1.webcorp.org.uk/

on predefined queries take the issue a step further by requiring even less skill. The only one to our knowledge is the KKMA concordancer, only available for Korean and with an interface in Korean.

The KKMA Project (꼬꼬마 프로젝트) conducted at Seoul National University has notably released a widely-used Morpheme Analyser[41] for Korean, as well as an interesting concordancer tool available online.[42] This concordancer not only allows to build concordances of tokens or n-grams but also to retrieve syntactic constructions if they match the predefined patterns.

Let us take a look at Figure 4.10. KKMA's concordancer corpus exploration process is similar to GrETEL's but all steps are gathered into one single window (although there are different tabs).

- **Step 1** – the user types a sentence (a phrase also works but a full sentence is more likely to be analysed correctly by the parser) in the textual field (see ①);
- **Step 2** – then, he or she clicks on the `Syntactic analysis` (분석) button (see ②) to validate the sentence and start the analysis. Unlike GrETEL, KKMA does not require the user to validate the parse tree nor even see it. The tree only appears when we click on the fourth tab "See the syntactic tree" (구문 분석 보기) and is very similar to that of Poly-GrETEL (shown in Figure 4.8) except that only the dependency relation is given for each link. We can also note the absence of a step to choose the corpus because KKMA only works with the Sejong Corpus, the reference corpus for Korean (thoroughly described in Appendix 3.5);
- **Next steps** – the user is confronted to a set of features displayed in different tabs. Those related to the input sentence are the first two tabs: "Examples following a pattern" (양식에 따른 용례) and "Examples of vocabulary use" (단어 쓰임 용례).

[41] Actually KKMA stands for "Kind Korean Morpheme Analyzer". This analyser is used in our preprocessing chain, see Chapter 6.

[42] `http://kkma.snu.ac.kr/search` for the search by morphemes and `http://kkma.snu.ac.kr/concordancer` for the search system based on grammar.

Examples following a pattern KKMA's concordancer has 99 predefined queries corresponding to 99 predefined syntactic constructions (see Appendix X). If one or several constructions found in the input sentence match those from the list, they are displayed in this first tab. In our case, we chose to use the default sentence again, "그가 규칙을 어겼기 때문에 규칙에 따라서 그를 처벌함으로써 본보기를 보이는 것이다." and the analysis revealed five different syntactic constructions (in the left 3-column table in ③). The first column is the ID of the construction within the list, the second column shows the syntactic construction's representation and the third and last column shows the number of occurrences of this construction in the corpus.

When a construction is selected, the concordance lines appear on the right (see ④). We selected the third construction, *-a/ese* -아/어서[43], a conjunctive suffix bound to verbal stems and which typically denotes causality. In the concordance lines, the tokens containing this morpheme are in red, metadata about the corpus sample are given in blue while the context is written in black colour. The last part of each concordance line (in brackets) can be clicked on to show more context. Finally, clicking on the matching sentence shows its morphosyntactic analysis (see the grey part in ⑤, which appeared when we clicked on the first matching sentence) and clicking on the metadata parenthesis opens a pop-up window with more information on the sample (notably the type of sample, the language register (standard or not), the number of tokens or morphemes, year of collection and even the header of the original XML file).

Examples of vocabulary use The second tab concerns the vocabulary used in the input sentence but works in a very similar way to the syntactic construction tab. Figure 4.11 shows the step-by-step process to show a concordance of a given word, here "따르다" (*to follow*). This time, what is displayed in the table on the left is the vocabulary (i.e the nouns, verbs, pronouns, adverbs, etc.) found in the

[43]This morpheme is the only one with no "+" between the verb and the affix because unlike the other affixes shown in the table, it cannot be simply attached to the verb stem. First, vocal harmony applies: allomorph *-ase* -아서 is used with "bright" vowels (/a/ and /o/) and *-ese* -어서 is used in all other contexts. Then if the verb stem ends with the same vowel as the first syllable of the affix, the latter is dropped. One exception is the verb *hata* 하다 ("to do") which combines unexpectedly with *-ese* -어서 and appears as the contracted form *hayse* 해서 (*ha* 하 + *-ese* -어서).

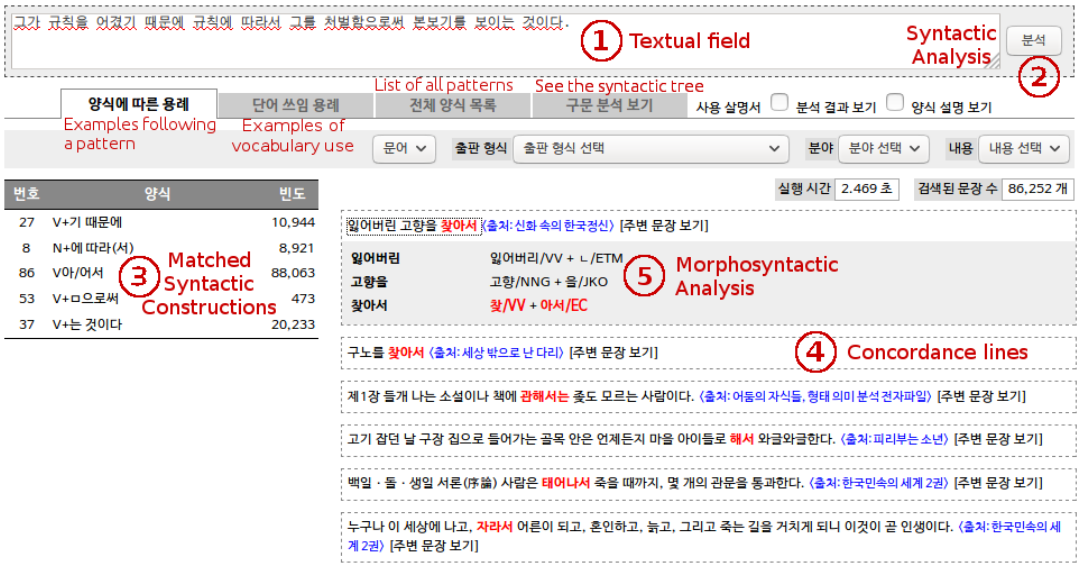


Figure 4.10: KKMA's concordancer interface: an example of search using a predefined syntactic query

input sentence. The first column only contains checkboxes to be checked to select one of several words[44], the second column shows the lemma, the third shows the part-of-speech and the fourth is still the number of occurrences in the corpus. We decided to show the process step-by-step in the figure because it is less intuitive than on the previous tab: clicking on the lemma itself this time opens a pop-up window of the bilingual Naver dictionary[45] with a direct link to the selected word.

Incidentally, unlike the syntactic constructions, the words in the table are not predefined and thus do not necessarily have any occurrence in the Sejong Corpus. In this case, the guessed part-of-speech is followed by "추정범주" (*estimated category*) and the last column simply contains `0`.

The predefined syntactic queries are not completely hidden from the user. They actually show in the third tab, the "List of all patterns" (전체 양식 목록). In the case of *-a/ese* -아/어서, the query is `[V][아서/EC,어서/EC]`, which put into words would be: "any verb followed by *-ase* -아서 or by *-ese* -어서 both as conjunctive

[44] If several words are selected, all of them must appear at least once in the same sentence but in any order and not necessarily in a contiguous way. This search resembles AntConc's "context words" briefly described in Section 4.4.1.

[45] http://endic.naver.com

Figure 4.11: KKMA's concordancer: an example of search using an automatically segmented word

particle". In this part of the tool, specialists are given the possibility to adapt the query. For instance, we could search only verbs which are followed by *-ase* -아서 and not its allomorph by typing `[V][아서/EC]`. In other words, such query would retrieve verbs containing "bright" vowels.

4.5 Conclusion

This chapter described a wide range of functionalities through the overview of corpus exploration tools. We saw that functionalities are implemented depending on various factors: the general purpose of the tool (whether it should be used to characterise a text or a corpus, or whether it should be used to observe some linguistic phenomenon), the methodology of research it supports (fostering a data-driven

approach or rather being useful for a data-based approach), the type of resources exploited (raw corpora that users can upload, annotated corpora following a certain standard or any, specific reference corpora, parallel corpora etc.) and of course the width of the range of users targeted (from complete novices to experts).

The attractiveness of corpora as a resource allowing to look at naturally occurring data urged programmers of corpus exploration tools to simplify the use of their tools either by rethinking the interface to be more user-friendly, or by simplifying or even hiding the query language or by providing help and support in many ways, including building a vast ready-to-help community and offering training sessions for all levels of users. This last point is important especially when the tools remains complex and challenging. In this case, both technical and human assistance are crucial [Schaeffer-Lacroix, 2015].

Limits of available tools A natural but implicit limitation of all corpus exploration tools is that they conform with linguistic theories and rely exclusively on **decisions made by specialists**, from the segmentation to the annotation of corpora via the set of functions implemented in tools. Meurers and Müller [2009, p.3] warn that"one needs to keep in mind that the annotation schemes used are the result of linguistic theorizing and insight."

Considering that the very first step of corpus processing – the segmentation – already involves linguistic decisions that have implications for every following step, it is **unavoidable** to rely to some extent on linguistic theories. What is important is the **degree of reliability**: the case of KKMA's concordancer and its predefined syntactic queries is particularly interesting in that it fully and explicitly relies on syntactic annotations, leaving very little room for flexibility and no room for serendipidity. Predefining queries is an excellent solution to open the access of syntactic queries to language learners with no background in corpus exploration as they require almost no effort from users – especially because some of the syntactical constructions appear as such in language textbooks. However, the possibilities of queries are limited to the 99 constructions predefined by linguists, unless users are skilled enough to understand how to build new queries from the observation of the predefined queries, as no support is included. In fact, to have access to refined

queries, users must be able not only to infer the method to build new queries but, most importantly, to define exactly what they are looking for. In other words, the user must be a specialist.

To sum up, we raised two important issues.
Firstly, if users are non-specialists and do not know **how to describe precisely what they are looking for** in terms of a query or even in linguistic terms, they cannot find anything. It is far from easy to assign a part-of-speech to a word properly, or to be able to put boundaries to a syntactic construction in a foreign language that we are still learning. Example-based query systems are a good means to improve this situation because they rely on an semi-automatic processing chain where users are asked to put as much knowledge as they have, but they also may input nothing else than a sentence they have heard or seen somewhere. In case of language learning, reproducing a sentence that contains a construction they are interested in from a textbook may be sufficient. The rest depends on the quality of the processing chain and its default settings.

Secondly, if users are specialists, they may go as far as their skills allow them to go but always within the **boundaries unconsciously set** by their own expertise or by the decisions made by the team that has built the corpus they are working on as well as the corpus exploration tool.

In both cases, the approach relies more on **hypotheses made by linguists rather than on the data itself**. Among the tools that we mentioned in this chapter, only a few have a data-driven function, among which ConcGram©: concgrams can be computed in a fully automatic way starting from scratch, without any input from the user. However, concgrams focus on word associations, not on syntactic constructions.

In Chapter 5, we describe a syntactic similarity-based query system which could be complementary to all the functionalities mentioned in this chapter. Our system is meant to be used by non-specialists and it has to comply with the simplification rules that we raised. Therefore, we also chose an example-based system but instead of matching *exactly* the query produced by the processing chain according to information input by the user, it matches syntactically *similar*

segments. This flexibility gives the opportunity to observe close but not identical structures and aims at questioning the established analyses and grammars. Our innovative approach conforms with the following situation, described by Meurers and Müller [2009, p.4]:

> ‘ [...] current syntactic research frequently questions the established analyses, and a particular set of data might be interesting precisely because the delineation of a phenomenon and/or its analysis are not yet adequately understood.”

Chapter 5

Example-based and Similarity-based Syntactic Query System

5.1 Introduction

When language learners come across some unknown grammatical construction, they may naturally tend to look it up in textbooks or directly in grammars, which provides a definition as well as several examples of canonical uses. However, in some cases, explicit rules and a small number of uses are not sufficient to comprehend fully a grammatical construction, especially if the learners' native language(s) is (are) typologically distant from the target language.

Another solution could be to search more examples, perhaps in authentic corpora, to observe and analyse what is considered as natural and usual in the target language. In this case, learners would therefore be actors of the construction of their own knowledge, which was encouraged by John's Data-driven learning approach (see Section 2.3.3). However, using a grammatical construction as a query may not be as easy as using plain words to obtain concordances of single words or sequences of words. Indeed, learners would need to provide a **description of the construction** to be used as a query. This seemingly simple step actually requires not only **linguistic knowledge**, in so far as certain constructions are searchable only with (morpho)syntactic annotations, but also knowledge in the **query language** of the corpus exploration tool. None of these are self-evident for novice

users, including language learners. Furthermore, non-specialists might want to focus more on the output, rather than spend time and efforts in order to master a query language.

In this chapter, we present our attempt to provide the missing link between examples taken from textbooks to illustrate grammatical constructions, and subsidiary instances of those constructions that can be found in context in native corpora. After a brief description of the issue of syntactic querying and of our objectives, we provide two complementary descriptions of the whole processing chain: in Section 5.3, each step of the chain is disclosed in its full potential, whereas Section 5.4 illustrates the simplified processing chain, designed specifically for novice users. Lastly, Section 5.5 gives a brief overview of the use of similarity and dissimilarity measures in Information Retrieval (IR), and points at their original use within our query system.

5.2 Presentation

From a sentence in input containing the target syntactic construction, our tool provides other sentences with this construction and its context, ranked by similarity with the initial input. We could therefore retrieve hundreds of relevant examples of a given construction based on a few examples displayed in a textbook, including similar constructions which are not mentioned in grammars as possible variations.

Syntactic constructions The tool that we propose simply works like a search engine, the main difference being that it retrieves sentences based on the *syntax* of the query instead of its words, as we search for syntactic constructions. We use the term "construction" not in reference to Construction Grammar but in the sense of "structure" and "pattern":

> "Syntactic structures are analysable into sequences of syntactic categories or syntactic classes, these being established on the basis of the

> syntactic relationships linguistic items have with other items in a construction."

Following this definition from David Crystal's dictionary of linguistic terms, we believe that a syntactic construction is characterised by its syntactic configuration, i.e., the position of categories of words in relation to each other, rather than on the words that instantiate them. For this reason, we decided to build a system which relies as little as possible on lexical items, and which does not use 'bag-of-words' models. For this purpose, we adapted our measures so that the word order is taken into account (see the adaptation of the Jaccard/Dice distance to bigrams in Section 6.3.3). Indeed,'bag-of-words' models represent textual data as vectors of words, in which the position of words in the document is not taken into account. The image conveyed by this expression shows that in these models, words could have been cut out of a text and mixed in a bag, just like we do when we vote by secret ballot: the order in which the ballots were put in the bag is completely lost and ballots are taken from the bag random. However, the vote counting is strict and the number of votes is considered crucial, as crucial as the frequency of words in 'bag-of-words' models.

Illustration Let us consider the following two phrases. Each phrase is annotated in (morpho)syntax, as each of the tags represents the word class, or *part-of-speech* (hereafter POS), of the word above. Incidentally, the tags used in those examples do not belong to any specific tagset: `DET` stands for determiner, `NOUN` for noun, `PROREL` for relative pronoun, `PRO` for (personal) pronoun and `VERB` for verb. They were made up to keep the tags as transparent as possible for a better apprehension of the illustration.

(12) the person whom I see
DET NOUN PROREL PRO VERB

(13) that dream that you had
DET NOUN PROREL PRO VERB

Examples 12 and 13 both contain a relative clause. Although they share no common lexical items, they do have the same syntactic configuration, i.e., the same sequence of morphosyntactic tags in this case.

Requirements for a syntactic query A person who is working on relative clauses cannot retrieve the two phrases *the person whom I see* and *that dream that you had* unless they do a specific query like "`DET NOUN which|that|whom PRO VERB`". This query is built for the sake of the example and does not comply with any existing tool. Nonetheless, it illustrates the degree and variety of knowledge typically required to define a syntactic query:

- Anyone could list relative pronouns that may appear in the context of the example, as we did with *which*, *that* and *whom*, but provided that they already have some knowledge on relative clauses and the use of the different relative pronouns.
- Unlike other items in this query, the relatives pronouns are not separated by a space but by a vertical bar called *pipe*. This character is well-known by users of regular expressions, among others, as the disjunction symbol. However, to novice users, it may only be a rare character whose usage and even position on the keyboard are unclear (see the anecdote in Section 4.4.2), and this character is only a detail in the syntax of the language query.
- Except for the relative pronouns, the remaining items were all replaced by their morphosyntactic tag in the query. This task is necessary to search for similar constructions that do not contain exactly the same words. Yet, it is not an easy task for two reasons:
  - Replacing a word by its POS first implies to know what POS are. In France, we are trained to identify word "categories" or "classes" (that we actually call "nature") from elementary school[1] but the struggles students face in distinguishing a preposition from a subordinating conjunction in university, even students in Linguistics[2], show that its learning is far from achieved and its teaching still a challenge.

[1] Among the competences to be mastered by the end of the "cycle de consolidation" (*cycle of reinforcement*, around the age of 11), we note the identification and categorisation of nouns, verbs, determiners, adjectives and pronouns (as well as prepositions implicitly, as prepositional phrases are mentioned) and by the end of the "cycle des approfondissements" (*cycle of enhancements*, around the age of 14), prepositions (explicitly this time), adverbs, conjunctions and interjections are added. Programmes were retrieved on the website of the Ministry of National Education, Higher Education and Research `http://www.education.gouv.fr`.

[2] Obviously students from other disciplines are not spared, our students are not particularly

– As we mentioned, the POS from 12 and 13 were invented and chosen for their transparency. However, 'real' POS tags are hardly as transparent. In a real situation, a novice user has to master a specific tagset (the one used on the annotated corpus to be investigated) enough to be able to use it adequately, i.e., to replace a word by its POS correctly. A large tagset means a high degree of precision in morphosyntactic tags and thus in the query, as well as a higher difficulty in its apprehension. Moreover, errors are not cost-free as a wrong POS may lead to a wrong search; and an error is not easily spotted and corrected as there is no specific feedback on it. As a matter of fact, mastering a tagset also takes time and effort to specialists. Even though there are some conventions and a tendency to use identical (see the works on Universal Dependencies) or at least similar tagsets, the problem remains unsolved.

5.2.1 Objectives

The objective of our system is twofold:

- the system must be **accessible to non-specialists**;
- the system must allow a **flexible search** for a syntactic construction.

The first objective can be achieved with the help of an example-based query system, similar to the one that we described in Section 4.4.3. This system allows complex queries to be defined by a natural language input and by the further participation of the user who is asked to use their intuition to refine the query. No prior knowledge of the query language nor of the tagset is required. Furthermore, the experiments described in this dissertation is nothing more than a proof of concept of the system. The remarks made about the simplification of the interface in Chapter 4 should be taken into account for the future development of the tool.

It is important to note that this simplified query system is the default version for the general (non-specialist) public. In order to come up to the needs of intermediate and expert users as well, advanced options are hidden at different steps.

to blame, and as instructors, we shall bear our share of responsibility... and so shall the government(s) for its implication in overcrowded classes.

These options allow a greater control on the query (e.g. the choice of the way each token is considered in the query, described in Section 5.3.3) as well as more search possibilities (e.g. the choice of a *search mode*, described in Section 5.3.4).

The second objective is strongly linked to the first. A concordancer is a very powerful corpus exploration tool when used adequately, i.e., with a precise query. However, defining a specific need or question – especially on potentially complex syntactic constructions – and transforming it into a query is not a simple task for non-specialists. As query systems such as the ones currently implemented in concordancers do not allow vagueness, we decided to build a complementary function based on (syntactic) similarity measure instead of strict matching. We believe that the flexibility allowed by similarity measures matches the vagueness of the query defined by the user. Vagueness in the input inevitably calls for vagueness in the output. Yet this is not necessarily a problem, as we can make good use of it and retrieve unexpected output, or we can help the user to define the query progressively by using relevance feedback (see 5.3.6). Like a search engine which presents alternatives to users who entered imprecise or inaccurate inputs[3], we propose a query system with different alternatives anticipating on the user's need (see the modes in Section 5.3.4).

5.2.2 System Architecture

To address the issue of allowing syntax-based exploration of a corpus while ensuring that our program is accessible by non-specialists, we designed a processing chain that takes as input simple natural language queries (phrases or sentences) and automatically provides morphosyntactic tagging and classification of the matching sentences.

The processing chain is divided into seven main steps:

1. user input (in natural language);

2. automatic syntactic analysis;

[3]For example, Google suggests "did you mean x?" for mispelled or inaccurate keywords and sometimes even include directly documents retrieved using synonyms of the input keywords if their algorithm considered the alternative as adequate.

3. query formulation;
4. similarity computation;
5. ranking and clustering;
6. query refinement or validation of propositions;
7. final output.

The complete architecture of our proposal is detailed in Figure 5.1. The flowchart can be interpreted using the following hints:

- the beginning of the process (a manual input) and its ending (the output) are both represented with thick contours;
- shapes represent different types of objects: rectangles are actions or operations, ovals are external resources and diamond shapes are used in their most common function to represent decisions (hence, the *yes* and *no* arrows);
- dashed items are optional;
- as for colours, they are used for different subtypes of objects:
  - the dominating green colour is used for operations executed by the tool;
  - yellow is used for external tools;
  - orange is used for corpora, the other type of resource;
  - eventually, all steps requiring an intervention by the user are represented by shapes with blue background and contour. The number of interventions were reduced to the bare minimum in compliance with our objective of calling upon the user as little as possible.

5.3 Step-by-Step Processing

The seven steps that we mentioned in the description of our system architecture also appear in the algorithm flowchart. Each step is detailled in this section.

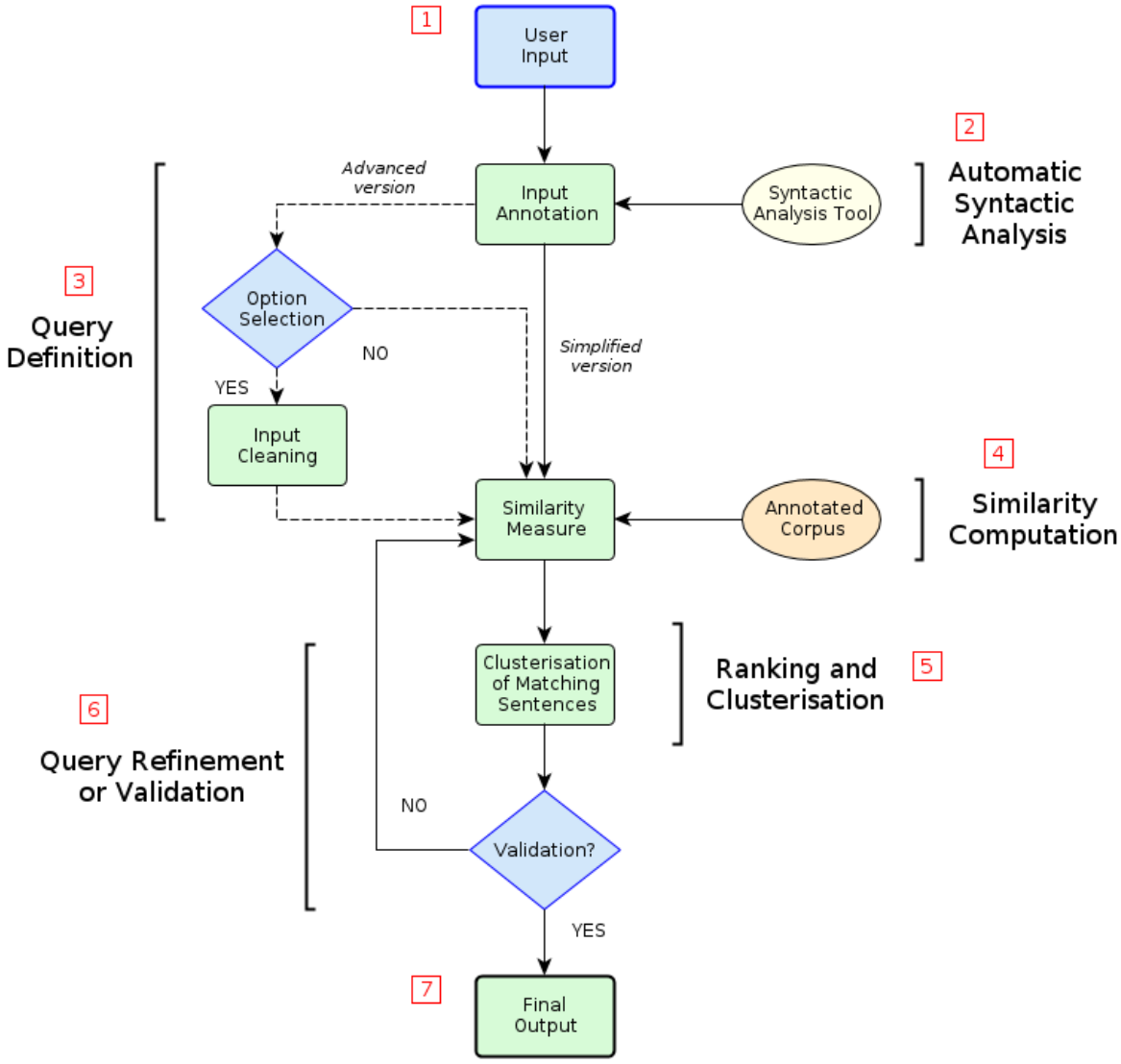


Figure 5.1: Algorithm flowchart of the syntactic query system

5.3.1 User Input

A query system is used to answer a specific question. The very first step of the system is therefore naturally the initialisation of the system with an input given by the user. Because our system is **example-based**, the input is an *example in natural language*. This implies two properties:

1. the input is supposed to illustrate the syntactic structure the user wants to

investigate;

2. the input must be typed in **natural language**, and not in a query language.

A typical input would be a sentence either copied from a textbook or extracted from a naturally-occurring written or spoken context. The first type of input has the advantage to be constructed by professionals of education specifically to help learners understand a grammatical item, but the second type is increasingly likely to happen as the learner becomes autonomous.

The input that we have described here illustrates the typical usage for which the tool was built, with default settings chosen to keep its use as easy as possible. Beyond those default settings, more options can be available:

1. the possibility to enter several inputs instead of a single one in order to define the query more precisely from the beginning of the search;

2. the possibility to type in a contiguous sequence of words[4] instead of a whole sentence although a sentence may be more accurately analysed by the syntactic analysis tool in the next step;

3. the possibility to enter an annotated sentence and skip the automatic syntactic analysis.

While the first two options are likely to be useful to non-specialists, the last option would only be useful to expert users.

5.3.2 Automatic Syntactic Analysis

Because our system is supposed to be used by non-specialist users, the morphosyntactic tags are provided automatically by an external resource: a morphosyntactic tagger. The purpose of this step is to transform the input written in natural language by the user into a query, i.e in our case, into something similar to what is found in the corpus.

[4] In other words, *n-grams* but not *skipgrams*.

As the corpus that we are using is the POS-tagged version of the Sejong Corpus, we chose to use a morphosyntactic tagger that uses a similar tagset: KKMA.[5] If we choose to explore the parsed version of the Sejong Corpus instead, we would have to use a syntactic parser trained[6] on the Sejong to make sure the query and the sentences from the corpus can match. In the same way, a dependency tree-tagged corpus requires a dependency parser. In other words, the processing chain is valid as long as the output of the syntactic analysis tool has the same annotations as the ones found in the corpus.

Annotation errors? Using an automatic tool in a processing chain raises the issue of annotation errors. If we use a reference corpus, there will inevitably be a discrepancy between the annotation of the query and the annotations in the corpus as the latter has been corrected. However, even a (large, by definition) reference corpus is *never* error-free. Better still, as our intent is to match an automatically annotated query with a corpus, it could be beneficial to reannotate the corpus so that annotations are consistent between the query and the corpus. Another solution is to use both the corrected version of the corpus and the reannotated one altogether.

These two points are discussed respectively in paragraphs "errors in corpus linguistics" and "errors in example-based systems" in Section 4.4.3, and we consider these remarks in the perspectives of our work in Chapter 7.

5.3.3 Query Formulation

Once the input used as query[7] has the same annotations as in the corpus, what needs to be defined next is what exactly in the input pertains to the syntactic construction the user wants to investigate.

[5] `http://kkma.snu.ac.kr/`, from Seoul National University. The details on this choice are found in Section 6.2.3.

[6] In machine learning, a program is said to be *trained* on a corpus if this corpus was used to build the tool. In this case, calculations are based on the properties of this particular corpus.

[7] From this step on, the input is not considered as an input anymore but as a query.

Type of input We have mentioned in the first step that using a sentence as input may be better because it would be more accurately analysed by the syntactic analysis tool. Most of NLP tools are indeed trained on whole sentences, with the exception of the ones specialised in short messages such as texts or tweets or the ones specialised in speech, where the notion of sentence is fuzzy. Their performance is therefore usually better on sentences than on isolated phrases or chunks. Yet, we should note that a sentence, let alone a long sentence, can also decrease the relevancy of the similarity computation because more words also introduces more noise if they are not relevant for the target construction. The program does not know what is relevant to the user: in order for the program to help the user define the query, the user has to help the program target the construction.

Search definition The query formulation corresponds to the third step of Poly-GrETEL's example-based system shown in Figure 4.9. The user is faced with the input they have entered, along with its (morpho)syntactic tagging, and has to indicate which part(s) is (are) relevant for their search. Several options are offered and for each word, the user has to choose if the word should appear in the query as such, or replaced by its available tag(s) (lemma, part-of-speech or both), or even... if the word should appear at all. The words not selected at all are the ones the user considers not relevant for the search. In other words, this single step allows to define implicitly two levels of information[8]:

1. first, a frame of relevant words for the search;
2. and second, within this frame, the words that are part of the target syntactic construction, and in which form they are relevant.

If we take a look at Example 14, we understand that the user here is interested in relative clauses with "who" as the relative pronoun. Indeed, an entire sentence was input but only the words with at least one black bullet underneath are used in the query. In the first row of bullets, only "who" was selected, which means

[8]These two levels are only available in the advanced version. The simplified version uses an automatic way of determining if a word should appear as such or be replaced (see Section 6.3.2). By default, all of the non-selected words are replaced by their POS tag if they are lexical words while all of the selected words are used both in their original form and their POS tag.

that it is the only token from the input that has to appear in output sentences as well. Tokens selected in the second row will all be replaced by their POS. In other words, only the embedded nominal phrase without the modifiers *lonely*, *over there* and *on the bench* are wanted in the query, as shown in Example 15. More details on the use of this query (in the simplified mode) and the POS tags are given in 5.4.

Non-contiguity Finally, another noteworthy possibility is the fact that words are not necessarily contiguous, as *lonely* was discarded although it appears between two selected words. This possibility is particularly interesting for syntactic constructions based on non-contiguous elements, such as the negation in French ("ne [...] pas", "ne [...] plus" etc.) or constructions such as "the more `A`, the more `B`"[9] which expresses the parallelism of the increase of `A` and the increase of `B`, and the implicit consequence of the former on the latter. Incidentally, searching similar constructions can be particularly useful on such expressions because the superlative in English is not always formed using the word "more", as we can see in the idiom *the more the merrier*.

(14)

	I	think	we	should	do	something	for	the	lonely	person	who	is
Token	○	○	○	○	○	○	○	○	○	○	●	○
POS	○	○	○	○	○	○	○	●	○	●	●	●

	sleeping	over	there	on	the	bench
	○	○	○	○	○	○
	●	○	○	○	○	○

(15)

the	person	who	is	sleeping
`DT`	`NN`	`who/WP`	`VBZ`	`VVG`

In our case, the morphosyntactic analysis on Korean does not only add POS tags but does a complete morphological analysis. Thus, users are not faced with their original input but with the input segmented into morphemes. However, the principle is similar to that of Poly-GrETEL in so far as the user has to indicate which morpheme (instead of word) may appear in the query, and how: in its lexical form (the token 프랑스 for 'France'), replaced by its POS (`NNP` for proper noun), or both used together as a pair (프랑스/`NNP`). This step is the only one in which

[9]Such expression also exists in other languages: among others, the French equivalent is "plus `A`, plus `B`" and in Korean a verbal ending is added on the verbs of the two phrases as in A*myen* A*swulok* B "A면 A수록 B")

the learner/user is exposed directly to POS.[10] In all other steps, POS are taken into account in calculations, but are not displayed.

5.3.4 Similarity Computation

At this point, the input is transformed into a *query*, and is therefore set to be compared to the sentences or utterances from the corpus, in order to retrieve those *similar* to it.

We are now at the core of our system. For this step, we need to define to what extent two sequences of tokens or POS can be called similar. This issue is not addressed directly by existing tools: concordancers only retrieve segments that *strictly match* the query, while phraseological tools allow some flexible search around words (namely, they allow *positional* and *constituency* variations, described in Section 4.3.2). However, none of them allow flexible search on syntactic constructions as such.

A measure adapted to non-specialist users In concrete terms, in order to study relative clauses with a concordancer, the user has either to use a query explicitly including the tag for relative pronouns[11], or to list them. For example, we stated in the introduction of this chapter that it would be impossible to retrieve the two phrases "the person whom I see" and "that dream that you had", without using a specific query like "`DET NOUN which|that|whom PRO VERB`" because they do not share any common lexical items even though they have the same syntactic structure. Using such a detailed query, relative clauses can be retrieved, but only

[10] At least in the "advanced version" of the tool. As mentioned in the introduction, the simplified version was designed for complete novice users and uses default settings and hides options. In this case, the user would only be asked to select the target word(s) which implicitly tells the programme that except this (these) word(s), every other word will be replaced by their POS.

[11] In the sense that relative pronouns have to be tagged, but not necessarily with a specific tag. In the CLAWS7 tagset for example, there is a distinction between `PNQO` "objective wh-pronoun" (*whom*) and `PNQS` "subjective wh-pronoun" (*who*) that does not exist in Treetagger's tagset for English. Using the latter, all wh-pronouns are tagged `WP` as shown in Section 5.4. Furthermore, in both tagsets, the pronoun *that* does not have the same tag as *which* although it is often possible to substitute one for the other. In French, relative pronouns are sometimes grouped together under a single tag, for instance `PRO:REL` in the Treetagger's French tagset. Korean does not have any relative pronoun.

those that match *exactly* the description, excluding for example "the jury, which was locked up". Incidentally, such query requires some knowledge both in linguistics (parts-of-speech) and in the query language (the use of the vertical bar, or 'pipe', as the symbol for the logic operation (*or*), i.e., the disjunction). Moreover, this knowledge is neither trivial nor general: on the one hand, how parts-of-speech are encoded depends on the tagset of the corpus, and on the other hand, how a disjunction should be represented depends on the query tool.

Conversely, an example-based tool such as GrETEL is based upon none of this knowledge. The query is a parse tree generated automatically by the tool from a natural language input. However, the matching system is the same as for a regular concordancer: GrETEL and Poly-GrETEL are enhanced concordancers allowing non-specialists to compare parse trees, but the retrieved trees also match *exactly* the one used in the query.

Search modes Using a similarity measure makes it possible to leave the binary representation of what does not match the query at all (absence or `0`) and what does match perfectly (presence or `1`). This opens the way to a wide range of possibilities. In fact, when using a similarity measure, everything matches the query in some ways, even a completely different segment. What we get with a similarity measure is a score between 0 and 1 (the score could be close to 0 but not equal to) or a distance[12]. In the latter case, the score would be low if the two segments are similar, and increasingly higher (with no boundaries) for each difference from the query.

Using our tool, the basic search, or **default mode**, is looking for a segment that is not exactly the same but which is similar to the input, both in terms of token and of context (i.e., sequence of parts-of-speech). However, we also made the most of these new possibilities by computing dissimilarities as well, and added 'modes' to allow other types of search: a distributional analysis-like search and a search on different usages. The differences between the three modes is shown in

[12]We used both similarity and distance measures in our experiments, see their definitions in Section 5.5.

Figure 5.2.[13]

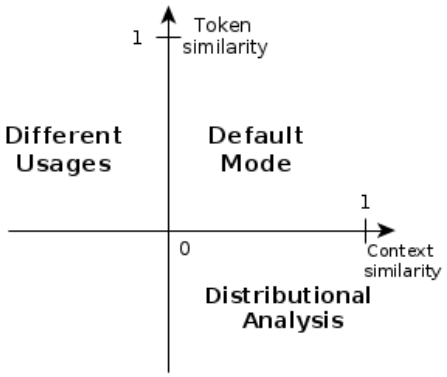


Figure 5.2: Illustration of the relations between the different modes

Distributional analysis search According to Figure 5.2, this mode is similar to the default mode in that it is based on the similarity between sequences of POS. However, in the default mode, the selected token(s) is (are) the one(s) that the user emphasised and which *must* appear in *all* output sentences. In other words, the user believes that the syntactic construction is centred on it or them. In the distributional analysis search mode, the selected token(s) is (are) excluded from the query.

This amounts to searching for another word or sequence or words that has the same *distribution* (i.e., position where a given word appears) as the one(s) excluded. This search mode is different from methods based on word embeddings such as word2vec (see for example Mikolov et al. [2013]) in so far as what we call the *context* of a given word is its *syntactic* distribution (here, a given sequence of POS), while word embedding techniques focus on *semantic* distribution. Indeed, word embeddings "capture the semantics of words by incorporating both local and global corpus context" [Desagulier, 2017, p.252]. Simply put, a given word

[13] A fourth mode was not implemented since it is difficult to conceive what a system that retrieves words that are different than those in the input *and* in a dissimilar context could be used for.

is represented by a vector defined by the plain words[14] appearing in its local and global context. The underlying idea is that "a word is characterised by the company it keeps", advocated by Firth [1957]. Words appearing in the same (lexical) context tend to be similar. Extending this method therefore theoretically allows to account for semantic relations such as homonymy (similar vectors of different words) and polysemy (dissimilar vectors of a given word).

The underlying idea of our system is similar, except that we assume that a word is characterised by its syntactic distribution and that words appearing in the same syntactic context tend to be similar. As a matter of fact, unlike distributional semantics, our system is bound to make use of *grammatical* or *function* words.

Different usages search If we go back to Figure 5.2, we can observe that this mode is somehow the opposite of the distributional analysis one: it uses the same selected token(s) but is based on a dissimilarity of context.

Like in the default mode, output sentences have to contain the selected token(s) to be relevant. However, in this mode, the context should not be similar but dissimilar. In other words, this mode allows to search for contexts in which the target token(s) appear(s), but other than that in the input.

The use of similarity (and distance) measures in Natural Language Processing and the measures that we selected for our experiments are described in the following sections. As for the search modes, they are illustrated in Section 5.4.

5.3.5 Ranking and clustering

As we mentioned previously, if we use a similarity measure instead of a strict matching system, everything matches the query to some extent. With such a wide range of possible matches, ranking becomes a critical step in the processing chain.

[14] Plain words, or *lexical words*, are opposed to *function words* or *grammatical words*. In many NLP applications involving semantics, grammatical words, as well as some lexical words, are considered non-significant because of their high frequency and low semantic implications. In these cases, they are put in a "stop word list" and filtered out in a preprocessing. This list may include articles, conjunctions, prepositions, auxiliary verbs and lexical words that do not discriminate the samples within a given corpus [Luhn, 1960].

Sorting option Indeed, the ranking of matching segments is presently not implemented in current concordancers. The concordance page (i.e., the output of a concordancer) is a list of all matching segments, usually spanning several pages depending on the number of results. Contrary to what we might expect, the first match is not necessarily the most relevant, since each of the segments in the concordance page *does* match the query *perfectly*. The matching segments are presented in an order determined by the order in which they were encountered in the corpus. The only function available to change this classification is the sorting option: the user is given the possibility to rearrange the concordance lines alphabetically, upon words in a given position in the context. Usually, users sort on the target word itself (t_0): if the query was set on the different forms of the verb BE for instance, matching sentences would contain first *am*, then *be*, *being*, *been*, *is*, *was* and then *were*. They may also sort on the left word (t_{-1}) or on the right word (t_1), or further away from the target word (t_2, t_3 etc.). The range of the sorting options depends on the possibilities given by the tool, as well as on the range of the context selected by the user at the beginning of the query.

Ranking in search engines However, sorting is a very distinct function from ranking, for two reasons: it does not rely on relevancy, and all concordance lines are still equal. In this respect, our tool is closer to information retrieval systems. The most revealing example of information retrieval systems for which ranking is essential is that of search engines. One of the main difference between a concordancer and a search engine is that the former retrieves segments matching the query in a corpus, while the latter searches through a corpus as well (be it the Web or a specific database such as corpus of scientific articles), but retrieves documents instead of segments. This difference implies complex calculations: if the query is composed of three keywords, considering their rareness and the size of the corpus, there is still a high probability that those three words appear in a substantial number of documents. The relevance of a document can be computed according to the answers to the following questions:

- does the document have the highest ratio of these three words compared to the whole corpus?

- does the document have the highest number of occurrences of the first word? (which implies that keywords are ranked and have different weight)
- do the keywords appear in important parts such as the title of the document?
- do the keywords appear in the same order as in the query? If so, is the query a contiguous sequence of words (n-grams) or not (skipgrams)?

There are also times when the chosen keywords do not suit exactly what the user had in mind. In these cases, is it relevant to retrieve documents containing an inflected form of the keywords? If the query contains the form "draw" for example, should related verbal forms such as "draws", "drew" or "drawn" appear? What about the ambiguous form "drawing", and the unambiguous noun "drawings"? This process, called **query expansion**, can also use resources and replace certain keywords with their synonym(s). Each of those questions and many others[15] are parameters that have to be taken into account in the ranking algorithm of the search engine.

To sum up, ranking is an important step in a search engine system because a search engine is valuable if its *precision*, a popular metric based on the number of relevant documents out of all retrieved documents is high. Indeed, users are usually looking for a specific information and do not seek exhaustiveness on a given subject.[16] In Rose and Levinson [2004]'s hierarchy of search goal, only one type of search – undirect navigational search – actually seeks a certain exhaustiveness on a subject. In this peculiar case, the user only inputs a topic, without any further specification. Incidentally, exhaustiveness is often not reachable considering the tremendous amount of documents in a database, especially if the database is the web.

Although our system deals with segments and not documents, ranking is as crucial as for search engines but for different reasons. Considering the variety and

[15]We intentionally left issues such as the commercial aspect of ranking in web search engines, as well as the PageRank [Page et al., 1999], based on the number of hyperlinks in and out of the page and used as a clue on the popularity of a webpage. These parameters certainly have a significant weight in the page ranking algorithm.

[16]This also explains why users generally do not go beyond the first pages of results, even though the results on the 21st page might also contain relevant documents. If they find what they need in the first page(s), there is no reason to go further.

the high number of possible constructions as well as the subtleties of language, it is also crucial to set a boundary to similarity, for example, by setting a threshold to the score. All the more so as matching segments may be similar in very different ways. In order to spare the users from browsing through an overwhelming result page of heterogeneous matches, we propose to group them into *clusters* by running another round of similarity measure within the relevant matches: the number of clusters is limited but not set in advance, and each cluster is represented by a single instance, the most representative one. Unfortunately, we did not have time to implement this functionality. Further details on the method that we propose are given in the perspectives in Section 7.2.1.

5.3.6 Query Refinement

This sixth step of our system is also the third where the user is asked to make a choice. Contrary to Step 3 (Query Reformulation) that the user could simply skip and carry on the search with the default settings, this time, the user *has* to indicate which of the proposed clusters is the closest to their expectations. This participation is limited to spare the user effort and time, yet necessary to keep the user active in their search, in compliance with the Data-Driven Learning theory.

Once a cluster is selected, the user can either validate this choice *definitely*, or refine the query. On the one hand, the first option leads the tool to proceed to the final output in Step 7 and all matching segments of the selected cluster will be displayed. On the other hand, the second option allows the user to refine their query indirectly: matching segments from other clusters are discarded, but the ones from the selected cluster are used to run another round of similarity measure. This step is therefore recursive[17], in that a similarity measure can be computed as long as the cluster has a high number of items, and as long as the user wishes to do so. This recursive indirect query refinement is an implementation of *relevance feedback*.

relevance feedback

The whole system is based on both the user's intuition and on the data, so the results are rather unpredictable. Because we want to encourage a free exploration

[17] There this step is represented as a loop in Figure 5.1.

of the tool, every choice in this step is 'undoable': the user has the possibility to 'undo' the selection or the validation of a cluster, and to make another choice after seeing that their first choice brought unexpected results. This possibility seems trivial but is actually fundamental for Donald Arthur Norman's concept of affordance described in Section 4.4.1. In terms of computation, this possibility implies that intermediate results must be stored to be re-established if needed.

Incidentally, it may happen that the representative example of the cluster matches the user's expectation but not the rest of the cluster. In this case, the user is given another possibility: to save the example and create a new query with the initial input *and* the saved example together.

5.3.7 Final Output

Eventually, the final output of our system is similar to the final output of current concordancers. Once they validated one of the propositions of the tool, the query refinement process stops and all matches from the selected cluster are displayed. This time, no ranking is available but considering the recursive query refinement step, all matching segments in the final output should be relevant to the user's search.

5.4 Illustration: the Relative Clause in English

In order to illustrate the use of our tool by non-specialist users, let us see a concrete case study on the acquisition of relative clauses in English as a foreign language.

Using relative clauses in English may be difficult for a foreign learner, especially when there is no such structure in their native language. Before any attempt to produce relative clauses, the learner has to understand how they are used, i.e., in which context they appear. A first solution would be to ask a profesionnal, preferably the teacher if the learner attends English classes. Or, for want of a better alternative, the learner may ask a native speaker, who is considered reliable on grammaticality judgements for introspective methods, but is not necessarily capable of delivering an explanation on a specific phenomenon. Another solution

would be to look up a grammar of English or a language textbook. In any case, the learner is likely to receive a concrete illustration (perhaps a fictive dialogue in which he or she is asked to take part), an explanation or an explicit rule, and/or exercises to practice what they just learned.

In case the two above-mentioned solutions do not satisfy the learner's need, we believe that looking up more relative clauses produced in a naturally-occurring written or spoken context can help the learner understand their usage(s). All they have to do is to provide our tool an example of a relative clause, for example, copied from their textbook or from what they heard from the teacher or a native speaker. This concrete example is shown in Figure 5.3, which was built for the sake of illustrating our processing chain, but the sentences used as output are authentic examples extracted from the Brown Corpus [Francis and Kucera, 1979], a corpus of American English of 500 samples of roughly 2,000 words distributed across 15 genres published in 1961.

We have established in Chapter 4 that a complex interface and processing repels novice users and prevents them from using useful corpus exploration tools. In order to cope with the possitibilities and options that we want to provide without raising unnecessary cognitive load, we designed a processing chain that can be either advanced for intermediate and expert users, or simplified for novice users. Since this section illustrates the simplified processing, all the possibilities mentioned previously in the step-by-step processing (Section 5.3) do not appear in the following description.

Step 1: User input. The learner gives an example containing a relative clause in English, in this case the noun phrase `the person who is sleeping`.

Step 2: Automatic syntactic analysis. An automatic morphosyntactic tagger is used to annotate this phrase with POS. The input is therefore transformed into `the/DT person/NN who/WP is/VBZ sleeping/VVG`. For our example, we simulated an error-free annotation of Treetagger [Schmid, 1994]. The English POS

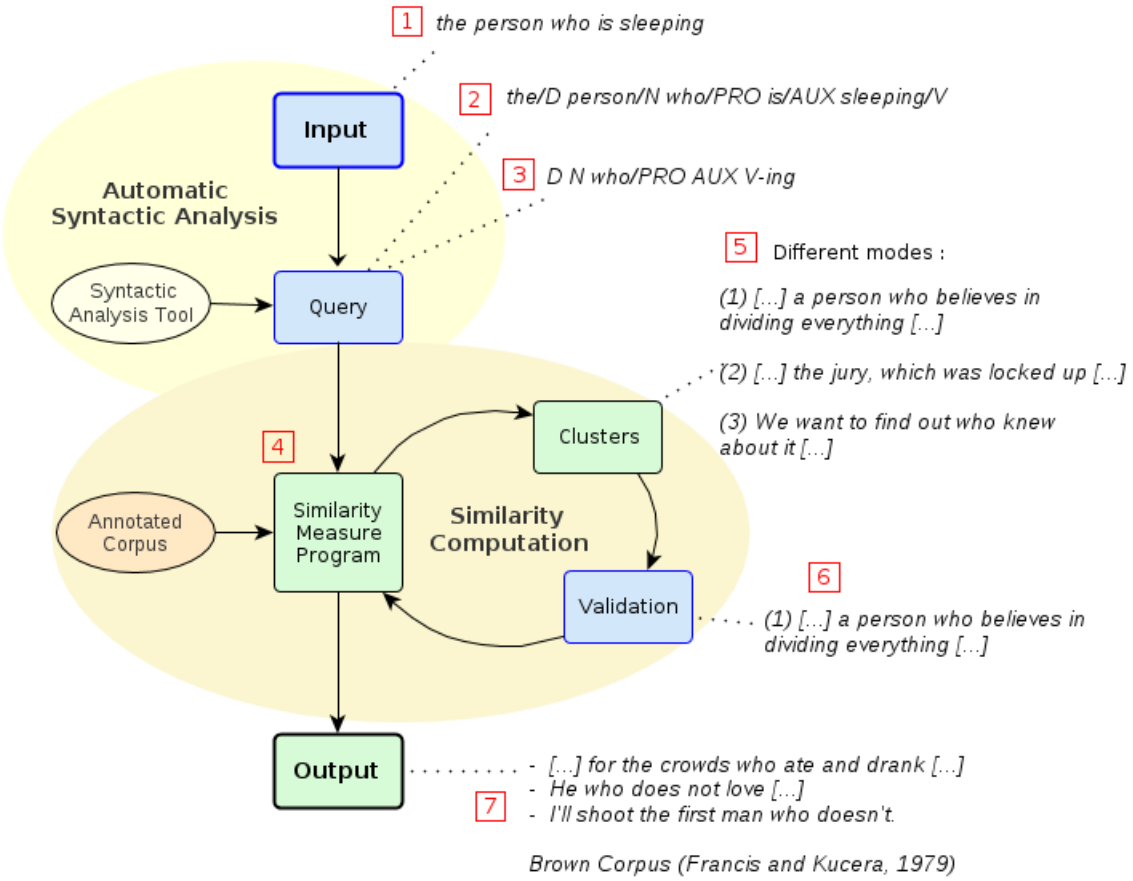


Figure 5.3: Process flowchart of an example of syntactic similarity research in English

tagset used by Treetagger is partly described[18] in Table 5.1.

Step 3: Query formulation. The newly-annotated input is showed to the learner who chooses what seems to pertain to the syntactic construction to be investigated. In Example 16, we simulated the choice of the relative pronoun "who", the only word with a black button underneath (unselected buttons are white). This simulated table-like interface is similar yet a little simpler than that of PolyGrETEL's third step (see Figure 4.9). The idea was indeed inspired from Augusti-

[18]Only the POS necessary to understand our example were copied in the table. The full tagset is available at `https://courses.washington.edu/hypertxt/csar-v02/penntable.html` or in the detailed guidelines [Santorini, 1991].

POS Tag	Description
DT	determiner
IN	preposition/subord. conjunction
NN	noun, singular or mass
PP	personal pronoun
RP	adverb
VBD	verb *be*, past
VBZ	verb *be*, pres 3rd p. sing.
VV	verb, base form
VVG	verb, gerund/participle
VVN	verb, past participle
VVP	verb, present non 3rd-p.
VVZ	verb, present 3rd p. sing.
WP	wh-pronoun
:	general joiner (;, -, – etc.)

Table 5.1: Selection from the English POS tagset used in Treetagger

nus et al. [2012]'s work on GrETEL. In the original tool, each token is aligned with buttons (technically called *radio buttons*), allowing the user to select one of the representations proposed (between 'word', 'lemma', 'word class'[19]). In our example, we aligned each token with a *single* bullet. This simplification narrows the possibilities offered by query formulation step, but facilitates its apprehension by novice users.[20]

(16) the ○ person ○ who ● is ○ sleeping ○

The pronoun is therefore kept untouched, with its wordform ("who") and its POS (WP, which stands for wh-pronoun). The rest of the words in the phrase is

[19]"Word class" corresponds to part-of-speech in GrETEL's purposely simplified terminology.

[20]However, a more complex table allows an interesting range of possibilities and is certainly suitable for the advanced version, reserved to more experienced users, that we mentioned previously. The options of the advanced version are described in Section 5.3.3.

replaced by their respective POS. The query is defined as follows: `DT NN who/WP VBZ VVG`.

Step 4: Similarity comparison. The newly-defined query is compared to all sentences from the corpus (here, the Brown Corpus), one by one. Corpora used in our system do not need any prerequisite except to be presegmented in sentences and have the same annotation format as the one from the automatic syntactic analyser.

Step 5: Ranking and clustering. The most *similar* sentences to the query are kept and ranked according to their score. For the sake of illustrating the different 'search modes' that we mentioned in Section 5.3.4, we manually searched in the Brown Corpus each of the examples shown in Figure 5.3 and detailed in the Examples 17, 18 and 19. Each of them represent a different option since all of them are therefore highly similar, but in different ways. In an actual search using our system, the clusters would be calculated based on one of the modes, not the three of them (see, for example, the suggestions of clustering for *-(u)lo* -(으)로 in the results of Section 6.3.2).

Example 17 illustrates the **default mode**, i.e., the search for a similarity computed based both on the context, and on the relevant word(s) selected by the learner in Step 3. Indeed, we can note that the word "who" appears with the correct POS (`WP`) and the sequence of POS is similar. The first three POS (`DT`, `NN` and `WP`) are exactly in the same position, and a fourth similar POS appears in another position (`VBZ`), but still in the same order.

Example 18 illustrates the **distributional analysis** search mode. When using this mode, the learner chooses to see other words that may appear in the context of the input, but different from the one selected. The first step is therefore to discard all sentences containing the selected word(s). Only after this, the similarity can be computed, based on the context. It can be noted, again, that the context is similar: `DT`, `NN` and `WP` appear in both the query and this example, even though a minor punctuation mark, called "general joiner" in the description of the tagset in Table 5.1, appears in the position of `WP` (third) in the query. More importantly, the

word "who" does not appear, but its position is filled by another relative pronoun, "which". We can imagine that other relative pronouns such as "that" could also appear here.

(17)	a	person	who	believes	in	dividing	everything	[...]	
	DT	NN	WP	VVZ	IN	VBZ	NN		
(18)	the	jury	,	which	was	locked	up	[...]	
	DT	NN	:	WP	VBD	VVN	RP		
(19)	We	want	to	find	out	who	knew	about	it	[...]
	PP	VVP	WP	IN	RP	WP	VVD	IN	PP	

Example 19 illustrates the search mode of **other usages of a given word** so that, this time, the preliminary step eliminates all sentences that do not contain the selected word(s). At opposite ends of the distributional analysis search mode spectrum, the similarity computation is based on the dissimilarity between the query and the sentences from the corpus. On the one hand, Examples 17 and 18 both contain basic relative clauses, respectively *restrictive* and *non-restrictive*. Example 17 can be interpreted as "Only a person who believes in dividing everything [...]", and bears a restriction on the noun *person* while Example 18 can be interpreted as "The jury, which happened to have been locked up [...]". On the other hand, Example 19 contains a *free relative clause* in that *who knew about it* has no antecedent, is not even preceded by a noun but a verb, and actually has the same distribution as a noun phrase (compare with *we want to find out the answer*). Indeed, in terms of distribution, the difference between these two occurrences of "who" is big enough to question the analysis of "who" as a relative pronoun. In Example 19, "who" can be substituted by the pronoun "whoever", which is not possible in Example 17, nor in Example 18 with "whichever".

Step 6: Query refinement or validation of the proposition. As mentioned in Step 5, the simplified version illustrates all search modes: each of the three examples proposed is supposedly the most similar to the query with regard to their respective mode.

Out of the three examples proposed, the learner selects and validates the first. By doing this, they implicitly choose the first mode and show that they need more examples of relative clauses with "who" as the relative pronoun. From this point

on, clusters are computed on the basis of the first mode recursively, until the user decides that the refinement is satisfying.

Step 7: Final output. All other matches of the cluster represented by the segment "a person who believes in dividing everything [...]" appear in the concordance page. Only segments of sentences appear in our example, but the representation is the same as for current concordancers: a KeyWord In Context display, centred on the target word(s), but with an extendable context. The proportion of context which can be shown depends both on the accessibility to the corpus, and on the type of exploration tool: third generation concordancers (like AntConc, for which the context in fully available because the corpus is stored on the user's computer), and fourth generation concordancers (like the BYU Corpora website which allows only restricted snippets of the context because of the copyrights on corpora, which are stored on an external server[21]).

5.5 Similarity Measure(s)

Similarity is a familiar concept of everyday life. Based on works from psychology, Schwering [2008] introduces similarity judgement as "probably the most central construct in human cognition". Humans use similarity both unconsciously and consciously: we constantly compare new experiences and items to old ones, we naturally and instantly bound with people with similar interests and when learning a new language, we memorise cognates faster than unrelated words[22] and invariably fall into the trap of *faux-amis*. Similarity is strongly related to *categorisation*: similar items are grouped together in such a way that the degree of dissimilarity between groups must be stronger than between items of a same group.[23] In social categorisation, this behaviour tends to be emphasised: "between-group differences are accentuated" whereas "within-group differences are minimized" [Liv-

[21] See 4.2 for a brief description on the difference between third and fourth generation corpus exploration tools, and McEnery and Hardie [2012] for a full development.

[22] Cognates are indeed a bridge into a new language and it is advised to introduce cognates early in a language course to foster vocabulary enhancement [Nation, 2001].

[23] This is also the definition of clustering. What statisticians call "categorisation" is clustering in computer sciences.

ingston et al., 1998].

Similarity measures are fundamental in numerous and various domains, and their diversity may be as considerable as the number of their applications. Our objective in this section is not to give an exhaustive overview of similarity measures and their uses, nor is it to explain and compare their mathematical properties, since this would be beyond our competence. We simply aim at giving the reader the keys to understand how and why similarity measures are used in Natural Language Processing (NLP) and Information Retrieval (IR), as well as in the query system that we worked on.

For a detailed presentation of similarity measures, we recommend the work of Bandyopadhyay and Saha [2012], which gives an introduction to clustering as suggested by its title, as well as to pattern recognition. The chapter introducing similarity measures is mainly based on an online tutorial freely accessible[24].

As our work aims at computing the similarity between two sequences of words, similarity measures are indisputably needed. Among the measures, those which are interesting for our system have to meet the following properties:

1. to be an interpretable measure;
2. to be efficient even on short size items (as broad as a sentence and as short as a segment);
3. to take word order into consideration;
4. to be applicable to (tree-)structured data.

The first property allows a better understanding of the efficiency of the measure as well as an interpretation of the results. More importantly, it also provides a means to adjust calculations in order to obtain more satisfying results. The second property is linked to the possibility that we give the user to provide a contiguous segment shorter than a sentence (such as a phrase). The similarity measure should thus be efficient with a low number of words. Since syntactic construction

[24] http://people.revoledu.com/kardi/tutorial/Similarity/index.html

rely both on a hierarchical order (vertical) and on a linear order (horizontal), the query should not be considered as a 'bag of words' (defined in Section 5.2). The third property therefore ensures that words are considered with their immediate (preceding or following) context. Eventually, the last property is essential because syntactic annotations are not limited to POS. For further experiments, we may use a syntactic parser (either constituency-based or dependency-based) instead of a morphosyntactic tagger, which would result into parse trees.

In this section, we start by defining the concept of similarity and its counterpart, distance. Then, we focus on the use of these measures in text mining and information retrieval, which we illustrate with some examples.

5.5.1 Definitions

While two identical items should be identified steadily with ease, "the more similar two stimuli are, the more likely they are to be confused" [Medin et al., 1993].

Same-different judgement A large number of pairing games play with this principle, such as the UNO card game (a shedding game based on the colour and the rank of cards from an original deck), the Jungle Speed game (revolving around matching cards with similar identical symbols), or even the Mahjong solitaire[25] (whose objective of matching identical tiles as quickly as possible is hindered by the similarity of tiles from the same 'family' and by rule of removing only 'exposed' tiles) or memory card games (which shares the same objective as the Mahjong solitaire with the additional difficulty that cards are laid face down and can only be flipped up two by two). The more the cards (or tiles) are similar, the more challenging the game is.

Perception of similarities If we take a close look at a Jungle Speed deck, we notice that all cards share some similarities (and differences): in shapes, in colours, in symbol orientation. According to Medin et al. [1993, p.254] referring to Goodman

[25] We refer to the tile-matching game, which is *not* a variant of the Mahjong game. The two games are only related because they use the same 144 tiles.

[1972]'s work, "the similarity of A to B is an ill-defined, meaningless notion unless one can say 'in what respects'[26] A is similar to B." and that "[j]ust as one has to say what something is moving in relation to, one also must specify in what respects two things are similar". Furthermore, same-difference judgements are not only perceptual. Judging the similarity between items also involves higher cognitive functions: memory and attentional mechanisms. "When viewing an item, subjects may attend to some aspects and ignore others. Contextual, instructional, and motivational variables may influence what stimulus attract attention" [Goldstone, 1994, p.179].

The similarity between two items is defined by the features they share. Incidentally, two items may be similar if we consider certain features, but dissimilar if we consider other features.

Similarity in word relationships Relationships between words based on pronunciation, spelling and meaning are interesting in that matter. *Homographs* "close" (the verb) and "close" (the adjective) are similar in their written form but dissimilar in their pronunciation; conversely, *homophones* "carat" and "carrot" are both pronounced [karət][27] but are dissimilar in their written form; *homonyms* "fly" (the insect) and "fly" (the verb) are similar in both their written form and their pronunciation, but still dissimilar in meaning; conversely, synonyms "mortal" and "lethal" are similar in meaning but dissimilar in both their written form and their pronunciation. We can also note that "human being" may be a synonym of "mortal", but not "lethal" because the meaning they share is not the same. When comparing two (and more!) objects, it is therefore essential to understand *what* we want to compare, and *which feature(s)* is (are) relevant, as the number of shared features may be less important than their quality.

Similarity in our query system Our own system provides another practical example. We defined earlier in this chapter that in the main mode of our system, similarity is based on two features: first, on the word(s)[28] selected by the user

[26] Quotes in the original text.

[27] According to the Merriam-Webster dictionary, both "carat" and "carrot" can also be pronounced ['kerət].

[28] Or based on the morpheme(s), depending on the granularity of the segmentation.

in the query formulation step, and secondly, on the context. Indeed, a sentence is considered similar to the query if the context (i.e., the sequence of morphosyntactic tags) is similar. However, the very first condition is that it contains the word(s) the user specified as relevant for the target construction. In this mode, a sentence which does not contain the target word(s) is automatically discarded. Said shortly, every word in the query is used as a feature for the similarity computation, and the presence of the target word(s) is the most important one.[29]

Distance The notions of similarity and distance are inversely defined: the greater the similarity, the smaller the distance, and vice-versa. This relation is mathematically represented as:

$$s_{(x,y)} = 1 - D_{(x,y)}$$

with the distance D assumed to range from 0 to 1.

A distance satisfies the following properties mentioned by Teknomo [2015] cited in Bandyopadhyay and Saha [2012]:

1. Non-negativity: $d_{x,y} \geq 0$

   → the distance between two objects should be always positive (or zero).

2. Coincidence axiom: $d_{x,x} = 0$

   → a distance is zero if and only if it is measured with respect to itself. In other words, $d_{x,y} = 0$, if and only if $x = y$ [Niemytzki, 1927].

3. Symmetry: $d_{x,y} = d_{y,x}$

   → a distance is symmetric.

4. Triangular inequality: $d_{x,y} \leq d_{x,z} + d_{y,z}$

   → a distance should satisfy the triangular inequality, which states that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side.

[29] Precisely, in similarity computation, the target word(s) do(es) not act as a weight but rather as a pre-selection condition. This decision ensures a better precision of the system, since a sentence that does not contain the target word(s) is by all odds irrelevant.

5.5.2 Applications

That similarity measures have a wide range of applications, is partly due to the fact that they can be applied to various data types. Yet, their application is not totally free but rather subject to certain conditions. Some similarity measures only work with a certain type of data and possibly require transforming in the representation of the data set.

Possible data sets include:

- textual data of various size;
- image data;
- audio data (signal);
- spatial and temporal data.

Data type This variety of data illustrates the variety of problems that similarity measures can address. Let us mention, among similarity measures for processing **textual data**: the computation of synonyms and antonyms (using respectively semantic similarity and dissimilarity between word meanings), exploration of databases through search engines, using the similarity between a query (single or set of words) and a document (large set of words), as well as the classification of documents by topic or author (using the similarity between two documents) and plagiarism detection (using a strong degree of similarity based on style rather than meaning, still between two documents).

Applied to **images**, similarity measures are useful to a specific field of research in pattern recognition: Optical Character Recognition (OCR). OCR is a text (pre)processing task which consists in *reading* images, i.e., in transforming an image containing text into raw text. While the human brain is able to read words on the pages of a book, for a machine, scans or photographs of books – even printed – are only images.[30] The regularity of printed characters makes them quite

[30]Incidentally, this characteristic is the underlying principle of CAPTCHA whose backronym is "Completely Automated Public Turing test to tell Computers and Humans Apart", the widespread Turing test found in numerous websites to counter the use of bots. CAPTCHA are distorted words or sequences of letters and/or digits that users have to type to prove that they are indeed humans.

easy to recognise, but the same cannot be said for handwritten texts, let alone calligraphed manuscripts with illuminations. We have also mentioned in Section 4.3.2 that OCR is affected by the quality of the physical document as well as the quality of the image (e.g. resolution, contrast).

Finally, we can note that comparing **audio data** can be used, for instance, for music recognition services. Systems such as the software Shazam[31] are used to identify an audio sample captured by a smartphone. An *acoustic fingerprint* of the sample is created and compared to the songs and musics from an audio database. Like our query system, this service is also example-based: users do not define a query explicitly but provide a built example. Unlike our system, the service only outputs the highest-ranking match[32], since users expect to find *the* matching song. Nonetheless, similarity measures are essential to music recognition services because strict matching is out of reach. The quality of the sample is inevitably altered by the following factors: the output device broadcasting the music (e.g. loudspeakers), the input device of the user (e.g. the built-in microphone of the smartphone), as well as the noise surrounding the recording (e.g. human voices, traffic noise, other music).

In this present work, we only use textual data, included for the Sejong Spoken Corpus, which only consists in transcriptions of spoken samples. Unfortunately, we did not have access to the audio files of those transcriptions. In our experiments, we adapted two similarity measures which are more commonly used in information retrieval: the method is the same but instead of considering two documents as two vectors of words, we reduced the scope and considered two sentences as two vectors of words.

Similarity Measures in our Experiments The similarity measures used in our experiments are the Jaccard and the Sørensen-Dice coefficients (or indexes), two similarity measures commonly used in information retrieval to compute the proportion of common terms between two documents or, in other words, the overlap

[31] www.shazam.com

[32] The score is computed from "the number of matching points", matching *hashes* that also have a similar relative time sequence [Wang, 2003].

between two documents.

The Jaccard coefficient of two sets A and B is defined as

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

and corresponds to the overlap or *intersection* ($|A \cap B|$) of the two sets A and B compared to the *union* ($|A \cup B|$) of the two sets. $|A|$ is the cardinality of the set A, i.e., the number of elements in A. $|A \cap B|$ is therefore the number of elements of the intersection of A and B.

The intersection of two sets is illustrated in Figure 5.4: terms that are common to both sets, the purple area in the figure, are said to be at their intersection. In this figure, the union of the sets A and B corresponds to the whole coloured area, whether red, purple or blue.

Indeed very similar to the Jaccard coefficient, the Dice coefficient is defined as:

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

The difference between the Jaccard and the Dice coefficients lies in the fact that the intersection of A and B is divided by the union of the two sets for Jaccard, while for Dice, it is divided by the sum of the two sets, which counts the intersection twice.

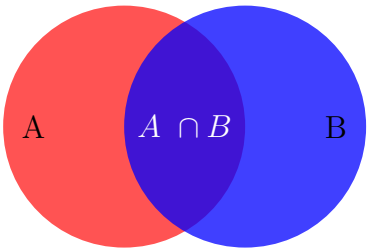


Figure 5.4: Venn diagram illustrating the intersection of two sets A and B

Both the Jaccard and the Dice coefficients are used to compute the overlap of elements from two sets – in our case, two annotated sentences. Each sentence is represented by a vector of words, which does not take word order into account,

hence resulting in a 'bag-of-words' representation of the sentence.[33]

The two measures range from 0 to 1, and are equal to 1 if and only if A = B and Jaccard(A,B) $\leq$ Dice(A,B) because $|A| + |B| = |A \cup B| + |A \cap B|$ *leq* $|A \cup B|$.

Let us consider two annotated sentences in English represented by the following vectors A and B.[34]
A = { `I_PNP, left_VVD, everything_PNI, like_PRP, it_PNP, was_VBD` }
B = { `I_PNP, know_VVB, what_DTQ, it_PNP, feels_VVZ, like_PRP, now_AV0` }

The intersection between the two sets A and B are the words that are common to both sets, represented as:
$A \cap B$ = { `I_PNP, like_PRP, it_PNP` }.
As for the union, it contains all words from A and B:
$A \cup B$ = { `I_PNP, left_VVD, everything_PNI, like_PRP, it_PNP, was_VBD, know_VVB, what_DTQ, feels_VVZ, now_AV0` }.

The Jaccard coefficient Jaccard(A,B) and the Dice coefficient's Dice(A,B) are therefore calculated as follows:
J(A,B) = 3 / 10 = 0.3
D(A,B) = (2*3 / 6+7) = 0.461538462

5.6 Edit Distance as a Dissimilarity Measure

Edit distance, also referred to as the *Levenshtein distance*, is the most popular (dis)similarity measure for textual data. Contrary to the previously mentioned metrics, edit distance considers textual variables not as boolean nor vectors but as *strings*, i.e., as sequences of *characters*. The national motto of France, "liberté, égalité, fraternité" (*liberty, equality, fraternity)*, has only three words ("`liberté`", "`égalité`", "`fraternité`") separated by commas but as a string it contains 29

[33] See the definition of 'bag-of-words' in Section 5.2.

[34] These two sentences are tagged using the Free CLAWS tagger with the CLAWS5 tagset, and not the CLAWS7 as we did before such as in Section 3.4.4 or 7.2.1. The first sentence was used as a query in the experiments on English presented in Section 6.4 and the second sentence was retrieved in the version of the BNC that was tagged with the CLAWS5 tagset.

characters: 25 letters (including "é", which is indeed a single character although it is composed of the letter "e" and its diacritic, the acute accent), the two commas, as well as the two whitespaces.

Definition of Characters (computing) We have roughly defined the notion of character in Section 4.2 when we mentioned the original ASCII character set as comprising lowercase and uppercase letters of the English alphabet, numbers from 0 to 9, punctuation symbols, and control characters, i.e non-printable characters originated from typewriter systems. Since then, a large number of character sets have been defined (some still based on ASCII) and word characters are not limited to letters of the English alphabet. Naturally, word characters can also be letters of other alphabets (e.g. Arabic, Cyrillic, Greek, Hebrew, Syriac), syllables from semi-syllabaries and syllabaries (e.g. Zhuyin fuhao (注音符號), a transcription system used for standard Mandarin Chinese[35], hiragana and katakana for Japanese, the Cherokee syllabary), as well as logograms (e.g. Egyptian hieroglyphs, sinogram[36]). What they all have in common is that they are treated as *units*, even though they may be split into smaller units.

Korean 'Characters' (computing) The case of Korean is interesting in that Korean has an alphabet called *hankul* 한글, and each of the letters from the alphabet can be considered as characters if they are isolated. However, when not used for themselves, they always appear arranged in a configuration corresponding to a syllable. For instance, the word *hankul* 한글 is composed of two syllables and therefore two characters, *han* 한 and *kul* 글. *han* 한 is decomposable into three letters, *h* ㅎ, *a* ㅏ and *n* ㄴ, while *kul* 글 is decomposable into three other letters, *k* ㄱ, *u* ㅡ and *l* ㄹ. In comparison, a syllable from a syllabary such as the hiragana character *ne* ね is not decomposable into *n* and *e*. It is not possible either to retrieve a hypothetic form corresponding to the 'letter' *e* when comparing the strokes in *ne* ね to the ones in *ke* け .[37] As a matter of fact, the syllable *e* is え

[35]The Zhuyin was introduced by the Republic of China and later replaced by Hanyu Pinyin but it is still in use in Taiwan

[36]Also known as "Chinese characters". Sinograms are called *kanji* in Japanese, and *hanca* 한자 is Korean.

[37]The hiragana syllabary was actually developed from Man'yōgana, an ancient writing system which uses sinograms for their phonetic rather than for their semantic qualities. There is

.

It is also argued that *hankul* 한글 is not an alphabet but an alphabetic syllabary (or an *alphasyllabary*) because it has unique properties [Pae, 2011; Taylor and Olson, 1995] it does not share with alphabets.

1. *Hankul* 한글 is not written in a linear form (from left to right or right to left) but in *blocks* corresponding to syllables: as we noted, the syllable *han* 한 is written with three letters, *h* ㅎ, *a* ㅏ and *n* ㄴ. Each of them occupies a specific position: the onset *h* ㅎ is in the top-left corner, the vowel *a* ㅏis in the top-right corner, and the coda *n* ㄴ occupies the lower space. The underlying reason is that *hankul* 한글 *kul* 글 has a specific reading direction: from the top to the bottom and from left to right. The syllable *kul* 글 illustrates the other possibility, with the onset, the vocalic nucleus and the coda aligned from top to bottom.

2. A second property, linked to the first, is that graphemes must be bound together into syllables, making the boundaries between syllables very distinct.

3. Pae [2011, p.107] adds that the use of spaces is different from "alphabetic languages" such as English, and Korean. While the former puts spaces between each 'word', the latter puts spaces after grammatical markers (e.g. *i/ka* 이/가 marks the subject, *ul/lul* 을/를 marks the object) attached to base words. However, we believe that this last orthographic argument has little relation to the syllabic structure of Korean. This argument may be tested against languages other than Korean and English. For instance, on the one hand, Finnish is known for its grammatical case marking, but there is no spaces between the base word and the case marker either (e.g. *lomalla* where *loma* is the base word for "vacation" and *-lla* is the locative case marker), although Finnish has an alphabet. On the other hand, Japanese has two syllabaries and grammatical markers similar to Korean ones, but does not make use of any space in their writing system other than after a punctuation mark.

therefore an undeniable link between *graphemes* and *sounds* (compare 倍 and 陪, two sinograms from Man'yōgana used for *ne*) but this system is not alphabetical for phonetic components of sinograms are by no means letters.

5.6.1 String-based Edit Distance

The edit distance $d_{(x,y)}$ between two given words (or sequences of inseparable units) x and y is "the minimal cost of a sequence of *operations* that transform x into y"[38] [Navarro, 2001]. The Levenshtein distance is based on three operations:

- insertion: cue → c**l**ue;
- deletion: de**s**sert → desert;
- substitution: **a**llusion → **i**llusion.

Each operation is associated with a *cost* and the sum of the costs of each individual operation is a *path* (see the green-coloured path in Table 5.2). One of the most notable uses of edit distance is the correction of misspellings and typing errors widely used in search engines (Google's "did you mean x?"), as well as directly in spelling checkers. Suggestions are ranked in ascending order of distance, which means that the best candidate is the one with the shortest distance to the input.[39]

Let us consider the strings "`france`" and "`ireland`". The distance $d_{(france,ireland)}$ from "`france`" to "`ireland`"can be computed as follows:

1.	substitution	`f` → `i`	(`i r a n c e`)
2.	insertion	`e`	(`i r e a n c e`)
3.	insertion	`l`	(`i r e l a n c e`)
4.	substitution	`c` → `d`	(`i r e l a n d e`)
5.	deletion	`e`	(`i r e l a n d`)

The proposed path is only one of the shortest possible paths but others are possible, either with the same number of steps or with more. However, since it is not possible to transform "`france`" into "`ireland`" using less than 5 operations, the Levenshtein distance between the two strings is therefore $d_{(france,ireland)} = 5$, if we set the weight of the three operations to 1. We can therefore define the

[38] Italics from the original text.

[39] In the case of spelling checkers, criteria such as the physical distance between two letters on the keyboard and the frequency rate of letters in a given language are also taken into consideration.

Levenshtein distance as the minimum sum of the number of operations weights. Incidentally, the longest path would be to delete all characters from "`france`" and add all characters from "`ireland`". In this case, the longest distance would be 13.

Step-by-step dynamic algorithm The following algorithm is a basic implementation of the computation of the minimum edit distance between two input strings. The computation consists in drawing a table with the cost of all the possible operations concealed.

Let x and y be the two strings that we want to compare and $d_{(x,y)}$ the minimum edit distance between them, i.e., the minimum number of operations necessary to transform x into y. The computation of the minimum edit distance consists in drawing a table T in which the calculations are represented and *cost* is the variable used to store the cost (i.e., the minimum number of operations) necessary to go from x to y. T is filled row by row, from left to right.[40] The edit distance for $d_{(france,ireland)}$ can be dynamically computed with the following step-by-step processing. In this example, all operations cost 1.

1. The header row is filled with characters from y and the second row reads as follows: `A1` is the minimum number of steps from "ø" (empty string) to "ø", `B1` the minimum number of steps from "ø" to "i", `C1` the minimum number of steps from "i" to "ir", `D1` the minimum number of steps from "ir" to "ire" etc. At this stage, each operation costs 1 insertion.

	A	B	C	D	E	F	G	H
		i	r	e	l	a	n	d
1	0	1	2	3	4	5	6	7

2. Row `B` is filled in a very similar way as row `A`, except that this time, we do not start from scratch but from "f". `A2` is therefore the minimum number of steps from "f" to an empty string. This first step does not cost 0 as above, but 1 deletion. `B2` is the mininum number of steps from "f" to "i". We have the possibility between the three operations:

[40] Another implementation of the minimum edit distance consists in filling both the first row *and* the first column at the same time. This is the case of following pseudo-code and our script in B.2.

(a) insertion: we add 1 to the number in the left one position (1+1).

(b) deletion: we add 1 to the number in the up one position (1+1).

(c) substitution: a substitution is needed ("f" and "i" are different characters), we add 1 to the diagonally up-left one position (1+0).

As the less costly operation for `B2` is the substitution, `B2` = 1. Apart from this substitution, each of the next stages need 1 insertion. The rest of the row is therefore the same as the previous row.

		A	B	C	D	E	F	G	H
			i	r	e	l	a	n	d
1		0	1	2	3	4	5	6	7
2	f	1	1	2	3	4	5	6	7

3. For row `C`, we start from from "fr". `A3` is the minimum number of steps from "fr" to an empty string, which this time costs 2 (a supplementary deletion relatively to the previous line). `B3` is the minimum number of steps from "fr" to "i". Again, we have the possibility between the three operations:

   (a) insertion: we add 1 to the number in the left one position (1+2).

   (b) deletion: we add 1 to the number in the up one position (1+1).

   (c) substitution: a substitution is needed ("r" and "i" are different characters), we add 1 to the diagonally up-left one position (1+1).

   This time, deletion and substitution have the lowest cost: `B2` = 2. `C2` is the minimum number of steps from "fr" to "ir". As the two strings have the same length and share a common character ("r"), the only operation needed is the substitution from "f" to "i": `C2` = 1. The deletion of "f" necessarily implies the insertion of "i" (2 operations), and the other way round costs the same.

		A	B	C	D	E	F	G	H
			i	r	e	l	a	n	d
1		0	1	2	3	4	5	6	7
2	f	1	1	2	3	4	5	6	7
3	r	2	2	1	2	3	4	5	6

4. The same processing continues, row by row, until the end of the initial string is reached. Once the table is complete, we can determine the minimum edit distance by looking at the very last cell: in this case, the edit distance is indeed 5, as shown in Table 5.2.

When the end of the similarity computation is reached, a table similar to that of Table 5.2 is output. This table was built automatically using an online demo of the Levenshtein distance[41]. The website allows to configure the similarity computation, including the adjustments of the weights assigned to insertion/deletion together, and to substitution apart.

	-	i	r	e	l	a	n	d
-	0	1	2	3	4	5	6	7
f	1	1	2	3	4	5	6	7
r	2	2	1	2	3	4	5	6
a	3	3	2	2	3	3	4	5
n	4	4	3	3	3	4	3	4
c	5	5	4	4	4	4	4	4
e	6	6	5	4	5	5	5	5

Table 5.2: Table of edit distance computation between the strings “france” and “ireland”

Dynamic programming It goes without saying that step-by-step process just described here can easily be fully automated: following an algorithm is easier for a machine than for a human being. The pseudo-code below is the description of the procedure $d_{(x,y)}$ divided into two main steps: the initialisation of the table, and its iterative filling.

- Step 1: initialisation of the first column and the first row of Table T

  Unlike the manual processing of the edit distance, the automated processing requires the header column to be defined before any computation is performed in order to determine the size of the table.

[41] http://odur.let.rug.nl/kleiweg/lev/

```
read input strings x and y
set each element in T to 0
for i from 1 to length(x)+1 do
    T[i,0] := i
end for
for j from 1 to length(y)+1 do
    T[0,j] := j
end for
```

Figure 5.5: Algorithm of the first step of an edit distance program

- Step 2: the computation of the edit distance

```
for j from 1 to length(y)+1 do
    for i from 1 to length(x)+1 do
        if  x[i] = y[j]  then
            substitutionCost := 0                    ▷ no substitution needed
        else
            substitutionCost := 1                    ▷ substitution needed
        end if
        T[i,j] := minimum(T[i-1, j] + 1,                                  ▷ deletion
T[i, j-1] + 1,                    ▷ insertion T[i-1, j-1] + substitutionCost)
    end for
end for
output last cell
```

The core of the edit distance simply resides in those few lines. The imbrication of the two `For` loops simulates the processing that we operated manually above: the first run can be explicitly described as "compare the first character of the first input to each character of the second input incrementally; if the characters are identical, then there is no substitution needed and there its cost is equal to 0; otherwise, its cost is equal to 1. Compare the cost of insertion, deletion and substitution, and keep the smallest value."

In our experiments described in Chapter 6, we do not apply edit distance on strings, nor on parsing trees, but on sequences of POS.

5.6.2 Tree-based: Syntactic Edit Distance

Syntactic or parse trees A parse tree is the tree-shaped representation of the syntactic analysis of a linguistic unit, usually a phrase or a sentence. Incidentally, as any tree, a parse tree has:

- a *root*, which corresponds to the head of the unit;
- at least one *branch*, which corresponds to an intermediate unit;
- and at least one *leaf*, which corresponds to a terminal unit.

Each of these units is a *node* and has a *label* (or *tag*, as in "POS tag"). Different rules are applied to each kind of nodes: the *root node* is the highest in the hierarchy of the tree, which means that it does not have any node above, a *branch node* is a mother node and has at least one child node, while the *leaf node* is a child node without a child node.

In Figure 5.6b (below), `is` is the root node, `sentence` is a branch node, and both `this` and `a` are leaf nodes.

Dependency vs. constituency Any parsed sentence can be derived into a tree. There are two types of parse trees: a parse tree is said to be dependency-based if it complies to a dependency grammar and constituency-based if it complies to a constituency grammar.[42]

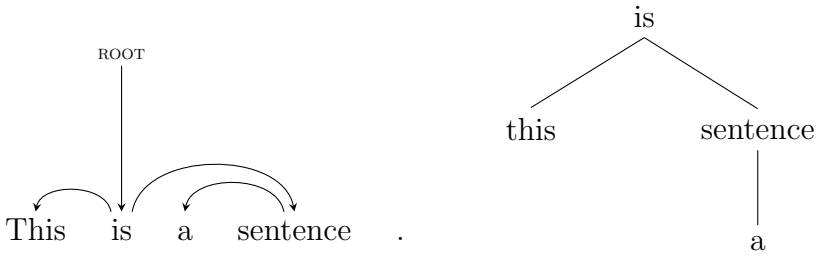
The respective theoretical background and purposes of these two grammars are different, but one can arguably easily transform one type of tree into the other. The conversion from a dependency-based parse tree into a constituency-based parse tree is possible because constituents are retrievable from a dependency-based parse tree: if we take a look at the tree representation in Figure 5.6b, we can see that the constituent ‘‘a sentence" can be extracted if we 'cut' the branch of ' sentence". The other way around is slightly more complex but still also achievable provided that we have the information on the head of each constituent.

One of the main differences between the two trees is that constituency-based parse trees are ordered, which means that the dependants of each node are linearly

[42]More details on the difference between dependency grammar and constituency grammar are given in Section 3.4.4.

ordered[43] [Kahane, 2008]. Indeed, we can see that in constituency-based trees as the ones shown in Figures 3.8 and 4.8, the words of the sentence are written underneath the syntactic tags. The segment is therefore readily readable.

This is not the case for Figure 5.6b where the words "sentence" and "a" are aligned in no particular order. Incidentally, the two dependents of "is" (namely "this" and "a sentence") could be inverted with no consequence on the dependency analysis. It is precisely because of this characteristic that dependency-based trees are often provided with a caption displaying the original linear segment.



(a) Tree linear representation (b) Tree representation

Figure 5.6: Example of a dependency-parsed sentence

Tree edit distance Applying an edit distance algorithm on parse trees is similar to edit distance on strings (the Levenshtein distance), with an obvious difference being that edit operations do not occur on strings, but on nodes of trees. On the "Tree Edit Distance website"[44], the edit distance between ordered labeled trees is merely defined as "minimal-cost sequence of node edit operations that transforms one tree into another" and the operations are reformulated as follows:

- *delete* a node and connect its children to its parent maintaining the order;
- *insert* a node between an existing node and a subsequence of consecutive children of this node;
- *rename* the label of a node.

[43] In the original text, "Un arbre est dit ordonné si les dépendants de chaque nœud sont linéairement ordonnés entre eux, ce qui induit un ordre total sur les feuilles de l'arbre."

[44] http://tree-edit-distance.dbresearch.uni-salzburg.at

Tree edit distance is highly relevant for syntactic similarity measures. Indeed, edit operations occur on nodes directly, which allows to transform a sentence like "This is me" to "This is a sentence" or "This is a sentence used as an illustration in a dissertation thesis" is only one single *rename* operation. The same transformation using string edit distance would take 2 operations in the case of ' This is a sentence" (for example, (1) the substitution of "me" for "a" and (2) the insertion of "sentence" at the end) but at least 10 for the longer sentence.

However, in our preliminary experiments, we only used the morphosyntactically tagged version of the Sejong Corpus and not the parsed version. The main reason is that a syntactic parser relies on POS annotations, which implies that POS annotations are much more common in annotated corpora, including reference corpora, and that morphosyntactic analysers are more reliable: wrong POS tags are likely to lead to a wrong parsing. As a matter of fact, if we want our system to be as generic as possible to be used on as many languages as possible, we need to verify whether relying on POS annotations is sufficient or not. Carrying out experiments on parsed sentences and parsed trees is a step further, all the more so as the cost of such experiments is much higher. First, it is higher not only in terms of resources but also in terms of linguistic implications (see the discussion in the Conclusions of Chapter 4, in 4.5). Secondly, it is also higher in terms of processings, as the processing of trees is less trivial than simply adapting measures, and research developments on this type of processing takes the dedication of a specialised branch of NLP that we did not explore.

5.7 Conclusion

The underlying motivation for the construction of our system is the lack of a corpus exploration tool function that is both accessible to non-specialists of language *and* based on syntax instead of lexical words. This chapter provided a thorough description of the whole processing chain of our *example-based* and *similarity-based* query system. None of these characteristics were invented for that purpose: while the first is already used in a user-friendly corpus exploration system (GrETEL), the second was borrowed from Information Retrieval systems. However, the key to address the issue that we raised is their combination.

Indeed, the originality of our system lies in the combination of the use of natural language examples in input *and* similarity measures, as we can see in Table 5.3. This table compares the different types of corpus exploration tools presented in Chapter 4 to our system, based on four criteria: the type of input provided by the user, the possibilities to insert annotations in the query, the search type and the target linguistic unit for which the tool was constructed.

	Concordancer	Phraseological search engine	Example-based tool	Our system
Input	n-gram	skipgram	example (natural language)	example (natural language)
Annotation in query	explicit or selected	explicit or selected	provided	provided
Search type	exact matching	flexible matching	exact matching	similarity
Target	(sequence of) word	phraseological unit	syntactic unit	syntactic unit

Table 5.3: Comparison table of different corpus exploration tools

It is fundamental to keep in mind that the types of systems presented in Table 5.3 display different characteristics because all of them were created with distinct purposes. As a matter of fact, we can observe that our system shares many characteristics with an example-based tool such as GrETEL, since both allow to search complex structures and are aimed at non-specialist users. The only difference seems to be the search type, which is expected as our system is unique in that matter. On the other hand, this table clearly shows how our system differs from a concordancer such as AntConc or a phraseological search engine such as ConcGram for English and The Lexicoscope for French. More details on the use of those two types tools are given in Chapter 4.

In Chapter 6, we present the experiments that we conducted using our example-based and similarity-based system to search similar syntactic constructions. As the system is still at the specification stage, the experiments focused on the exploration of the wide range of possibilities for the system configuration. Different *options*

were therefore tested in order to define which configuration(s) provide(s) the most relevant results for a given construction. These options pertain to:

1. the input;
2. the similarity measures;
3. the search modes.

Testing these differents options is necessary to answer – even partially – the following questions: which type of input is likely to be used? Which type is likely to get the most out of our system? What is the optimal number of inputs needed to define the query? Which similarity measure, if any, performs best on syntactic similarity? For what kind of syntactic constructions are each search mode relevant? How are these options related? Are there optimal combinations for different types of target construction?

The experiments were mainly conducted on Korean syntactic constructions. Information about Korean language and, in particular, its grammar, are given in Appendix A. For the sake of the genericity of the tool, we also tested our system on English data with the concern to keep the language specificities as small as to fit in a simple configuration file, such as those used in Treetagger for each language it was implemented for. The results of this experiment are displayed in Chapter 7.

Chapter 6

Preliminary Experiments on System Configuration

6.1 Introduction

This last chapter puts in practice the theoretical system described in Chapter 5. Given that the core of this dissertation is the design of a processing chain addressing the problem of the use of corpora by non-specialists to search for syntactic constructions, these experiments are not built on solid ground but they are rather the modest proof of concept of an original functionality that, we believe, is interesting.

This chapter gives a thorough overview of the whole processing chain, from the data processings to the concrete experiments of the similarity computation described in 5.3.4. As mentioned before, the experiments do not include the query refinement through the clustering of the output, which is the very next step of our work.

In Section 6.2, we explain how external resources were involved in our experiments: textbooks from which we extracted sentences that were used to illustrate grammar points and that we selected to similate the queries that learners could use as input, the Sejong Corpus that we sampled to be used as the corpus to which the queries are compared, and a morphosyntactic analyser of Korean. Along

with those resources, we also used a table that we built to compare the grammar points found the first three years of studies of Korean, and based on syntactic and semantic criteria (see A.2.3).

Section 6.3 describes the experiments we conducted and revolve around the four parameters we tested. Although the focus of our study is on the Korean language, the methodology we propose is theoretically extendable to any language. In order to provide evidence to the genericity of our system, we applied our experiments to the English language and described the necessary adaptations in Section 6.4.

6.2 Data Preprocessing

Our experiments involve the comparison between a query (sentence(s) from textbook used to simulate the typical input of a language learner) and a corpus (the POS-tagged version of the Sejong Corpus). Such a comparison implies that both types of data are *comparable*. In other words, both the query and the corpus need to be preprocessed to the right format before we can conduct any experiments. For the Sejong Corpus, preprocessings imply a conversion of encoding and file format, as well as the extraction and sampling of data. For the sentences illustrating grammar points in Korean language textbooks, it inevitably involves an additional step of transcription.

6.2.1 Sampling of Sejong's Tagged Corpus

The version of the Sejong Corpus we use in our study is the version from the official DVD released in 2009[1]. The DVD contains all resources produced during the 10 years of the Sejong Project, including the different versions of the Sejong Corpus. In our preliminary experiments, we only used the morphosyntactically tagged version of the Sejong Corpus for the reasons mentioned in 5.6.2.

In the similarity computation step (see Section 5.3.4 for more details), our system compares the input of the user to each sentence of a corpus. In order to

[1]Kindly provided by Dr. Lee Kihwang to whom I am very much obliged for his kindness and his attention during our only meeting.

do so, the corpus should therefore be segmented into sentences (or utterances in case of spoken data). Precisely, our system takes as input a corpus following the format: one sentence per line, and each sentence has to be morphosyntactically tagged and rigourously composed of blocks of `token/POS` separated by spaces. If we take the example of the sentences we used to illustrate the annotations of the Sejong Corpus in Section 3.5.3, the corpus would be

> 기상청/NNG 은/JX 7/SN 일/NNB 에/JKB 는/JX 전국/NNG 적/XSN 으로/JKB 눈/NNG 이나/JC 비/NNG 가/JKS 내리/VV ㄹ/ETM 것/NNB 이/VCP 라고/EC 말/NNG 하/XSV 았/EP 다/EF ./SF
> 그/MM 애제자/NNG 는/JX 이번/NNG 에/JKB 모/MM 음대/NNG 에/JKB 들어가/VV 았/EP 다/EF ./SF[2]

instead of

> "기상청은 7일에는 전국적으로 눈이나 비가 내릴 것이라고 말했다.
> 그 애제자는 이번에 모 음대에 들어갔다."

It is interesting to note that in this format, we completely lose the notion of *ecel* 어절 (presented in Section 3.5.2) as the smallest unit for morphosyntactic analysis in Korean is the morpheme. This loss has no incidence on our experiments but the original segmentation in *ecel* 어절 should be kept in memory for the output: the segmentation in morphemes with their POS tag is far from being readable for specialists, let alone for non-specialists.

In order to transform the Sejong POS-tagged Corpus into this format, we used the preprocessings presented in Figure 6.1. The steps (represented in green rounded squares) are organised as follows:

1. Samples containing the Sejong POS-tagged Corpus are all converted from UTF-16LE to UTF-8, a more convenient encoding for the manipulation of multilingual data and for the compatibility with our operating system configuration (Ubuntu set in French).

[2]This example is naturally not a real excerpt from the corpus as the two sentences belongs to different samples. We chose to use them for the simple reason that they were analysed in another chapter and are therefore more familiar to the reader.

2. For each sample, we extracted the text from the TEI compliant XML files and stored it in raw text files. At this stage, the text is composed of three columns: the ID of the sentence within the Sejong Project, the *ecel* 어절 and finally the annotated morphemes (see the text with the red background in the figure).

3. For each sample, we extract the third column only as the segmentation in *ecel* 어절 is not useful in our experiments (see the selection using the red dotted line). Notice that the separation between the headline of the article "박쥐장 명인의 서러움" ('Sorrow of bat-pattern closet master craftsman') and the following sentence "문짝 위에 크고 작은 박쥐들을 하나하나 손으로 그려 나갔다" ('He drew one by one, different sizes of bats on doors') is kept by a blank line between them.

4. For each sample, the morphemes of each sentence (or similar unit) are grouped together. Sentences are not separated by blank lines anymore but by line-breaks. The numerotation in the figure (`1` and `2`) was simply artificially added to show that the sentences are indeed separated.

These preprocessings result in 369 POS-tagged raw text samples of modern written Korean, totalising more than 12 million 'words' (*ecel* 어절). Since these experiments are only preliminary and are conducted to test a high number of parameters, we only used a sample of the samples. Indeed, we randomly selected one sample of each of the main genres represented in the written part of the Sejong Corpus:

- **book**, with a sample with texts from several publishers, including *changpi* 창비 one of the major publishers in Korea, historically for "critical writers and intellectuals" but which covers nowadays a vast range of topics, including poetry, the humanities and foreign literature (for adult mostly, but also youth and children), *minumsa* 민음사 specialised in literature and academic publications, *samseng* 삼성 출판사[3] specialised in children books);

[3]Contrary to what the name may suggest, this publisher is not affiliated to the Samsung Group.

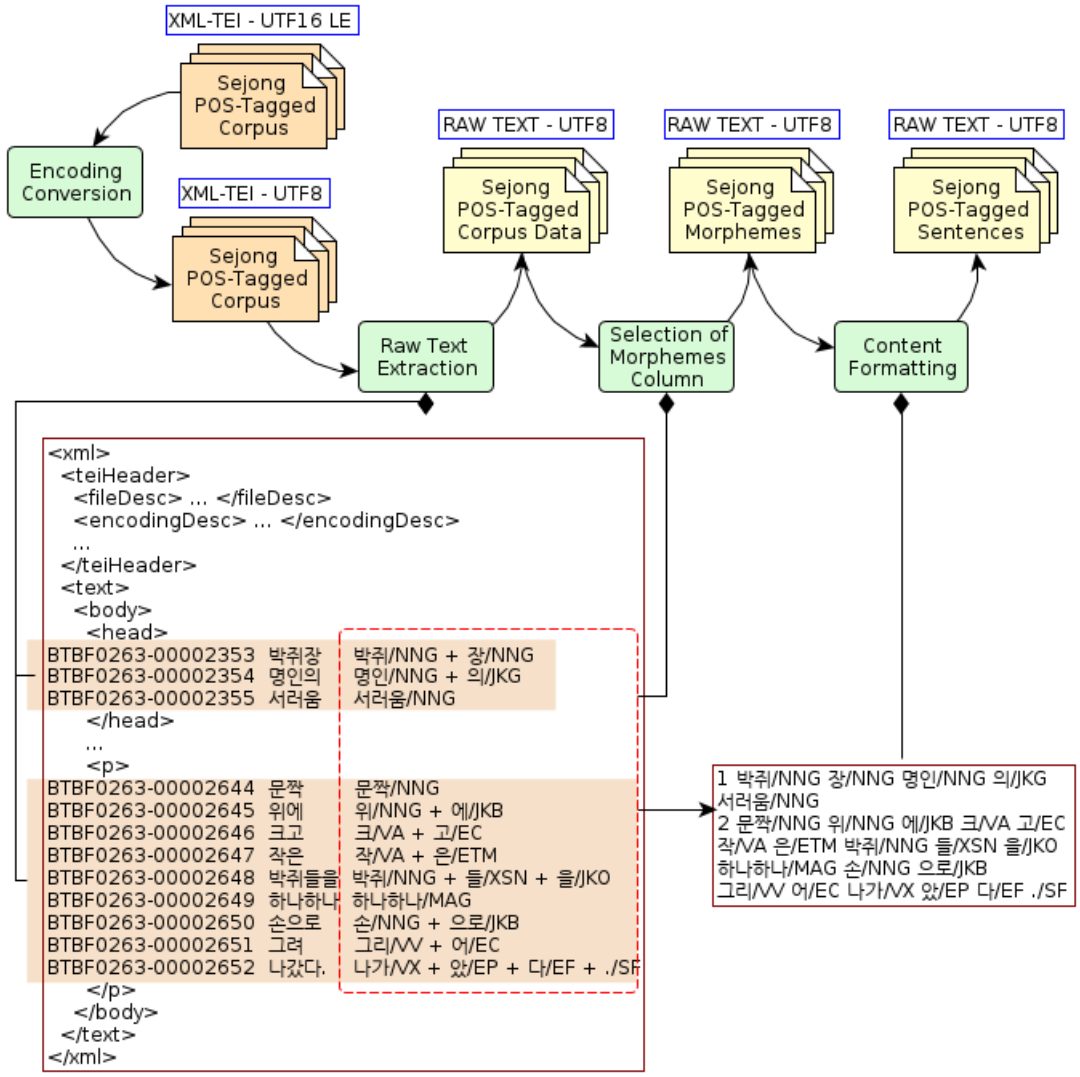


Figure 6.1: Preprocessing of the Sejong Corpus

- **journal**, with texts from different sources as well, including 좋은생각 'Positive' (lit. 'Good Thought'), 월간문학 'Monthly Literature' and 월간 에세이 'Monthly Essay'):

- **newspapers**, in this case, 'the hankyoreh' 한겨레 신문사).

From these samples, we extracted respectively 117,998 sentences (2,274,732 words), 50,392 sentences (1,203,246 words) and 54,022 sentences (1,968,108 words).

6.2.2 Selection of Data from Korean Language Textbooks

This section shows how the target syntactic constructions were selected, from their extraction from textbooks to their actual form used in our experiments.

Extraction from Textbooks The process of extraction of grammar examples to simulate the input of a novice user learning Korean as a foreign language is shown in Figure 6.2. Steps are materialised by green rounded square blocks, the automatic syntactic analysis done by KKMA's tagger is an orange circle, while the textbooks used in our experiments are represented by blue rectangles and the files produced during the preprocessing are in yellow. The flowchart reads as follows:

1. the first step consists in transcribing the sentences used in the introductive dialogue (Figure 6.3) as well as in the various grammatical explanations (Figure 6.4) in a raw text file (one sentence per line);
2. the second step takes the file produced in the first step, and performs an automatic syntactic analysis using KKMA to segment each sentence into morphemes and annotate each morpheme with morphosyntactic tags;
3. in the last step, we randomly select some sentences from the annotated file illustrating grammar points we chose to use for our experiments.

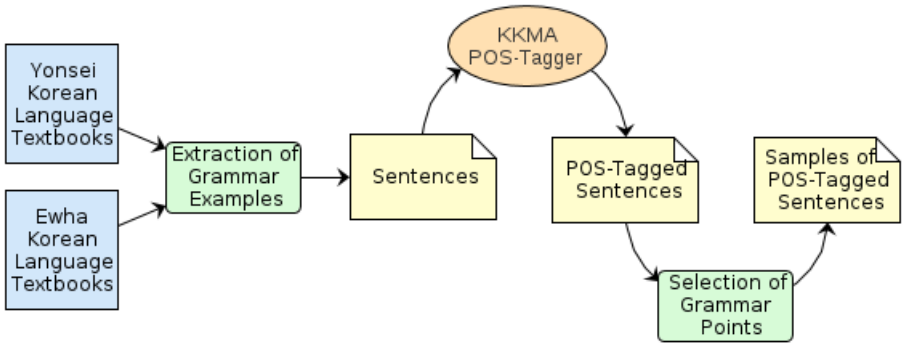


Figure 6.2: Flowchart of the processing of sentences from textbooks examples to input

The two textbooks series we used in our experiments are from Yonsei University and Ewha University. Both were chosen for practical reasons: the Yonsei textbook

series is – or at least, was – used at Inalco where we initially studied Korean, and the Ewha textbook series was used during the author's exchange programme studies at Ewha University. Since the use of these textbooks is limited to the input of sentences illustrating a grammar point (as a simulation of what a language learner could typically do), the limited number of textbooks we have in our possession has no incidence on the quality of our experiments.

The Yonsei Korean (연세 한국어) series currently spans six levels and twelve sublevels from 1-1 to 6-1.[4] One level is thus divided into two sublevels (beginner level is then composed of textbooks 1-1 and 1-2), each corresponding to approximately one semester. The Ewha Korean (이화 한국어) series has a similar division, with the exception of levels 4, 5 and 6 which are not divided into sublevels.[5] The textbooks we used to investigate on grammar points in Korean as a foreign language are the ones that were available to us: Yonsei 1-2, 2-1 and 2-2, as well as Ewha 3-1 and 3-2. This list could be extended to avoid introducing bias but we believe they are decent for our preliminary experiments.

Figures 6.3 and 6.4 are scans of two pages from the textbook level 2-1 of the Yonsei series.

Figure 6.3 shows a typical beginning of a lesson: each lesson corresponds to a *speech act*[6] and is depicted by a constructed short dialogue with an illustration. In this case, the lesson in centred on asking for directions. In the left margin beside the dialogue, we can see the vocabulary from the dialogue that textbook designers judged relevant for this lesson. The title of the lesson is a sentence from the dialogue (with a red box in the figure). Grammar points that are emphasised in this lesson were circled in green.

The dialogue page is followed by an exercise page, which is in turn followed by a grammar explanation page, shown in Figure 6.4. A translation in English of the initial dialogue appears before grammar explanations, also made in English.[7]

[4]Retrieved from http://www.yskli.com/pr/book.asp (in Korean), last checked on 15th June 2017.

[5]Retrieved from http://cms.ewha.ac.kr/user/indexSub.action?codyMenuSeq=9200670&siteId=edukoreankor (in Korean), last checked on 18th August 2017.

[6]Both Yonsei and Ewha textbook series are inscribed in the communicative approach of language teaching.

[7]Translations in English in this textbook are due to the fact that it is intended for international learners whose first language is not Chinese nor Japanese nor Russian. The same choices

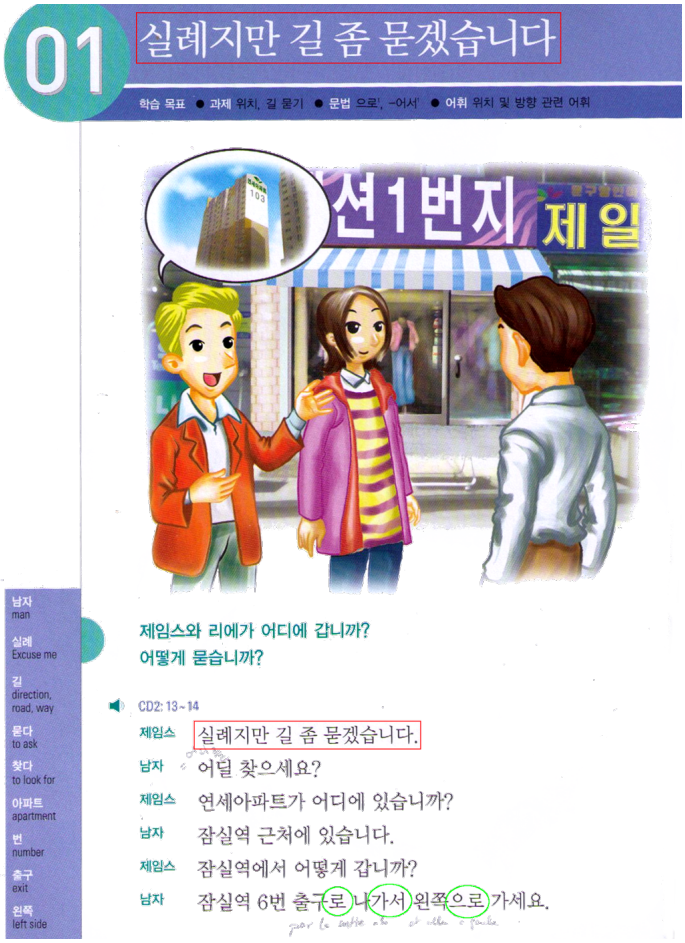


Figure 6.3: Yonsei textbook 1-2: example of dialogue

What catches our attention in this page is inside the red box: the systematic illustration of each grammar point through examples (roughly, between 3 and 10). The grammar point that is described in this lesson was also circled in green.

As explained, the first step in the selection of data from Korean language textbooks consists in gathering all sentences used to illustrate grammar points in a text file. In the case of the lesson on *(u)lo* (으)로 illustrated in those figures, we simply transcribed all sentences containing the grammar point: the last sentence of the dialogue ("잠실역 6번 출구**로** 나가서 왼쪽**으로** 가세요") and the three sentences

concerning versions of the textbook in foreign languages were made for the Ewha textbook series. A unique full Korean version of the textbook is available only starting from level 4 in the Yonsei series and level 3 in the Ewha series.

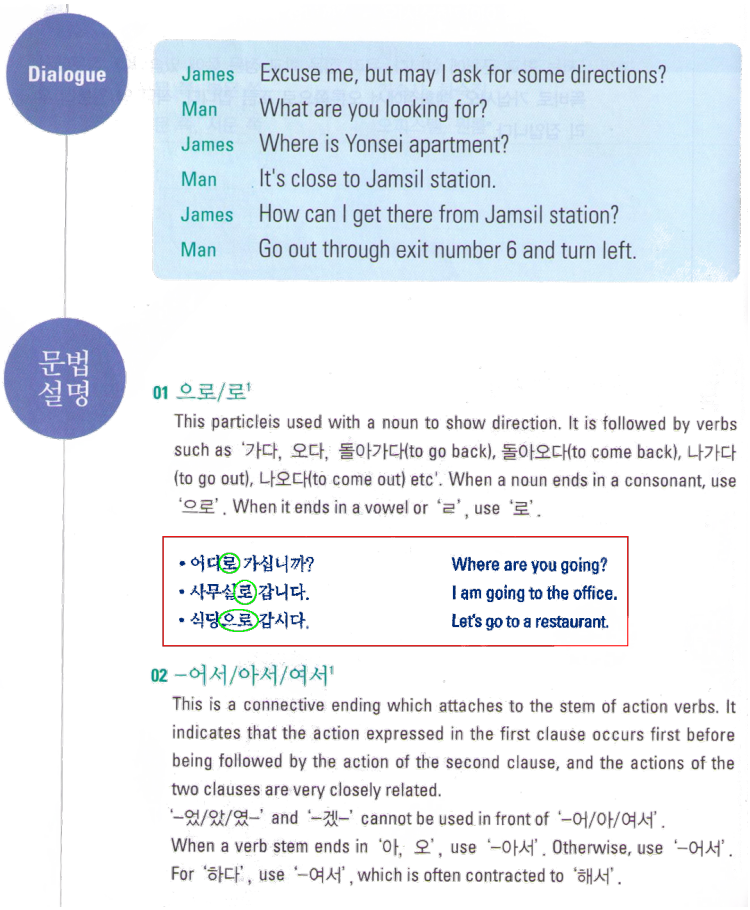


Figure 6.4: Yonsei textbook 1-2: example of grammar lesson

from the grammatical explanation ("어디**로** 가십니까?", "사무실**로** 갑니다." and "식당**으로** 갑시다.").

This step was repeated for all grammar points of the available textbooks (Yonsei 1-2, 2-1, 2-2 and Ewha 3-1, 3-2) and all sentences were transcribed in a single file, in accordance with the input format required for our system, i.e., a raw text file with one sentence per line. In consequence, the only preprocessing that is necessary after this step is the morphosyntactic tagging of sentences in this file, described in Section 6.2.3. However, in order to evaluate our system qualitatively, we tested our system only on a selection of grammar points that we considered particularly relevant for the differents modes of our system. This selection is thoroughly explained in the next pararaphs.

Selection of Grammar Points Our similarity-based system has an original approach to corpus exploration but is not meant to be the solution to all problems. Previous corpus exploration tools or functions provide answers to numerous questions. Incidentally, if, as we have seen at the beginning of this section, Korean grammar relies mostly on grammatical morphemes, would it not be possible to retrieve grammatical constructions by simply using concordancers on these morphemes? This question led us to apply a selection on grammar points for our experiments: which grammatical constructions may be particularly interesting to study using an example-based and similarity-based system?

If we look at Example 26b in the Appendix Section A.2.2, we observe that the specificity of this sentence lies in the use of the morpheme *keyss* 겠. If we search for the morpheme *keyss* 겠 in a corpus of Korean, we will positively retrieve all sentences containing this morpheme without any noise. As a matter of fact, *keyss* 겠 is a non-ambiguous morpheme in that it has no homograph, i.e., no other construction is displayed in this form. However, *keyss* 겠 has slightly **different senses** depending on the context.

Examples in 20 were extracted from Sohn [2013]. Each of them illustrates a different sense of what Sohn names the "presumptive/intentional modal suffix" [Sohn, 2013, p.350].[8]

1. The first sense in Example 20a corresponds to the intentional meaning, also described in *Korean Grammar in Use* within the "intentions and plans" unit. Sohn chose to use the English modal 'will' as a translation but in this sentence, *keyss* 겠 could also be glossed as 'intend to'.

2. A second sense is presented in 20b and denotes a quite high degree of certainty based on the "circumstancial conjecture of the speaker (in declaratives) or the hearer (in interrogatives)" [Sohn, 2013, p.350]. In *Korean Grammar in Use*, this sense *keyss* 겠 appears in a different unit entitled "conjectures".

3. The last example illustrates the presumptive meaning of the suffix *keyss* 겠 in an interrogative sentence.

[8]Other studies consider the suffix *keyss* 겠 as a marker for the future tense [Lukoff, 1982; Martin, 1954]. This hypothesis is discussed and repudiated in Shin [1988, p.76-77].

(20) Different senses of *keyss* 겠

a. 지금보다는 나중에 떠나**겠**어요.
cikum-pota-nun nacwung-ey ttena-keyss-eyo
now-than-TOP later-at leave-**will**-POL
'I will leave some time later rather than now.'

b. 민자는 거기 있**겠**다.
Minca-nun keki iss-**keyss**-ta
Minca-TOP there stay-**may**-DECL
'Minca may be there.'

c. 그 비밀을 누가 알**겠**니.
ku pimil-ul nwu-ka al-**keyss**-ni
the secret-AC who-NM know-**think**-Q
'Who do you think knows the secret?' or 'Do you think someone knows the secret?'

The subtle differences between those senses are naturally disambiguated by the discursive context, i.e., with the knowledge of the situation in which the sentence is produced. We have seen in Section 5.3.5 that a concordancer does not provide a ranking or grouping of the matching sentences but displays *all* matches with no particular order. If learners of Korean are specifically interested in one of these senses, they should therefore go through the concordance result page and try to infer each sense by themselves. However, such suffix with different senses, even subtle, may be an ideal candidate for our tool if the senses could be disambiguated not only by the *discursive context* (which is interpretable only by a human being) but also by the *distributional context* (interpretable by the machine). Indeed, if *hints* can be inferred from the distributional context to help disambiguate the different senses of a given word or morpheme, they can be used as features for similarity computation in order to retrieve only one specific sense, or to construct clusters of sentences sharing these features. A possible hint may be the following rule: "in its intentional meaning, [*keyss*] cannot be preceded by a past tense suffix" [Sohn, 2013, p.346]. In the following section, we investigate the possibility to make use of the distributional context to disambiguate polysemous grammatical morphemes with experiments on *(u)lo* (으)로 (details on the different usages of *(u)lo* (으)로 are found in the Appendix A.2.4).

Practical reasons also plead for the use of a system integrating a morphosyntactic analyser over a simple concordancer for Korean language. Those reasons are due to the use of an alphabet structured in syllables and morphophonological rules.

First, as we have seen for example in the introduction in Section 1.3 or in a note in Section 4.4.4, some constructions display **allomorphy** and retrieving these constructions implies that the user is aware of this allomorphy and is willing to perform two concordances. That is the case of *(u)lo* (으)로 for instance, which is *ulo* 으로 when attached to a stem ending in a consonant, and simply *lo* 로 when attached to a stem ending in a vowel or in the liquid *l* ㄹ.

Considering the rather small size of Korean particles, we can easily imagine how such concordances might end up retrieving undesirable sentences. Indeed, the syllable *lo* 로 is not always the directional/ablative/instrumental morpheme but part of another morpheme or words, as in *tolo* 도로 'road'. The same happens with the concessive morpheme *ato* 아도 as in *cohato* 좋아도 'even if it is good' and not in *lesiato* 러시아도 'Russia too'. This **morphological ambiguity** can be solved by a morphosyntactic analysis since *cohato* 좋아도 is composed of the verb stem *coh* 좋 'to be good' and the morpheme *ato* 아도 and *lesiato* 러시아도 is composed of the proper noun *lesia* 러시아 'Russia' and the delimiter particle *to* 도 'also, even'. In the Sejong Corpus, the former is analysed 좋/VA + 아도/EC and the latter is decomposed into 러시아/NNG + 도/JX.

Some constructions are not fully 'concordanceable' because one (or more) of the morphemes that compose them is '**infrasyllabic**' (lit. 'below the syllable'), i.e., do not constitute a syllable in themselves and involve the restructuring of another syllable (usually, the free consonant becomes the coda of the preceding syllable as Korean has mostly suffixes). This has been discussed in the introduction with the prospective suffix *-(u)l* -(으)ㄹ taking the form *-ul* -을 when attached to a verb stem ending in a consonant but *-l* -ㄹ when attached to a verb stem ending in a vowel. Therefore, while the suffix is directly retrievable using the query 을 as in *mekul* 먹을 'which will eat' or 'to be eaten'[9], it is not in *kal* 갈 'which will go'

[9]The translation of *-(u)l* -(으)ㄹ depends on the function of the word it modifies: *mekul salam* 먹을 사람! 'people who are going to eat!' but *mokul ke ebsta* 먹을 거 없다 'there is nothing to eat' (lit. 'to be eaten').

because a query such as *l* ㄹ only matches an isolated letter.

Another reason that prevents concordancing on a morpheme is what Sohn [2013, p.475] calls *coalescence processes*, processes that involve the concatenation of two morphemes and their restructuring. Those processes include diphthongisation and vowel fusion. We have seen in Section 4.4.4 the verb *hata* 하다 ("to do") which combines unexpectedly with *-ese* -어서 but appears as the contracted form *hayse* 해서 (*ha* 하 + *-ese* -어서). The obligatory contraction between *-si-* -시- (the honorific suffix) + *-eyo* -어요 (the declarative ending) resulting in the form *-seyyo* -세요 is another evidence of vowel fusion.

Those criteria as well as other criteria were used to characterise the grammar points listed from Yonsei and Ewha textbooks. They are summed up in the Appendix in Table A.4. Each criterion is checked (with a star in the corresponding cell) if the grammar point satisfies the respective conditions:

- allomorphy: the grammar point displays contextual allomorphy, i.e., it has different forms depending on the context and this alternation is *distributional* in that one allomorph does not appear in the same context as the other;
- morphological variation: the attachment of the grammar point potentially entails morphological variations of the stem, or the grammar point potentially contains a morpheme subject to morphological variation (for instance, a verb);
- infrasyllabic: the grammar point is composed of at least one 'infrasyllabic' morpheme integrated in the preceding syllable;
- morphological ambiguity: the grammar point has at least one homograph, either a homographic morpheme, or a homograph resulting from a fortuitious combinaison of morphemes which incidentally happens to be similar to the grammar point and therefore causes a morphological ambiguity;
- polysemy: the grammar point has different senses;
- 'concordanceable': it is possible to retrieve the grammar point with a high precision and recall using a single and simple query in a concordancer, i.e., the concordance lines only contain the target grammar point (and not ho-

mographs) in all or most of his forms and usages. By simple query, we imply a query with no regular expressions.

More details on those criteria and the annotation of the table are given in the Appendix A.2.3.

It is still interesting to test concordanceable constructions to see what kind of constructions are similar to them, but we chose to test our system with constructions that illustrate some of those properties:

- the rather complex[10] construction *-l cito moluta* -(으)ㄹ 지도 모르다, used to indicate the speaker's strong uncertainty and composed of the prospective suffix *-(u)l* -(으)ㄹ, the indirect question noun *ci* 지, and the verb *moluta* 모르다 'not know' or 'ignore' according to [Sohn, 2013, p.350];
- the directional/ablative/instrumental morpheme *-(u)lo* -(으)로, whose usages are thoroughly described in Appendix A.2.4;
- the concessive morpheme *-ato/-eto* -아도/-어도.

Those three grammar points are emphasised in bold letters in Table A.4 and extracted to Table 6.1 for the sake of readability.

As we can see in both tables, *-l cito moluta* "-(으)ㄹ 지도 모르다" (line 38) displays allomorphy because of the alternation between *-l* -ㄹ and *-ul* -을, has an infrasyllabic morpheme (*l* ㄹ) and may contain morphological variations, not only because of the infrasyllabic morpheme, but also because of the verb *moluta* 모르다 which has different forms depending on the endings attached to it. For all these reasons, this construction is quite difficult to retrieve exhaustively using a concordancer.

The morpheme *-(u)lo* -(으)로 (lines 56 and 57) displays allomorphy and has different senses but is still 'concordanceable' in form *-ulo* -으로. However, the form *-lo* -로 earns this morpheme a star in the "Morphological ambiguity" column.

As for the morpheme *-ato/-eto* -아도/-어도 (line 121), it has a star in most columns, which means that it displays allomorphy (due to the vowel harmonisation

[10]Especially compared to the numerous monosyllabic morphemes of Korean.

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Sense Ambiguity	Concordanceable	Attached to	Level
38	**-(으)ㄹ 지도 모르다** N+일 지도		*	*				A,V,N	E3-2_10
56	**-(으)로**	*			*	*	*	N	Y2-1_5
57	**-(으)로**	*			*	*	*	N	Y1-2_7
121	**-아/어도**	*	*	*	*			A,V	Y2-1_3 E3-1_4

Table 6.1: Characteristics of a selection of grammar points used in our experiments

with the stem), morphological variations due to the potential infrasyllabic vowels which fusion with the vowel of the stem (*o* 오 'come' + *-ato* -아도 [concession] = *wato* 와도 'although coming'), and morphological ambiguity.

Input in our Experiments For each selected grammar point, we also selected illustration sentences to be annotated (see 6.2.3) and used as queries:

- for *-l cito moluta* "-(으)ㄹ 지도 모르다":
  - before preprocessing: 내일은 맑을지도 모릅니다. 'I have no idea whether or not (the weather) will be clear tomorrow.'
  - after preprocessing: `'내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF'`
- for *-(u)lo* -(으)로:
  - before preprocessing: (1) 젓가락으로 먹습니다. 'I eat with chopsticks.'; (2) 한국말로 말하십시오. 'Please say it in Korean.'; (3) 버스로 왔습니다. 'I came by bus.'; (4) 연필로 씁니다. 'I write with a pencil.'; (5) 김치는 배추로 만듭니다. 'Kimchi is made with cabbages.'

  - after preprocessing: ’전가락/NNG 으로/JKB 먹/VV 습니다/EF ./SF’, ’한국말/NNG 로/JKB 말하/VV 시/EP ㅂ시오/EF ./SF’, ’버스/NNG 로/JKB 오/VV 았/EP 습니다/EF ./SF’, ’연필/NNG 로/JKB 쓰/VV ㅂ니다/EF ./SF’, ’김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF’

- for *-ato/-eto* -아도/-어도:

  - before preprocessing: 문제가 어려워도 끝까지 풀 거예요. ‘Even though the problem is complex, I will solve it entirely.’
  - after preprocessing: ’문제/NNG 가/JKS 어렵/VA 어도/ECD 끝/NNG 까지/JX 푸/VV ㄹ/ETD 거/NNB 이/VCP 에요/EF ./SF’

For the morphosyntactic tagging of those input, please refer to Section 6.2.3 and to Table A.2.2 for a description of each tag and at least one illustration for the Sejong Corpus.

6.2.3 Morphosyntactic Tagging

We have seen in Chapter 5 that in order to compute the syntactic similarity between user input(s) and the corpus, at least two conditions have to be satisfied:

- firstly, there must be a minimum of syntactic information, which means that both the input and the corpus should be annotated in *(morpho)syntax*;
- secondly, the tagset used for the annotation of both the input and the corpus must be *identical*, as for a computer program, different annotations mean different objects.

This criterion led us to choose the morphosyntactically tagged version of the Sejong Corpus, as shown previously in Section 6.2.1. Since this corpus is the Korean National Corpus, the morphosyntactic annotations are *gold standard*, which means that they should have very few errors thanks to manual correction.[11]

[11] Errors in Corpus Linguistics (and especially in gold standard corpora) are discussed in Section 4.4.3.

On the other hand, the input is entered dynamically by users who may not be experts in language, linguistics or NLP. In our example-based system, the input is made of one (or multiple) *raw sentence(s)*, without any annotation. As a consequence, the beginning of our processing chain integrates an automatic morphosyntactic analyser.

Selection of a morphosyntactic analyser Since the core of our system is not focused on morphosyntactic annotation, we integrated an existing morphosyntactic tagger instead of trying to reinvent the wheel. This tagger was loaded from the Python package KoNLPy (Korean NLP in Python), an open source software [Park and Cho, 2014] in which five of the major recent open source morphological analysers for Korean were wrapped:

- HanNanum[12], from KAIST;
- KKMA – pronounced [k͡ok͡oma] for *kkokkoma* 꼬꼬마[13], from Seoul National University;
- KOMORAN[14], developed by Shineware;
- twitter-korean-text[15], built to be used on tweets specifically;
- and MeCab-ko[16], the Korean version of MeCab, mostly known for its original version for Japanese.

All of them provide both the *segmentation into morphemes* (required as Korean is an agglutinative language) and their *labelisation* (or annotation). However, they do differ, not only in terms of *time analysis* (which encompasses both the resources loading time and the execution time), but also in *performance* with regard to the size and degree of sophistication of their tagset, as well as with regard to the coverage of the variations in the Korean language.

[12] http://semanticweb.kaist.ac.kr/home/index.php/HanNanum
[13] **K**ind **K**orean **M**orpheme **A**nalyzer, http://kkma.snu.ac.kr/
[14] **Ko**rean **Mor**phological **An**alyzer, http://shineware.tistory.com/tag/KOMORAN
[15] https://github.com/twitter/twitter-korean-text/
[16] https://bitbucket.org/unjeon/mecab-ko/

We may observe for example that unlike other morphosyntactic taggers, twitter-korean-text is most efficient on short texts since it was built to annotate tweets.[17] Although its application is not limited to tweets, the original purpose of twitter-korean-text also imply that this tagger should be able to handle non-normalised data and a wider range of variations, including internet slang and proper nouns (especially *named entities* such as organisations or product names).[18]

A comparison of the five morphosyntactic taggers available on KoNLPy's website[19] allows us to comprehend those differences. KoNLPy's development team conducted tests on the performance of the taggers regarding different specific issues:

1. segmentation into *ecel* 어절[20] and thus, the spacing following the official rules;

2. analysis of ambiguous words;

3. treatment of unknown words (e.g. slang, or words that are not included in the dictionary used by the taggers).

The first issue was tested using a **sentence with no spacing**. This peculiar display is actually likely to happen in Korean: despite an official orthography reform in 1989, spelling and spacing rules are considered "too rigid and too relaxed for consistent application" [Kim, 2013], and remain a "continuing challenge" to both Korean natives and non-natives who still write with very few spacing compared to what is recommended. The tests show that MeCab-ko's segmentation of such input is the most accurate, which automatically leads to a better morphosyntactic analysis. KKMA and twitter-korean-text follow with one error, while HanNaNum

[17] Tweets are messages limited to 140 characters. The definition of a character for Twitter is available on https://dev.twitter.com/basics/counting-characters.

[18] Indeed, it is shown on their official website that twitter-korean-text does include a normalisation module, converting for example 입니닼ㅋㅋ to 입니다 ㅋㅋ, where the symbol ㅋ used to simulate laughter (similar to "haha" in English and French, or "jaja" in Spanish) was inserted in the previous word *ipnita* 입니다 (the verb 'to be' in formal speech), but also 샤릉해 to 사랑해, where *syalunghay* 샤릉해 is the somewhat cuter pronunciation of *salanghay* 사랑해 ('to love' in informal speech).

[19] http://konlpy.org/en/v0.4.4/morph/#comparison-between-pos-tagging-classes

[20] See the description of this peculiar unit in Section 3.5.2.

and KOMORAN do not engage in segmenting the sentence and end up tagging the sentence as a noun.

The second test uses an **ambiguous word**, *nanun* 나는 which could be either the pronoun *na* 나 'me' to which the topic marking particle *nun* 는 is attached (meaning 'as for me') or the verb *nalta* 날다 'to fly' with the adnominal ending *nun* 는. In this case, KKMA is the only morphosyntactic tagger which manages to correctly identify the verb in a given example.

Finally, the third test consists in giving a sentence containing several words that are **not likely to be included in the taggers' dictionaries**, such as *ayphulkonghom* 애플공홈 'Apple's official website', *aiphon* 아이폰 'iPhone' and *enlakphon* 언락폰 'unlock phone'[21]. The first two words are correctly segmented in most cases, except for KKMA which separated *ai* 아이 from *phon* 폰, mistaking *ai* 아이 (transliteration of 'I') with *ai* 아이 ('child'). However, the third word was more challenging since it was given concatenated with the following verb, in the form *enlakphoncillepelyessta* 언락폰질러버렸다. Consistent with its behaviour in the first test, HanNaNum analyses this form as a single noun but the four other taggers at least separated the noun (언락폰) from the verb (질러버렸다). We note that twitter-korean-text is the only tagger which analyses *enlakphon* 언락폰 as a single block, while KKMA and MeCab-ko analysed *enlak* 언락 (which is the transliteration of the English word 'unlock') and *phon* 폰 (for 'phone') separately. Both of these segmentations are possible, but KOMORAN made an error in analysing each syllable as a separate noun (언/NNG + 락/NNG + 폰/NNG).

Considering the application of our system and its original use, words that are given as input by learners of Korean are more likely to be regular nouns than unknown nouns. As a matter of fact, the analysis of the verb here is more interesting for our study. For the segmentation of the verbal form *cillepelyessta* 질러버렸다 into morphemes, KOMORAN shows the best results: the tagger correctly recognised the verb *cilu* 지르 and the verbal endings attached to the verbal stem *e* 어 *peli* 버리 *ess* 었 and *ta* 다. KKMA is close but gives the wrong verb stem (namely, *cillu* 질르, which does not exist) while MeCab-ko and twitter-korean-text do not

[21] An "unlock phone" refers to a contract-free cellphone that can be used on any carrier or network. Recent iPhones are most likely to be unlocked, but since the debut of iPhones, there were SIM restrictions to a particular carrier.

separate each morpheme.

Still according to tests conducted by KoNLPy's development team, KKMA is the slowest morphological analyser among the five. However, **time consumption** is not considered as a major drawback in so far as in our case, the tagger is meant to be used solely on input sentences (a few, at most) as a preprocessing operation.

Although the conclusions of the available description of the tests seem to be only drawn from one example for each issue, we did not consider it necessary to conduct further tests, as our decision concerning the choice of a morphosyntactic tagger ultimately relies on another criterion. If we want to match sentences from the Sejong Corpus (the corpus we are using for our experiments) with the automatically annotated input of the user, the morphosyntactic tagset must be absolutely identical. This criterion eliminates HanNaNum and twitter-korean-text whose tagsets are considerably smaller. Among the remaining taggers, KKMA has a similar tagset to that of the Sejong Corpus, and KOMORAN and Mecab-ko have the exact same tagset. Considering the performance tests (especially on ambiguous words and the detailed segmentation of verbs) and the fact that KKMA was used for the Sejong Corpus [Lee et al., 2010], we chose to integrate KKMA in our processing chain. Only minor adaptations were needed for the POS tags to match perfectly as it seems that the tagset of KKMA is a little more precise than the one found in our CD version of Sejong Corpus.

Tagset Adaptation The adaptations that are necessary for our experiments are shown in Table 6.2. The first column contains the general description of the tags in the following columns; the second column contains the tags from KKMA that do not exist as such in the Sejong Corpus; and the third column contains the corresponding tags from the Sejong Corpus. Using this table, the adaptation is simply executed by a simple function we implemented: replace each tag from the second column by the tag of the same row in the third column. Since the KKMA tagset is a little more precise than the one used in the Sejong Corpus, the adaptation of the tagset used by KKMA to the one used in the Sejong Corpus necessarily implies a simplification of tags. For instance, words originally tagged `MDT` (i.e., 관형사, 'common determiner') or `MDN` (i.e., 수관형사, 'numeral determiner') by KKMA

will both be converted into the simplified tag `MM` (i.e., determiner) before they are compared to the sentences of the Sejong Corpus. In other words, the distinction between `MDT` and `MDN` is lost.

As a matter of fact, KKMA's original POS tagset contained 60 tags but the tags `XSM` for 'adverb derivation suffix' (부사 파생 접미사), `XSO` 'other suffix' (기타 접미사), `UV` 'unknown predicate' (용언추정범주) and `UE` 'non-analysable word' (분석불능범주) were deprecated.[22] The version of KKMA provided by KoNLPy has a large set of 56 tags (although the independent version of KKMA seems to enable an option limiting the number of tags to 10 or 30), versus 42 tags for the Sejong Corpus (or 48 if we include tags used for the spoken corpus). The similarities and differences between the tagsets of the five taggers and reference corpora are summarised in a comparison chart provided by KoNLPy's documentation.[23] Table 6.2 was inspired by this chart, with the difference that tags are listed in alphabetical order for the sake of readability[24] and tags that were simply converted and those that imply a simplification are presented separately in different sections.

Figure 6.5 below shows different steps of an automatic syntactic analysis in Korean performed using KKMA. The sentence we used is extracted from Yonsei's textbook 1-2 to illustrate the use of the causal suffix *-u(nikka)* -(으)니까. This sentence is therefore a typical example of input we designed our tool for.

The first line in the figure simply shows the raw sentence given as input: *nalssika chwuwunikka anulo tulekaseyyo* 날씨가 추우니까 안으로 들어가세요 (literally, 'as the weather is cold, please go inside'). In the second line, the sentence has been segmented into four words based on the provided spacing. Finally, the third and last line shows the output of the segmentation into morphemes, the morphosyntactic annotation of the sentence using KKMA and the linguistic gloss.

[22] The original tagset is shown on http://kkma.snu.ac.kr/documents/index.jsp?doc=postag, where the deprecated tags are struck-out.

[23] https://docs.google.com/spreadsheets/d/1OGAjUvalBuX-oZvZ_-9tEfYD2gQe7hTGsgUpiiBSXI8/edit#gid=0, last retrieved on 3rd March 2016.

[24] Since we selected POS tags that differ from the Sejong tagset, we did not group POS tags by word classes (as KoNLPy did) and thus needed another ranking to present them.

Part-of-Speech	Tag from KKMA	Corresponding Tags in Sejong
Conversion		
Adnominal ending 관형형 전성 어미	ETD	ETM
Vocative particle 호격 조사	JKI	JKV
Adverbial particle 부사격 조사	JKM	JKB
Conjunctive adverb 접속 부사	MAC	MAJ
Sinogram 한자	OH	SH
Foreign loanword 외국어	OL	SL
Number 숫자	ON	SN
Non-analysable word 분석불능범주	UE	NA
Unknown noun 명사추정범주	UN	NF
Unknown predicate 용언추정범주	UV (dep.)	NV
Simplification		
Conjunctive ending 연결 어미	ECE, ECD, ECS	EC
Final verbal ending 종결 어미	EFN, EFQ, EFO, EFA, EFI, EFR	EF
Prefinal verbal ending 선어말 어미	EPH, EP, EPP	EP
Determiner 관형사	MDT, MDN	MM
Bound noun 의존 명사	NNM	NNB
Auxiliary verb 보조 용언	VXA, VXV	VX

Table 6.2: Comparison table between tags from KKMA and their corresponding tags in the Sejong Corpus

Annotation Errors Despite the strong similarity between the tagset and the simple conversion function we implemented, there is still one difference crucial

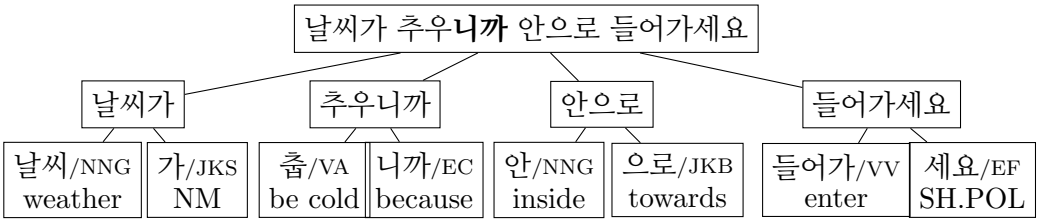


Figure 6.5: Morphosyntactic analysis of a sentence illustrating *-u(nikka)* -(으)니까

enough to point out between the annotation of the corpus and the one in the input. The Sejong Corpus has gold standard annotations which have been manually corrected whereas the input is automatically annotated and therefore certainly contains errors. It is not so much the errors in themselves as the discrepancy between the corrected and the non-corrected annotations that may lead to complications: as mentioned at the end of Section 4.4.3 and stressed in Section 5.3.2, the matching of similar constructions is the key feature of our system.

The sentence in Figure 6.5 illustrates this issue, as its automatic morphosyntactic analysis performed by KKMA contains one error of segmentation. The verbal form *tulekaseyyo* 들어가세요 should not have been segmented into two morphemes but three: *tuleka* 들어가/VV (the verb stem) + *si* 시/EP (the honorific suffix) + *eyo* 어요/EC (the declarative ending). The segmentation algorithm here failed to recognise the two different morphemes in the contracted form *-seyyo* -세요. Since this form does not exist in the segmented and annotated version of the Sejong Corpus, this sentence will not match any sentence from the Sejong Corpus. However, the fact that our system is based on similarity rather than on strict matching certainly allows to retrieve sentences that contain *-u(nikka)* -(으)니까. Yet, the wrong segmentation does prevent the program from relying on the honorific suffix *si* 시 and use it as a feature in the similarity computation, which is a pity because contrary to another widely used causal morpheme, *ase/ese* 어서/아서, *-u(nikka)* -(으)니까 can be used in imperative or propositive sentences such as the one in Figure 6.5.

6.3 Preliminary Experiments: Objectives and Results

The objective of these preliminary experiments is to test different parameters for the system configuration in order to understand which work best, or which are the most relevant for a given task. This section presents the objectives, the implementation, as well as the results of the following parameters:

- number of input sentences (one, two, three or more?);
- type of input (using both wordforms and POS, or only the POS of lexical words such as nouns and verbs?);
- similarity measure (the Jaccard/Sørensen-Dice distances, minimum edit distance?) and data representation (unigrams or bigrams?);
- genre (written – book, journal or newspaper).

These experiments are at first limited and adapted to the Korean language. Parameters were therefore all tested to retrieve Korean syntactic constructions with various properties. Among them, we selected three constructions on which we will particularly focus, based on the criteria listed in "Selection of Grammar Points" (Section 6.2.2):

- *-(u)lcito moluta* -(으)ㄹ지도 모르다, a complex construction used to indicate the speaker's strong uncertainty;
- *-(u)lo* -(으)로, the directional/ablative/instrumental morpheme;
- *-ato/eto* -아도/어도, the concessive morpheme.

In order to keep this section as comprehensible as possible, we chose to give the results of the experiments on one or two of the grammar points for each parameter, as well as to comment on some retrieved sentences, but not all of them. In order to see the original output files, please refer to Appendix C.

Moreover, unless otherwise specified, we comment the results obtained **without lexical words** in the query and the corpus, and with either the **Jaccard distance using bigram** (and *not* unigrams, except for the experiments focusing specifically on the representation of data) or the **Levenshtein distance**.

A Word on Modes Each parameter has been tested in two of the three proposed search modes presented in Chapter 5: the *default mode*, based on both token and context similarities, and the *distributional analysis* search mode, solely based on the context similarity. As a matter of fact, the third mode has also been tested. However, the results are not satisfying enough to be presented and further experiments need to be conducted.

The third mode, called *different usages* search mode, is designed as the opposite of the distributional analysis mode. While the latter retrieves sentences that do *not* contain the target word(s) but are constructed in a way similar to the query (similar sequence of words and/or POS tags), the former retrieves sentences that *do* contain the target word(s), but which appear(s) in a context that is as dissimilar as possible to the query. In order to do so, we first compare the query to the sentences of the corpus, select the ones that contain the target word or sequence of words (not necessarily contiguous, but in the same order) and then rank the context by decreasing order of similarity. However, our experiments using this simple method revealed two major flaws.

First, no matter the type of similarity measure, the most dissimilar sentences seem to be the longest. For the Jaccard or the Dice distances, long sentences increase the number of different words between the two sets.[25] For edit distance, this mode is particularly not suitable as long sentences inevitably increase the number of operations as soon as their length exceeds that of the query. Retrieving the least similar sentences simply amounts to retrieving the longest sentences of the corpus provided they contain the target word(s). In both cases, the longer the sentence, the bigger the distance between the query and the long sentence.

Second, observing two sentences in a foreign language and understanding why they are similar is not an easy task. The same exercise but this time trying to infer why they are *dissimilar* is somewhat even more complex. Added to the fact that in our experiments on this mode, the Jaccard and the Dice distances' scores reached 1.0 (most dissimilar)[26] for several sentences (thus bypassing any kind of

[25] See the formulas of the two coefficients given 6.3.3.

[26] We did not used the coefficient of both Jaccard and Dice but their corresponding distance: 0.0 is the lowest distance and therefore the strongest similarity, while 1.0 is the highest distance

ranking), the analysis of dissimilarities becomes close to impossible.

Modes Implementation For the two remaining search modes, the method is the same: first, a selection is applied on sentences from the corpus.

In the default mode, we only keep the sentences that contain the target word(s). If several morphemes (or words) are targeted, such as for *-(u)lcito moluta* (으)ㄹ 지도 모르다, the morphemes need not be contiguous, given that we want to allow words like modifiers between certain morphemes. However, they have to appear in the same order: a sentence that contains the verb *molu* 모르 'ignore' *before -lcito* (으)ㄹ 지도 would not be retrieved. In other words, this primary selection amounts to an internal concordancing.

Conversely, for the distributional analysis mode, the key of the selection is exactly the opposite: we only keep the sentences that do *not* contain *all* of the target word(s) in the same order but their POS. This means that at least one the morphemes need to be replaced by another form tagged with the same POS, among *-(u)l* -(으)ㄹ, *-ci* -지, *-to* -도 and *molu* 모르.

Then, similarity measures between the query and each selected sentence of the corpus are computed.

Evaluation? In the following sections, the results were only evaluated qualitatively by the author. A quantitative evaluation was not performed for different reasons.

First, the one of the two classic evaluation methods in information retrieval, the *recall* (the number of relevant sentences retrieved, among all of the relevant sentences in the corpus), cannot be computed, because we do not know the total number of relevant sentences.

Second, our system is a kind of hybrid between a concordancer and an information retrieval system, given that it uses similarity measures and ranking, but also integrates an internal concordancing. Consequently, all of the retrieved sentences are presumably relevant for the user's need: any sentence that contains the target construction is somehow similar to the query, and is therefore likely to provide an answer to the user. The problem lies in the definitions of 'similarity' and that

and therefore the strongest dissimilarity.

of the user need. As we stated in Section 5.2.1, vagueness in the input inevitably calls for vagueness in the output. The computing of the other classic evaluation method, the *precision* (the number of relevant sentences retrieved, among all of the retrieved sentences), is possible but this property makes it delicate. Obvious mistakes can be pointed out (as we do in the analysis of the results), but the relevancy of the retrieved sentences depends on the user's need and judgement.

A proper evaluation would be to ask users to provide feedback on the system, which is a perspective we consider.

6.3.1 Number of Inputs

We initially tested the parameter of the number of inputs without the primary internal concordancing, with the selection of sentences that do or do not contain the target word(s). Sentences that contained the right construction with the same meaning was considered relevant, while sentences that did not were irrelevant. From the results of these tests, we selected the query that has the most relevant results to run all of the single query searches.[27]

However, in this section, we focus solely on experiments including the internal concordancing.

6.3.1.1 Objective(s)

Our system offers the possibility to combine multiple sentences as input. This possibility was initially thought of due to the fact that we use examples as queries, and that the input can be as vague as simply as a single sequence of POS and a single word targeted.[28] A greater number of sentences could be relevant to **define the query more precisely** from the beginning of the search. The objective of this parameter is to check if indeed the use of several sentences may be helpful for the user, or if it only produces more confusion (or noise) when the sequences of words or of POS are not similar enough for the system.

[27]Precisely, the score of each sentence was the mean average precision, an evaluation method from information retrieval that allows both the relevancy and the rank of each document (in our case, sentence) to be taken into consideration.

[28]Multiple input is one of the options mentioned in Section 5.3.

6.3.1.2 Implementation

We arranged our similarity measure script to take as input a list of sentences: if the list contains only one sentence, the search will be based on a single query, if the list contains more than one sentence, the search will be based on multiple queries. In the latter case, the final score is the harmonic means of the individual scores (see the function `get_hmeans` in the script in Section B.1).

We decided to comment the results of the tests conducted on the polysemous morpheme *-(u)lo* -(으)로. As mentioned, the sentence we use as queries in our experiments were all extracted from textbooks either from the Ewha or from the Yonsei series. Likewise, the sentences we used as queries for a 'multiple input search' were all extracted from a textbook, precisely from the same unit.

For example, the sentence we use for the 'single input search' for *-(u)lo* -(으)로 is *kimchinun paechwulo mantupnita* 김치는 배추로 만듭니다. 'Kimchi is made with cabbages.', chosen to illustrate the use of the particle *-(u)lo* -(으)로 to indicate **means**. The four other sentences used as queries were extracted from the same grammar lesson. They are shown and translated in the "Input in our Experiments" paragraph, at the end of 6.2.1.

6.3.1.3 Results in C.1

Using multiple sentences as input may not be helpful if the different sentences are too different. In the case of *-(u)lo* -(으)로, the sentences are quite similar. If we remove lexical words, the sequences of POS of the five sentences are:

- `'NNG 으로/JKB VV 습니다/EF ./SF'`,
- `'NNG 로/JKB VV 시/EP ㅂ시오/EF ./SF'`,
- `'NNG 로/JKB VV 았/EP 습니다/EF ./SF'`,
- `'NNG 로/JKB VV ㅂ니다/EF ./SF'`,
- `'NNG 는/JX NNG 로/JKB VV ㅂ니다/EF ./SF'`

Despite some differences, the immediate context of the target morpheme is the same: in these examples, *-(u)lo* -(으)로 is always preceded by a common noun (`NNG`) and followed by a verb (`VV`).

Default Mode In the default mode, differences between results using a single query and results using multiple queries are very thin concerning *-(u)lo* -(으)로. Indeed, the rankings of the two modes share some sentences: 5/10, for the results obtained with the Jaccard distance and 3/10 for those obtained with the Levenshtein distance. The sentences that are not common to the two rankings are, in fact, rather similar as they display almost exclusively the directional function of *-(u)lo* -(으)로.

The results are a little more conclusive for the concessive morpheme *-ato/eto* -아도/어도 where the use of multiple input concretely helped targeting the context of the queries, since we find less sentences with the specific use of *-ato/eto* -아도/어도 in the construction *-ato/eto toeta* -아도/어도 되다 used for asking/giving the permission. Using a single input search, the results contain 1 2/10 sentences with this construction (sentences 2 and 3 in Jaccard using bigrams with lexical words, and 6 and 7 without lexical words; sentence 5 in Levenshtein with lexical words, and sentences 2 and 5 without lexical words). In the multiple input search, only one sentence contains this construction, and is only ranked 9 in Levenshtein without lexical words.

Distributional Analysis Mode In the distributional analysis mode, the situation is the same as in the default mode. 6/10 of the sentences retrieved with the Jaccard distance and 3/10 of the sentences retrieved with the Levenshtein distance, both using a single query, also appear in the results of the multiple query search.

We can therefore assert that, in most cases, one sentence given in input was sufficient to determine the context targeted. The only notable improvement observed from one to multiple queries is the experiments using wordforms. Surprisingly enough, it looks like one query was not enough to target the context correctly and to retrieve relevant similar sentences among the 48,722 matches (sentences that do not contain *-(u)lo* -(으)로, but another subject particle (`JKS`)). Conversely, using multiple queries seemed to allow the system to target the context given that the sentences retrieved are much shorter than the ones with a single query.

6.3.2 Type of Input

6.3.2.1 Objective(s)

Since our system is built to retrieve syntactic constructions, we could consider that lexical words should not appear in the query, or should, at least, have a minor weight compared to grammatical morphemes. In Section 3.4, we gave the example of the progressive form `V-ing` in English, where the suffix `-ing` is not specific to a particular verb, but possibly attached to any verb. Retrieving progressive forms in a corpus could therefore be achieved by simply searching for verbs as a category, or POS, followed by the suffix `-ing`.

However, deleting all lexical units could also prevent our tool from retrieving certain structures relying on a lexical word. This is typically the case of *-(u)lcito moluta* -(으)ㄹ지도 모르다, which uses the verb *moluta* 모르다 'to ignore'.

It is noteworthy that this option is not more or less compatible with the default mode or the distributional mode. The modes mainly focus on the construction identified as a target, while the type of input option's scope is not only as wide as the input but also include the sentences of the corpus. For instance, in the distributional mode, searching for *-(u)lo* -(으)로 means to search for particles (tagged `JKB`) which are *not -(u)lo* -(으)로. The rest of the input sentence may or may not have lexical words depending on the type of input option, not on the mode. Conversely, in the default mode, the wordform *-(u)lo* -(으)로 is used in the query along with its POS, regardless of the type of input option, since particles are not considered as lexical words.

The objective of our experiments on **removing lexical items** is to determine which type of input is the most relevant for our system: should we keep all wordforms, or does the removal of lexical items give more interesting results? Is there a type of input that is more adequate than others? Of course, as described in Chapter 5, the user can directly choose the word(s) he or she want to keep as a tagged word, or as POS only, or not at all. Yet, the results of our experiments would be useful for the default settings of the system.

6.3.2.2 Implementation

In order to answer those questions, in one of the series of tests we conducted, lexical units were systematically removed using the function `remove_lexical_item` we defined in our script (see B.1). For Korean, we removed the wordforms of tokens tagged: `NNG` (common nouns), `NNP` (proper nouns), `NR` (numbers), `NP` (pronouns), `VV` (verbs), `VA` (adjectives) and `MM` (determiners). These POS were chosen on the basis that they are mostly not part of syntactic constructions as specific words, but rather as a whole paradigm. This list therefore includes genuine lexical words (nouns, verbs, adjectives), as well as function words (numbers, pronouns, determiners). For the same reason, we did not include adverbs in those experiments on Korean since we believe that they might have some link with the verbal endings (if we think of sentences containing the adverbs *celtay* 절대 'never, absolutely', *yeksi* 역시 'by any chance', for example). The impact of adverbs could be measured in further experiments.

Removing lexical items was performed both on the query as a whole and on the sentences of the corpus. For example, for *-(u)lo* -(으)로, the sentence *kimchinun paechwulo mantupnita* 김치는 배추로 만듭니다. 'Kimchi is made with cabbages.' is sentence that was extracted from a textbook to be used as the query, while *molaylo mantun pyek* 모래로 만든 벽. 'A wall made of sand.' is a sentence from the Sejong Corpus. After the removal of lexical items, these two sentences appear, on the one hand as `NNG 는/JX NNG 로/JKB VV ㅂ니다/EF ./SF`, and, on the other hand, as `NNG 로/JKB VV ㄴ/ETM NNG ./SF`.

6.3.2.3 Results in C.2

Default Mode In the output file C.2.1 resulting from the experiment on *-(u)lo* -(으)로, we note that using wordforms and bigrams allows Jaccard to retrieve exclusively[29] sentences containing the target morpheme *-(u)lo* -(으)로 with the same verb as in the query, *mantul* 만들 'create, fabricate'. In other words, this combination of parameters retrieves one of the multiple usages of *-(u)lo* -(으)로, the material function.[30]

[29] At least in the top 10 most similar sentences.
[30] The different usages of *-(u)lo* -(으)로 are described in Appendix Section A.2.4.

On the other hand, removing of lexical words allows to retrieve other usages of *-(u)lo* -(으)로, such as the directional function (sentences 1 and 8, with the motion verb *ollaka* 올라가 'ascend'), the denotation of a change of state (sentence 9, with the verb *pyenha* 변하 'change'). This result seems less accurate than that using wordforms, but considering the query, one cannot determine if the user wants to retrieve sentences with *-(u)lo* -(으)로 denotating a material, or rather different usages of the morpheme. Moreover, considering the clustering phase that is supposed to group similar retrieved sentences together, this removing lexical words could be more interesting as the clusters might be the different usages, if the clustering algorithm uses the verbs (see Section 5.3.5). In this case, sentences with the directional function would be grouped together under a single representative, while sentences with the material denotation function would belong in another cluster.

Using the edit distance (the Levenshtein distance) without lexical words gives similar results to the Jaccard distance (sentences ranked 1, 2 and 3 using Jaccard's distance are found at ranks 2, 3 and 4 in Levenshtein's). However, this time, the experiments using wordforms do not allow to retrieve only one usage of *-(u)lo* -(으)로 but a confusing mix: most of the retrieved sentences are in fact subordinate clauses with no main predicate, and are therefore less easy to interpret.

We can also note from experiments on *-(u)lcito moluta* -(으)ㄹ지도 모르다 that searching for such a construction with several morphemes, and using the default mode (similar word(s), similar context), as well as wordforms, inevitably retrieves sentences that are almost identical in terms of syntactic construction, and even very similar in terms of meaning. The top three sentences ranked using the Levenshtein distance (but also retrieved by Jaccard) are evidence of this particularity.

Distributional Analysis Mode Testing the distributional analysis mode on *-(u)lo* -(으)로 amounts to searching for all of the adverbial particles (tagged `JKB`). Indeed, if we look at the results of Jaccard without lexical words in C.2.2, we note that the experiments allow to retrieve the allomorph *-ulo* -으로, the dative *-eykey* -에게, the locative *-ey* -에, the comparative morpheme *-pota* -보다 and many others. The Levenshtein distance gives similar results with and without lexical words. Only Jaccard with lexical words retrieves sentences that are much longer than that

of the query, and much more complicated to comprehend. The use of such results may not be relevant for language learners, but we can imagine that teachers of Korean could use them to create an exercise where learners have to determine the right particles to use. Indeed, if we look closely at the sequence of POS in Jaccard without lexical words for instance, we observe that the context is actually almost identical for all of the sentences: `NNG JX NNG JKB VV EF SF`. Interestingly, such an exercise compels the learners to focus on the meaning of words instead of the syntax.

In this mode, using lexical words to retrieve a construction based on several morphemes, such as *-(u)lcito moluta* -(으)ㄹ지도 모르다, can be glossed by "retrieve not exactly these words but similar, in a very similar context with very similar words". Such a search is most likely to retrieve the allomorph of the target, if any.

It is noteworthy that without lexical words, the results are roughly similar: it seems that even without lexical words (which implies that the verb *molu* 모르 'ignore' is *not* in sentences from the corpus, the system manages to retrieve constructions with this verb (9/10 for Jaccard, and 6/10 for Levenshtein). However, the system also retrieves constructions that are so far from the original query that they can hardly be considered as relevant (sentences 8, 9 and 10 in Levenshtein's result). In these constructions, the combination of POS (`EC JX VV`) is indeed the same but the meaning is quite different, which is not surprising considering the variety of items tagged either `EC`, `JX` or `VV`. The result would have been different with a more restricted POS, such as ETN (noun conversion endings), JKS (subject particles) or XSA (adjective derivation suffixes).

6.3.3 Similarity Measures

The similarity measures used in our experiments are the Jaccard distance (from Jaccard coefficient), the Sørensen-Dice distance (also from the Sørensen-Dice coefficient or index) and an implementation of minimum edit distance. The three measures were defined in Chapter 5, respectively in Section 5.5.2 (for the Jaccard and the Sørensen-Dice coefficients) and in Section 5.6. For this reason, we do not

present the similarity measures in this section, but focus on the specificities of their implementation, and on the interest of comparing such measures as well as the results of our experiments on doing so.

6.3.3.1 Objective(s)

Given that experiments of similarity measures between syntactic constructions have not been conducted yet to our knowledge, we naturally chose different similarity measures to be tested in our preliminary experiments. The objective is therefore to know whether or not those similarity measures, which are originally applied on other types of data (usually documents for the Jaccard and the Dice coefficients, and strings or, more recently, trees for minimum edit distance), are **relevant for our purpose**, and if they are, **to what extent**.

6.3.3.2 Implementation

Minimum Edit Distance In Section 5.6, we thoroughly presented the algorithm of edit distance applied on strings, its original and still most common objects. In our experiments, we changed the scale and adapted the computation of edit distance to sequences of words instead of strings.

Let us consider the two sentences we used to illustrate the Jaccard and the Dice coefficients: *I left everything like it was* and *I know what it feels like now*. If we consider them as strings, the minimum edit distance would be 21 if the three operations (addition, deletion, substitution) all cost 1. The following representation is one optimal alignment between the two strings to observe the common characters between the two (including spaces) and the characters that have to be added or deleted:

```
I  l e f t  e v e r  y t   h i n g  l i k e  i t  w a s
I  k n o w  w h a t  i t  f e e l s  l i k e   n o w
```

However, if we consider the two sentences as sequences of words rather than sequences of string characters, the minimum edit distance is only 5: (0) "`I`" remains

unchanged, (1) "`left`" needs to be substituted by "`know`", (2) "`everything`" needs to be substituted by "`what`", (3) "`like`" needs to be subsituted by the following word, "`it`", (4) the word "`feels`" needs to be added, (4) "`like`", which is now the fifth word, remains unchanged and finally, (5) the word "`now`" needs to be added.

```
I left everything like it    was
I know what       it   feels like now
```

Changing the scale was not the only adaptation we made: instead of keeping the cost of all three operations to 1, we changed the weight of operations based, not on their nature, but on the syntactic role of the manipulated data. The cost of substituting morphemes that are considered secondary role were reduced, while that of primary roles were augmented. For the preliminary experiments, we:

- reduced the substitution cost of adverbs (`MAG, MAJ`) and modifiers or morphemes composing modifiers (`MM, XSA, XSV`) to 0.5;
- reduced the substitution cost of interjections (`IC`), nominal prefixes (`XPN`) and noun derivation suffixes (`XSN`) to 0.1;
- augmented the subsitution cost of the heads of the sentences, in order words, predicates (verbs `VV`, adjectives `VA` and auxiliaries `VX`)to 1.5.

Of course, this adaptation is only preliminary and subject to improvement. For example, the highest cost is currently that of predicates, but we could also set a higher cost on grammatical morphemes such as particles, or set a different cost on the substitution of a particular lexical word instead of its POS. The script computing edit distance is available in Appendix Section B.2, and contains the adaptations to both Korean and to English. Indeed, as the adaptation of the weight of edit distance necessarily means targetting some POS or some words, this similarity measure is language-dependent, contrary to the other ones.

Jaccard and Dice In our script (shown in B.1), the two coefficients were implemented as distances, i.e., as their complementary functions, obtained by substracting the coefficient from 1. This explains why in the results we present, as well as in the output files (see Appendix C), the most similar sentences to the

query are the ones with the lowest score: the lower the distance, the more similar the sentences are.

Using distances instead of coefficients has no incidence on their computation or their ranking (apart from the fact that it is naturally reversed) but simply allows the ranking function to be the same for the three measures.

Unigram vs. Bigram In the illustration of the Jaccard and the Dice coefficients calculations in Section 5.5.2, each sentence is represented by a vector of words, which does not take word order into account, hence resulting in a 'bag-of-words' representation of the sentence (explained in Section 5.2). Indeed, these scores were calculated on vectors of sentences represented by isolated items, and therefore overlook some syntactic relations, including that the fact that the word "like" is different in sentence A and B. In A, "like" is a conjunction introducing the clause "it was" and could be replaced by "as", while in B, "like" does not introduce "now" but works with the verb "feel".

In order to take into account the word order in our experiments, we segmented sentences into bigrams, instead of unigrams as in this example. With bigrams, the two sentences *I left everything like it was* and *I know what it feels like now* would be reprensented by the following sets:
A = { `I_PNP left_VVD, left_VVD everything_PNI, everything_PNI like_PRP, like_PRP it_PNP,it_PNP was_VBD` }
B = { `I_PNP know_VVB, know_VVB what_DTQ, what_DTQ it_PNP, it_PNP feels_VVZ, feels_VVZ like_PRP, like_PRP now_AV0` }

With this representation, the intersection between A and B is *empty*. Given that each item is defined in combination with the preceding and the following item and that these two sentences do not share any consecutive common words, nothing matches. Using bigrams is therefore more *strict*, especially when each item is a couple `wordform/POS`, but allows all sentences containing "feels like" to be retrieved together, which can be useful if we are interested in this specific usage of the word "like". However, using bigrams is also relevant to retrieve some syn-

tactic constructions if we consider sequences of `POS` only instead of sequences of `wordform/POS`. In this case, if we replace "feels" by another verb tagged `VVZ`, we could retrieve verbs that take "like" as an privileged argument.

We initially wanted to compare traditional similarity measures (both the Jaccard distance and the Dice distance) results to that of the minimum edit distance. However, since the Jaccard and the Dice gave consistently the exact same ranking in our tests, we decided to use the Jaccard distance when using unigrams, and Dice distance when using bigrams.

6.3.3.3 Results in C.3

The most striking feature when comparing Jaccard and Dice distances to edit distance is that edit distance often retrieves sentences that have slightly the same number of words, given that each supplementary word implies an additional operation, which, in turn, implies more (edit) distance between the two words.

This specificity might be an advantage because, in our case, we use relatively short queries since they were transcribed from textbooks created for the use of students from beginner to intermediate level in Korean. Thus, we can imagine that if we use long sentences, Levenshtein would favour long sentences as well, which might indeed be more similar, but not necessary relevant nor easy enough to be helpful to language learners.

In addition, the Levenshtein distance attaches less importance to word order than Jaccard's distance using bigrams, and more to POS. This characteristic proved to me particularly interesting for a construction based on multiple morpheme such as *-lcito moluta* -ㄹ지도 모르다. Indeed, along with the allomorph of the target (*-ulcito moluta* -을지도 모르다 when the target is *-lcito moluta* -ㄹ지도 모르다), the system retrieved constructions that are not seen as such in any of the books we used, neither the Yonsei or the Ewha textbooks, nor the *Korean Grammar in Use*, nor even the *Korean Grammar for International Learners*.[31] Those constructions are:

[31] However, we note that those constructions, as well as other variations, are listed in Ross King's online "Korean Grammar Dictionary" on http://www.koreangrammaticalforms.com.

- *-**n**cito moluta* -ㄴ지도 모르다 and its allomorph *-**nun**cito moluta* -**는**지도 모르다;
- *-lci**nun** moluta* -ㄹ지**는** 모르다;
- *-l**nun**cito moluta* -ㄹ**는**지도 모르다, a peculiar construction resulting from the wrong (yet official) spelling of the concatenation of the early modern Korean form *-l i-itenci* -ㄹ 이-이던지.[32]

Indeed, the two constructions closest to *-(u)lcito moluta* -(으)ㄹ지도 모르다 found in the textbooks and grammars at our disposal are: *-(u)n/nuncito moluta* -(으)ㄴ/는지 모르다, composed of the indicative mood suffix *-n* -ㄴ to which the pre-nominal modifier suffix *-un* is attached, and followed by the defective noun glossed as '(the uncertain fact) whether' in Sohn [2013, p.57]. There is little semantic change in all of those constructions, only slight differences in degrees of uncertainty depending on whether the sentence is prospective or not (*-l* -ㄹ *vs. -n(un)* -ㄴ/는), on the emphasis put in the uncertain fact (the topical marker *-(n)un* -은/는 *vs.* the particle *-to* -도 'also'). Yet, we believe that these degrees should be taught at some point, not only so that the learner is not surprised to meet them (as the author initially was when observing the results) but also so that he or she can use them effectively.

6.3.4 Genres

6.3.4.1 Objective(s)

Given that searches with our system focus on syntactic similarity instead of strict matching based on keywords, all of the genres appear to be potentially relevant for language learners. The relevancy obviously depends more on the objective of their studies: learners who take Korean classes for the purpose of working in the media would probably be more interested in retrieving sentences extracted from newspaper articles or transcripts of news broadcasts while learners who are more focusing in speaking Korean would be more interested in transcripts of conversations.

[32] http://www.koreangrammaticalforms.com/entry.php?eid=0000000974

The objective of testing our system on different genres is to **show the possibilities** offered by similarity search **across genres**. We decided to comment on the experiments on *-(u)lcito moluta* -(으)ㄹ지도 모르다.

6.3.4.2 Implementation

For this parameter, the implementation is as simple as selecting a different sample for each search.

6.3.4.3 Results in C.4

Default Mode Before getting into the details of the results, it is noteworthy that newspapers and journals do not contain as many occurrences of *-(u)lcito moluta* -(으)ㄹ지도 모르다 as in books. Although it is a fact that the book genre has the most sentences among the three genres, this criterion does not seem to be the only one that plays in favour of this genre since the proportions are unequal: in newspapers, 42 sentences only were retrieved out of 54,022 (less than 0.0008%); in journals, 56 out of 50,392 (approx. 0.0011%) and in books 259 out of 177,998 (approx. 0.0015%). This may be due to the fact that both newspapers and journals tend to be factual and less incline to deliver uncertain facts.

Distributional Analysis Mode This mode was more interesting to study than the default mode in that it allowed very similar and relevant constructions to be retrieved.

Emphasising this metadata in the output either in an 'all genres search', or highlighting allowing to choose a genre – as observed in current corpus exploration tools such as The Lexicoscope and the corpus.byu.edu interface in Chapter 3 – could therefore raise awareness of the usages of each of them. Beyond the statistics on the uses of a construction in a specific genre, we showed that a similarity-based system is able to give alternatives.

6.4 Adaptation to English

We have mentioned in the introduction that this work was initially meant to be on French, but that we chose to study and conduct experiments on Korean instead. However, we do not intend to build a system specific to the Korean language. On the contrary, the whole system has been designed to be as generic as possible. However, this hypothesis has to be tested and the system adapted to another language. We chose to adapt to English, not only because English is internationally spoken and predominant in the scientific sphere, but also because it is typologically distant from Korean.

This section gives an account of the adaptations that were necessary to run our tests on English, as well as remarks on preliminary results.

6.4.1 Resources

For the experiments on Korean, we used sentences from textbooks as queries, samples of different genres from the Sejong Corpus as the corpus, and the KKMA tagger as the morphosyntactic analyser. Adapting the experiments to English implies to find resources that are equivalent to those we used for Korean. The resources we used for English are:

- queries: sentences from textbooks as queries, this time, from the *Advanced English Grammar in Use*[33];
- corpus: samples from the BNC World edition, composed of miscellenea (samples whose IDs start with`AM*`) from periodical and books from different domains, including "applied science", "imaginative", "leisure" and "world affairs";
- (morpho)syntactic analyser: the Free CLAWS tagger with the CLAWS5 tagset.[34]

[33]Hewings, Martin. *Advanced English Grammar in Use*, 3rd edition published in 2013 by Cambridge University Press.
[34]Available on `http://ucrel.lancs.ac.uk/claws/trial.html`.

6.4.2 Script Adaptations

Our system cannot be considered as *generic* – or *not* language-specific, unless it is possible to adapt the system from a language to another with little or no cost.

In our case, the modifications that were necessary to conduct experiments using the English resources are concentrated in the two scripts that constitute the core of our work: `similarity_measure.py` (shown in the Appendix, Section B.1) and `edit_distance.py` (shown in B.2).

Two of the three modifications were expected as they concern directly the language of application, but the third is related to the tagset, rather than on the language:

1. the removal of lexical words in the query and all of the sentences of the corpus (a parameter described in Section 6.3.2);
2. the computation of minimum edit distance;
3. the 'inverse' concordancing (to keep the sentences that do not contain the target word(s), see the "Modes Implementation" paragraph in Section 6.3).

Lexical Words Removing lexical words is a parameter that we implemented for the purpose of helping the system focus on grammatical words instead of lexical ones. In the experiments on Korean, the POS we decided to keep instead of the wordform they are attached to are `NNG` (common nouns), `NNP` (proper nouns), `NR` (numbers), `NP` (pronouns), `VV` (verbs), `VA` (adjectives)[35] and `MM` (determiners), as explained in Section 6.3.2.

The BNC samples we use are annotated using the CLAWS5 tagset.[36] The categories that we considered as 'lexical words', and that we therefore removed in the appropriate experiments are:

- nouns: `NN0` (number neutral nouns, such as *data*), `NN1` (singular nouns), `NN2` (plural nouns), `NP0` (proper nouns);

[35] Unfortunately, as pointed out in one of the pré-rapports of this dissertation, we did make a mistake and forgot to include adjectives in the list of lexical words made for the experiments on English. This mistake will be promptly corrected in the post-defense version of this dissertation.

[36] The whole tagset is available on http://ucrel.lancs.ac.uk/claws5tags.html.

- verbs: `VBD` (past form of *be*, i.e., *was*, *were*), `VBZ` (3rd person form of *be*, i.e., *is*, *'s*), `VHD` (past tense form of *have*, i.e., *had*, *'d*), `VHI` (infinitive of *have*), `VHN` (past participle form of *have*, i.e., *had*), `VHZ` (3rd person form of *have*, i.e., *has*, *'s*), `VVD` (past tense form of a lexical verb, such as *worked*, *slept*), `VVG` (-ing form of lexical verb, such as *working*, *sleeping*), `VVI` (infinitive of a lexical verb, such as *work*, *sleep*), `VVN` (past participle form of a lexical verb, such as *worked*, *slept*, *broken*), `VVZ` (3rd person form of a lexical verb, such as *works*, *sleeps*);
- adverbs: `AV0` (adverbs), `AVP` (adverb particles such as *up*, *off*);
- determiners: `AT0` (articles, such as *an*, *the*), `DPS` (possessive determiners, such as *yours*, *theirs*).

The tags from this list simply replaced those for lexical words in Korean in the `remove_lexical_item` function in the main script. Considering the syntactic properties of Korean and, in particular, the verbal endings, we decided to not include adverbs in this list. However, we did rehabilitate adverbs for English.

Of course, like the list of lexical word tags for Korean, this list leaves room to improvement. We could, for example, add pronouns as well, if we consider that their wordforms (personal pronouns such as *they* or *it* or indefinite pronouns such as *none* or *everything*) do not play a major role in the identification of syntactic constructions.

Edit Distance The differences between the `edit_distance` function for Korean and the `edit_distance_en` function for English are shown in the script in the Appendix, Section B.2.

In order to remain consistent with the decisions we made for Korean, we simply adapted the tags from the Sejong Corpus tagset to CLAWS5 tagset: the categories whose substitution should cost less are still the modifiers (adjectives, adverbs, articles including numerals) and the interjections; likewise, those whose substitution should cost more are still the head of sentences or clauses (all of the verbs, including *be* and *do*).

The cost of these first two adaptations is actually not that heavy if we consider that parameter files, such as those that are needed to use tools like the TreeTag-

ger,[37] Compared to them, the third was unexpected and dealt with while running the experiments.

Issue on Concordancing For the default search mode, the first step before the computation of the similarity measures consists in selecting the sentences that do contain the target word(s). This primary step is used not only to reduce the noise in output, but also to save time (which can be precious on a very large corpus).

Conversely, for the distributional analysis search mode, the first step consists in selecting the sentences that do *not* contain the target word(s), but their POS. For example, if the target is the preposition *like* tagged `PRP` (for prepositions except *of*) and if we use this mode, sentences that contain `like_PRP` will not be retrieved, but those that contain `towards/PRP` will.

However, in the CLAWS5 tagset, many tags are used for only one or two wordforms. We have seen, for instance, the tag `VBD` which is used to annotate the two past forms of 'be', *was* and *were*. Similarly, the `VHD` is only used to annotate the past form of *have*, *had* and its abbreviated form *'d*. It is therefore impossible to find a word annotated `VHD` and that is neither *had* nor *'d*. This characterisic led us to adapt the internal concordancing for the distributional analysis by exceptionally allow sentences that do not contain the POS of the target word(s) to be retrieved.

6.4.3 Preliminary Results

Experiments were led on two types of constructions: the past perfect continuous form, 'had been V+ing', and the *like* as a preposition.

Default Mode Using the Jaccard distance with bigrams, our system retrieves only relevant sentences containing the past perfect continuous form, including non-contiguous forms such as "[...] the European Commission had **also** been working on measures [...]" (sentence 4, with lexical words).

[37] TreeTagger is a widely used part-of-speech tagger that is applicable to many languages. While the core of the algorithm remains the same across languages, language-specific parameters – starting from the tagset! – are contained in "parameter files", one for each language or variety of language (in fact, there is one file per model built for a language). Details are available online on http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/.

However, the ranking obtained with the Levenshtein distance contains sentences that do not correspond to the target construction. Indeed, sentences 1, 6, 7 and 8 (with lexical words) and sentences 1, 4, 5 and 8 (without lexical words) do contain the words *had*, *been* and a progressive verbal form but the latter is not part of the same verbal phrase. They are in fact past perfect forms, followed by another clause containing a progressive verbal form.

For the preposition "like", the results are not so satisfying considering that Jaccard retrieves very long sentences (more than 30 for some of the sentences). Moreover, all of the retrieved sentences do display "like" as a preposition and not a verb, but, most of the time, "like" is used to introduce noun phrases, despite the fact that it is used to introduce a clause in the query.

Levenshtein does retrieves much shorter sentences, as we already noticed in experiments on Korean.

Distributional Analysis Mode This mode allows similar constructions to be retrieved, and *not* the one in the query. Using the past perfect continuous form as the target construction to avoid retrieves other complex tenses, namely and by order of frequency:

- the past perfect with a lexical verb – `'d_VHD VVN` or `had_VHD VVN`, as in *How did you know I'd met him*, sentence 3 in Jaccard with lexical words;
- the past perfect specific to the verb *be* – `'d_VHD VBN` or `'d_VHD VBN`, as in *Endill had been punished also*, sentence 7 in Jaccard without lexical words;
- the present perfect continuous – `'ve_VHB VBN VVG` or `have_VHB VBN VVG`, as in *I've been searching for ages*, sentence 2 in Levenshtein with lexical words.

These remarks remind those made on the construction *-(u)lcito moluta* -(으)ㄹ지도 모르다 and its variations (see the results of Section 6.3.2), except that the "variations" found here are actually different tenses of English.

As for the case of the preposition "like" using the distributional analysis mode, it is more similar to the case of *-(u)lo* -(으)로 (also in the results of Section 6.3.2).

In both cases, teachers could use results from this mode to create Cloze Exercises where the learner has to find the right particles for Korean, and the right tense or the right preposition for English.

6.5 Conclusion

While Chapter 5 consisted in defining the requirement specifications of an example-based and similarity-based system, this chapter aimed at serving as a *proof of concept*; in other words, it aimed at demonstrating the feasibility of our ideas, as well as giving preliminary results on which further experiments can be built.

We focused our experiments on the system configuration of the input, of the similarity measures, and of the search modes. The numerous combinations of parameters resulting from the different configurations makes it difficult to tell exactly which options are the most relevant for a specific task. Indeed, these preliminary experiments only provide arguments to draw temporary conclusions that need to be confirmed with a proper evaluation by end users.

With regard to the **input**, we can assert that a single input is enough to determine a target construction and its context of use and that the two types of input we tested – using both wordforms and POS, and using only the POS of lexical words – are apparently relevant for different reasons.

Among the three **similarity measures** we implemented, none seemed to work significantly better than another for now. What we are sure of is that **edit distance** retrieves sentences with similar length to the query – in our case, relatively short sentences, since the sentences illustrating grammar points are shorter than the average sentence that can be found in our written corpus for example. This observation is rather an advantage given that it means that the retrieved sentences are more likely to be of similar complexity as the query. However, the main advantage of edit distance is to allow the weighting of editing costs to be manually set but we did not test different weights. The weighting of editing cost have to be discussed and refined with further experiments. On the other hand, the **Jaccard distance applied on bigrams** is the only measure we tested that takes into account the word order, and has the advantage of being ready to use 'out of the box'.

The fact that this measure does not need any configuration means that, compared to edit distance, it is not only more objective and based on data rather than on the intuitions of a specialist, but also that it makes the adaptations to another language much easier. The last measure, **the Sørensen-Dice distance** applied on unigrams, did not have any significant advantage over the two other measures.

Finally, the two modes we described in these experiments both proved that they could be used for different purposes, in accordance with the purposes they were designed to serve: the **default mode** allows to retrieve the same construction in the same context and with the same meaning when combined to the right parameters (namely, the Jaccard distance applied on bigrams in C.2.1); the **distributional analysis mode**, when applied to a single morpheme, provides sentences showing the use of other words in the same context (which can be used to create Cloze Exercises), while the application to a construction with multiple morphemes provides sentences showing the variations of this construction.

Lastly, the choice of a genre is left to the user, either the teacher who wants to contrast data based on written or spoken genres, or the learner who is specialised in one of the proposed genres, or even simply the curiosity of any user who would like to see a grammar point illustrated in different genres.

Considering those observations, we believe that, in spite of an unconcealed need for further experiments, these preliminary results are positive on the practical potential of our system.

Chapter 7

Conclusions and Perspectives

7.1 Conclusions

Given that our work touches upon various *fields* and borrows ideas, concepts and hypotheses from academic disciplines of different backgrounds and concerns, this dissertation was described as a journey across these various fields. As mentioned in the introduction, the chapter order only reflects our own peregrination and the reader's route has not necessarily followed ours.

That being said, we may conclude this dissertation with a linear summary of the state-of-the-art and of our contributions, to ensure that the reader did not stray too far. This conclusive report is then followed by an overview of the perspectives awaiting our work.

7.1.1 Summary of the State-of-the-Art

The dissertation was introduced with the definition of a need, leading to the definition of our research problem: how to make attested examples of a given syntactic construction accessible to language learners. The key to the problem involves technical solutions, but the issue originates from *language learning*, a field that we first defined in opposition to *language acquisition*. However, this strong dichotomy hides a more positive relation between the two fields: research in language acquisition provides insights into the mental processes involved in the acquisition of both first

and second languages, and therefore gives a theoretical ground on which language learning and teaching methods can be built. Among those insights, we focused on the processing of linguistic input, especially *salient* and *comprehensible input*. Its crucial role led us to examine the linguistic data to which language learners are exposed: teacher-talk, foreigner talk and interlanguage talk through interaction on the one hand, and on the other hand, both *authentic* and *non-authentic* materials.

We consequently advocated the use of *native corpora* as a complementary source of input, albeit we agree that native speakers should *not* be seen as models for foreign learners and that their status of '*ideal* speakers' is delusive. In fact, native corpora are useful in so far as – and precisely because – they display what *may* be encountered in real life, what *can* be said but not necessarily what *should* be said. In addition to an indirect use of native corpora through concordance handouts or statistical inferences, we argued that *direct exposure* to native corpora, as proposed by John's Data-Driven Learning methods, is also beneficial to language learners.

Indeed, we showed that the expansion of Corpus Linguistics and technologies from Natural Language Processing goes hand in hand with the development of larger and more diverse corpora: over recent years, reference corpora have been built for a wider range of languages, as well as the constitution of smaller specialised corpora, with various purposes and applications. In any case, efforts have been made to provide richer annotations, ranging from morpho-syntactic to semantic annotations, as well as annotations on gestures and postures in multimodal corpora. Corpus exploration tools have undergone development accordingly, to meet the needs of researchers by allowing the search for sophisticated patterns.

However, this sophistication has a cost and we showed that, despite efforts towards simplifying interfaces and query languages, beyond simple queries, current corpus exploration tools have not been sufficiently adapted to non-specialist users such as language learners or even teachers. The least thing that is required to be able to make queries on syntactic constructions is to know not only the (morpho)syntactic tagset of the corpus to be investigated, but also the syntax of the query language. Since this level of technicality and the necessity to undergo a specific training course may be enough to drive non-specialists – both learners and teachers – away from corpora, our contributions consist in providing solutions

to allow novice users to rely not simply on a subsidiary knowledge but on the combination of an algorithm and their intuition.

7.1.2 Contributions

The core of our work is the specification of a system that allows syntactic constructions to be sought without prior knowledge in linguistics or in natural language processing, nor in any language or computer-related field. The originality lies in the processing chain that we constructed by assembling pre-existing concepts and tools:

1. the concept of example-based system, which spares the user from learning a query language and simply uses natural language instead;

2. a morphosyntactic analyser or a syntactic parser, providing (morpho)syntactic tags automatically;

3. similarity measures, which make it possible to go beyond strict matching.

In order to demonstrate that this processing chain is viable, we conducted preliminary experiments on the system configuration. Due to lack of time, we only performed tests on the similarity computation part of the system, leaving clustering experiments to perspectives. Our experiments showed positive results in the use of similarity measures (Jaccard/Dice, edit distance) and thus serve as a proof of concept, including on the relevancy of the three different modes that we proposed and the potential genericity of the tool.

Linguistically speaking, our system has demonstrated a certain capacity for serendipiditous findings based on syntactic construction similarity. We hope that the most relevant of these findings offer food for thought, not only for linguists, but also for both language teachers and learners, as it did for us in regards to Korean grammar points.

However, we have also seen that our system lacks relevant evaluation and should be tested at least on the three potential types of users: language learners and language teachers for the simplified version, and linguists trained in Corpus Linguistics for the expert version. Those evaluations would provide us with concrete feedback

on the system configuration and would help us to consider other adjustments and possibilities on further experiments. In the meantime, we envisage the following perspectives.

7.2 Perspectives

This hybrid work between its core in Natural Language Processing and its application to Language Learning and Teaching entails a dual track for perspectives: the first track leads to an overview of the technical experiments that we planned as a continuation of the preliminary experiments that we conducted, whereas the second track is oriented towards a more concrete integration of our system in language learning and teaching.

7.2.1 Further Experiments on System Configuration

In order to refine the specifications that we laid out, further experiments are needed to validate the whole processing chain and improve the results to a more satisfactory level.

clustering Users of tools such as concordancers are often confronted to 'overwhelming' results, since a simple query may match thousands of examples, if not more, in the case of a general query on a large corpus. In a similarity-based system, this phenomenon could be worse, given that any sentence in the corpus is similar to the query to various degrees. We therefore limited the number of retrieved sentences to a hundred in our preliminary experiments, but this threshold is arbitrary and even reduced, the overload remains an overload.

The solution that we proposed in Chapter 5 but have not been able to test yet is to divide retrieved sentences into *clusters*, i.e., groups of data objects resulting from an *unsupervised classification*. Contrary to *supervised* classification where *classes* (or clusters) are pre-defined and the algorithm is given examples of correct classification, unsupervised classification is a method where the algorithm has to find the most appropriate way to categorise objects given a set of *features*. In some models, such as k-means, only the number of clusters (along with a distance) has to be pre-defined. In our case, objects are the sentences that are most

similar to the query, and features are the words and/or morphosyntactic tags of each sentence. Among the many clustering models, centroid-based clustering is particularly interesting for us because each cluster is represented by its centroid, i.e., the arithmetic mean position of all objects in a cluster. Additionally, for each cluster, it is also possible to compute the medoid, the document vector that is closest to the centroid. In other words, this method allows the most representative (or central) sentence of each cluster to be picked. Instead of assailing the user with an overwhelming stream of results, we would be able to present only the medoids of a limited number of clusters.

Preclustering We also thought of applying clustering on the corpus as a preprocessing, prior to any query. This preclustering would allow the similarity computation step to be accelerated, since the query would be compared to the medoid of each pre-defined cluster instead of being compared to each sentence of the corpus. Depending on the size of the corpus to be investigated, this preprocessing could be crucial: while pre-clustering implies a supplementary step, it has the advantage of being efficient because it needs only one execution for all. This preprocessing is similar to the indexation of a corpus in a typical information retrieval task.

Contrary to the *ad hoc* clustering that we described above, preclustering is applied on the whole corpus with no preselection or no specification regarding what to focus on. Indeed, the *ad hoc* clustering benefits from the fact that, in previous steps, the user has already defined the focus of the query (see Step 2 in Section 5.3.3) and clustering is only used to discriminate retrieved sentences. Considering this difference, it may therefore be interesting to apply a *fuzzy clustering* model that allows each object to belong to different clusters. Indeed, hard clustering only assigns one class to each object, while a sentence is likely to contain several grammar points and should consequently be in a position to match as many queries.

Reannotation of the Corpus Another experiment that we hinted at in Chapters 4 and 5 is the reannotation of the corpus using the same morphosyntactic analyser as for the input.

In our experiments, we used the Sejong Corpus, which, as the Korean National Corpus, has benefitted from years of work and correction. Thus, on the one hand,

although probably not perfect considering the size of the corpus (roughly 13 million "words" for the POS-tagged version), the annotations of the corpus are still the gold standard. On the other hand, we annotate the sentence(s) input by the user dynamically using a morphosyntactic analyser and we do not leave any possibility of correction. This means that in case of an error of segmentation or POS-tagging, the input containing one (or several) error(s) is still used as a query and is compared to nearly error-free sentences from the corpus. Depending on its nature, the error is at least likely to decrease the similarity score of relevant sentences in favour of less relevant sentences. At most, it may tamper with the similarity computation and lead the user's investigation along the wrong track.

Reannotating the Sejong Corpus with the morphosyntactic analyser that we integrated to our processing chain (KKMA) may in fact increase the performance of our system. This is possible thanks to the algorithmic nature of the annotation: as a matter of fact, if the morphosyntactic analyser outputs the wrong segmentation or the wrong tag for a given word, it is likely to do so invariably and tirelessly in all similar contexts.

Nonetheless, it may be interesting to keep the clean version of the Sejong Corpus, at least for the output. While errors of segmentation or annotation are helpful for the similarity computation, an incorrect segmentation or annotation in the output is potentially confusing for the user. We could therefore combine both the corrected and the non-corrected version of the Sejong in further experiments. Whereas the latter would serve as the corpus of comparison with the input, only the corresponding error-free sentences from the former would be used in the output.

Parse trees We mentioned in this dissertation another possibility to improve the output of our system: the use of parse trees instead of relying merely on parts-of-speech annotations. POSs are indeed fundamental because they are often the very first layer of annotation (as shown in Chapter 3), but they are not the most efficient means to target some syntactic constructions. For example, let us consider the following sentences automatically annotated with the free version of the WWW tagger[1] using the CLAWS7 tagset that we already used in Chapter 5:

[1] http://ucrel.lancs.ac.uk/claws/trial.html

(21)	the	girl	*with*	*a*	*tattoo*	has	been	sleeping	for	hours	.
	AT0	NN1	PRP	AT0	NN1	VHZ	VBN	VVG	PRP	NN2	SENT
(22)	the	*tattooed*	girl	has	been	sleeping	for	hours	.		
	AT0	AJ0	NN1	VHZ	VBN	VVG	PRP	NN2	SENT		
(23)	that	girl	has	been	sleeping	for	hours	.			
	DT0	NN1	VHZ	VBN	VVG	PRP	NN2	SENT			

We notice that all sentences are not only almost identical in terms of meaning, but all of them also contain a present perfect continuous verbal form. If the object of the query launched by the user only focuses on this tense, the three sentences are equally relevant but in terms of similarity measure, as well as edit distance based on POS, they are considered different. If we only take words as units, the minimum edit distance between Examples 21 and 22 is 4, and is also 4 between Examples 21 and 23, and 2 between Examples 22 and 23. However, the minimum edit distance between all sentences can be reduced to only 1 or 2 if we consider either constituency or dependency parse trees: modifiers *with a tattoo* and *tattooed* are both adjectival phrases and are therefore possibly added, deleted or substituted as a single subtree, as shown in Figure 7.1.

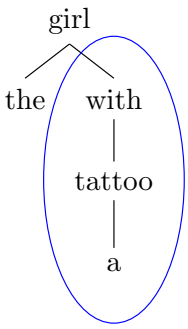


Figure 7.1: Dependency tree of the noun phrase *the girl with a tattoo*

Incidentally, this solution is already implemented in GrETEL's system. In this present study, we decided to use a morphosyntactic analyser of Korean as the provider of syntactic information on which the whole system is then based to compute similarity. This choice was made in regard to the genericity of the tool and the relative rarity of both treebanks and parsers in languages that are less equipped than English, French or Korean. In addition, following the example of the Sejong

Corpus, most syntactically parsed corpora are smaller than morphosyntactically tagged corpora, and therefore provide less examples to study.

Genericity of the Tool Experiments on the genericity of our system were only preliminary as well. Since we were careful to use the least language-specific configurations possible, we are genuinely eager to confront our system to more languages, especially languages that are syntactically different from an agglutinative language such as Korean. Since language-specific configurations are particularly important for the computation of minimum edit distance (POSs are not equally nor universally essential to syntactic constructions!), we would need the collaboration of specialists in other languages.

7.2.2 Towards a Pedagogical Tool

Our work consisted in defining the requirement specifications of a new *function* (rather than a complete tool, like the ones we presented in Chapter 3) allowing an original use of native corpora in language learning. Although it is designed specifically to be used by university students as well as self-directed language learners and language teachers, it does not compare with a real pedagogical tool. The efforts that we made were only focused on the simplification of a corpus exploration tool for *non-specialists* in general, and nothing specific for language learners. For this reason, the guidance of a teacher might be crucial, naturally, for beginners, although not only for the latter, especially as we chose to explore monolingual corpora exclusively.

We cannot but recommend the tips that Kennedy and Miceli [2001] gave to their language learners to help them in investigating corpora in their target language. The purpose of this Data-Driven Learning study was specifically to understand "*how* learners actually go on investigations" in the absence of a teacher, which is an important question if we consider that their goal is to encourage the use of a corpus exploration tool outside the classroom. Claire Kennedy and Tiziana Miceli divided corpus exploration into four steps:

1. Formulate the question;

2. Devise a search strategy;

3. Observe the examples and select relevant ones;

4. Draw conclusions.

While their study focuses on identifying and solving problems that can be overcome with appropriate training on the use of a corpus exploration tool rather than on language itself, Kennedy and Miceli acknowledge that the two steps where linguistic proficiency has the most significant influence are Steps 1 and 3:

> "In Step 1, for instance, appreciation of whether it makes sense to ask a given question depends to some extent on familiarity with the target language. In Step 3, of course, not understanding the examples can undermine even an impeccably conducted investigation." [Kennedy and Miceli, 2001, p.82]

The first step is indeed far from an easy task. Understanding what type of question might be answered by a concordancer and not from a dictionary or a grammar book is difficult, formulating a specific question and defining an effective and efficient strategy to get an adequate answer are even more so. With regard to these issues, we believe that the fact that our system is example-based and that users can test the different modes easily at least encourages them to try and to reflect on given outputs.

However, our system does not provide any help concerning the third step. When observing the examples retrieved, students were apparently easily distracted from their initial search due to lack of rigor in paying close attention to meaning and structures: sometimes they would rely on sentences where the target word does not have the same meaning as in their initial search, sometimes they would rely more on irrelevant sequences of words rather than on correctly segmented phrases.

Regarding the complexity of output sentences and the cognitive difficulty, several options could be implemented in our system.

First of all, the output could be linked to a **monolingual or a multilingual dictionary** so that users could simply hover on words to display their meaning(s),

or click on them and be directed to an online dictionary. This option was seen in Chapter 4 where we described KKMA's concordancer giving a direct link to the entry of the selected word in Naver's bilingual dictionary.

Secondly, the output could be enhanced using a **colour-coded grammar** similar to what was proposed for FipsColor [Nebhi et al., 2010]. Each part-of-speech is not explicitly shown but represented by a specific colour. This option is in line with the "grammaire en couleur" (literally, *grammar in colours*) approach originally advocated by Laurent [2004] and adapted to university students by Boch and Buson [2012]. This method is inspired by Caleb Gattegno's Silent Way and is based on a constructivist and inductive approach.

Finally, the output could be improved by preprocessing the corpus: to ensure that the sentences retrieved are not too complex for the user, each sentence, or at least each sample of the corpus, could be categorised in terms of genre but also in terms of **readability** degree, i.e., of reading difficulty. Based on machine learning methods, the assessment of text readability is not only efficiently automated, but is also able to rely on a large number of features, ranging from (lexico-)semantics, morphology, syntax, discourse to pragmatics [Collins-Thompson, 2014]. Given the importance of such assessment for communicative and educational purposes, algorithms have been developed for numerous languages including Arabic [Al Tamimi et al., 2014], French [François, 2014], Japanese [Sato et al., 2008], Thai [Daowadung and Chen, 2011], and also Korean but either on a specialised genre [Wha et al., 2011] or using only typographical features [Yi et al., 2011].

Even if an algorithm does not exist in a given language, a primary attempt at readability can be computed based on the sentence length (number of characters, words, or morphemes could be interesting for Korean) and on the vocabulary. For instance, if our system is integrated into a platform built for educational purposes, where the vocabulary learners are exposed to is layered and distributed in semesters, we can imagine that the system can match the level of the learner, found in his or her profile, to the vocabulary of retrieved sentences so that they contain a minimum of X% of known vocabulary.

The system that we built is not a pedagogical tool *per se*, but we believe that this program could in the end complement current pedagogical resources by

offering an original focus on the grammatical constructions of the target language, and, why not, find interesting applications in other disciplines as well.

Appendix A

What You Need to Know About Korean

A.1 General Presentation

According to the Ethnologue website[1], Korean has 77,166,230 speakers worldwide as of 2010, approximately 77,100,000 (62%) of which live in South Korea, approx. 23,300,000 (30%) in North Korea (2008 census), a little more than 2,700,000 (3.5%) in China (most in the Jilin (Kirin) and Yanbian (Hyanbian) Provinces, 2012 census) and approx. 900,000 (0.01%) in Japan (2011 census).

Typology Regarding subject-verb-object positioning, Korean is a typical 'SOV-order language', or a head-final language. Objects conventionally follow subjects and are followed by a predicate. This property also explains why Korean does not use prepositions but postpositional particles, and why modifiers always precede the head of the clause. Example 24 can be analysed as follows:
[[[[[[ton-ul]$_{NP}$ ilh-un]$_{VP}$ na-nun]$_{NP}$ cwuk]$_{VP}$-ko]$_{NP}$ siph-ess-ta]$_{VP}$,
where each head is the final word of the clause.

However, this order is not rigid and word order before the predicate is relatively free and is often scrambled for stylistic purposes or to emphasise a specific element. It is also possible to postpose constituents after the predicate, which happens especially in conversational interactions. Compare the word order in Examples 24 and 25: in the second example, both object and subject were omitted in the main clause and added afterwards.

[1] http://www.ethnologue.com/17/language/kor/

Morphologically-speaking, Korean is an agglutinative language. In other words, in Korean, morphemes agglutinate to form words and each of the morphemes usually has a consistent form and a single meaning, albeit not always, as seen in Example 24 where the *-un* -은 cumulates two functions as the past pre-nominal modifier suffix.[2]

(24) 돈을 잃은 나는 죽고 싶었다 [Sohn_64]
ton-ul ilh-un na-nun cwuk-ko siph-ess-ta.
money-AC lose-PST.MD I-TOP die-NMLZ want-PST-DC
'I, who lost money, wanted to die.'

(25) 오늘 아침 산책하다가 물렸어 개한테
onul achim sanchaykha-taka mwulli-ess-e kay-hanthey
today morning walk-while be.bitten-PST-INF dog-by
내가 [Sohn_212]
nay-ka.
I-NM
'While taking a walk, (I) got bitten, [by a dog, me]!'[3]

Script Korean used to be written solely using the 이두 Idu script based on sinograms, or *hanca* 한자 in Korean. In the fifteenth century, an alphabet called *hankul* 한글 was created by the court of King Sejong and has since undergone major changes due to the changes in the Korean phonological system. While the use of sinograms was officially banned in North Korea since 1949, it declined in South Korea but is still common and *hanca* 한자 are still taught to both natives and learners of Korean nowadays. Indeed, *hanca* 한자 are used in some contexts and for different reasons such as the disambiguation of homonyms, or as a convention – names or, in newspapers, countries such as China, Japan and North Korea are often represented by the first sinogram of their name, namely 中 (*cwung* 중), 日 (*il* 일) and 北 (*pwuk* 북) – or simply for their aesthetics. Yet, we can say *hankul* 한글 is the main script for Korean nowadays and has replaced sinograms in most contexts, albeit not all of them.

[2] In Korean, verbs have a past (pre-nominal) modifier suffix, as well as a non-past modifier suffix (*-nun* -는) and a prospective one (*-(u)l* -(으)ㄹ, the one seen in the -(u)lcito moluta -(으)ㄹ지도 모르다 construction, which we study in Section 6.3.

[3] Originally translated "While taking a walk, (I) got bitten, me by a dog" but we decided to keep the literal word order.

The current *hankul* 한글 alphabet is given at the very beginning of this dissertation, along with the transliteration for each letter, and their pronunciation in International Phonetic Alphabet (IPA).

A.2 Korean Grammar

In this section, we first define what is a part-of-speech in the Korean language before focusing on how Korean grammar is taught to foreign learners by analysing grammar points extractd from textbooks.

A.2.1 Parts-of-speech in Korean

Linguists do not agree on the number of word classes, or parts-of-speech, in Korean: their representation vary roughly from seven to eleven main POS. We agree with the classification presented in Sohn [2013, pp.216-228] except that we add an eighth category for interjections, which Sohn considers as "discourse adverbs". The eight POS and their subcategories are recapitulated and illustrated below:

- Nouns (명사)
  - Numerals: *hana/il* 하나/일 '1', *twul/i* 둘/이 '2', *seys/sam* 셋/삼 '3'[4]
  - Counters (or classifiers) *mali* 마리 for animals, *myeng* 명 for people, *khillo* 킬로 'kilo'
  - Proper nouns: *sewul* 서울 'Seoul', *seycongtaywang* 세종대왕 'King Sejong the Great'
  - Defective nouns: *ccok* 쪽 'direction', *mankhum* 만큼 'as much as', *ppwun* 뿐 'alone'
  - Verbal nouns: *kongbwu* 공부 'study', *nothu* 노트 'note'
  - Adjectival nouns: *hayngpok* 행복 'happiness', *simsim* 심심 'loneliness'
  - Common nouns: *ppang* 빵 'bread', *kongsancwuuy* 공산주의 'communism', *en.ehak* 언어학 'linguistics'

[4]Korean has a dual numeral system: Sino-Korean and native. The Sino-Korean system is complete whereas the native system is defective. However, this is not the only reason why these two sets are complementary: in certain contexts, the Sino-Korean system is expected, while in others, the native one is expected. For example, when telling time, the native system is used for the hour, and for the minutes, the Sino-Korean system is used.

- Pronouns (대명사)
  - Personal pronouns: *na* 나 'I', *cehuy* 저희 'we [humble]'
  - Interrogative-indefinite pronouns: *nwukwu* 누구 'who', *encey* 언제 'when'
  - Demonstrative pronouns: *i* 이 'this', *ku* 그 'that', *yeki* 여기 'here'
- Verbs (동사)
  - Main verbs: *mekta* 먹다 'eat', *swumta* 숨다 'hide'
  - Auxiliary verbs: *pota* 보다 'try', *pelita* 버리다 'finish up'
- Adjectives (형용사)
  - Copula: *ita* 이다 'be'
  - Existantial adjectives: *issta* 있다 'exist, possess', *epsta* 없다 'not exist, not possess', *kyeysita* 계시다 'exist, stay [honorific]'
  - Sensory adjectives: *mwusepta* 무섭다 'be afraid', *kipputa* 기쁘다 'be happy'
  - Descriptive adjectives: *pharahta* 파랗다 'be blue', *napputa* 나쁘다 'be bad'
- Adverbs (부사)
  - Negative adverbs: *an(i)* 안/아니 'not', *mos* 못 'cannot'
  - Attributive or property adverbs: time *pelsse* 벌써 'already', place *melli* 멀리 'far away', manner *panccakpanccak* 반짝반짝 'glitteringly', degree *cemcem* 점점 'gradually')
  - Modal adverbs: *ama* 아마 'perhaps', *hoksi* 혹시 'by any chance', *ceypal* 제발 'please'
  - Conjunctive adverbs: *tto* 또 'again', *kuliko* 그리고 'and, then', *kulena* 그러나 'but'
  - (Discourse adverbs, see interjections)
- Determiners (관형사)
  - Demonstrative determiners: *i* 이 'this', *ku* 그 'that', *ce* 저 'that over there'
  - Specifiers: quality *mwusun* 무슨 'what kind of', *say* 새 'new' and quantity *han* 한 'one', *motun* 모든 'all'
- Particles (조사)
  - Case particles: the genetive particle *-uy* -의, *-hako* -하고 'and', *-(u)lo* -(으)로 'towards, by, as'

  - Delimiters: *-man* -만 'solely', *-mata* -마다 'each, every'
  - Conjunctive particles: the quotation particle *-ko* -고

- Interjections (감탄사): *aiko* 아이고 'oh!', *yey* 예 'yes', *emena* 어머나 'oh my!'

Sejong tagset Table A.1 shows the tagset used in the Sejong Corpus as a whole. Each tag is presented with its description, authentic example(s) from the corpus and its overall frequency rate if it occurs in both written *and* spoken corpora.[5] Rates in parentheses were only calculated on either the written corpus (`SS`, `SE`, `SO`, `SW`, `NA`, `SL`, `SH`, `SN`) or the spoken corpus (`UNA`, `UNC`, `UNT`).

Contrary to our classification given earlier, the word classes used in the Sejong Project are 13. However, the differences are rather minor:

- some of the POS from our classification are grouped: nouns and pronouns in substantives, verbs and adjectives in predicates;
- some are divided: in the Sejong tagset, nouns particles and verbal endings are different word classes;
- some were created specifically to annotate corpora: punctuations, unanalysable words (for instance, due to typos in written samples, and inaudible in spoken samples), prefixes/roots/suffixes[6] and foreign words.

Determiners, adverbs and interjections belong to both classification, *a priori* similar.

Class	POS	Description	Examples	%
Substantives (체언)	NNG	Common nouns 일반 명사	*mal* 말, *salam* 사람	23.63%
	NNP	Proper nouns 고유 명사	*hankwuk* 한국, *mikwuk* 미국	2.18%
	NNB	Bound nouns 의존 명사	*kes* 것, *swu* 수	3.27%
	NR	Numbers 수사	*hana* 하나, *man* 만	0.39%

[5] The Korean description and the statistics were extracted from the table on: http://kkma.snu.ac.kr/statistic?submenu=postag.

[6] Of course, prefixes, roots and suffixes are not literally 'word' classes, but these categories are needed in annotated corpora because the minimal unit is not the word but the morpheme, as explained in 3.5.2.

	NP	Pronouns 대명사	*na* 나, *ku* 그	1.67%
Predicates (용언)	VV	Verbs 동사	*ha* 하, *iss* 있	7.84%
	VA	Adjectives 형용사	*eps* 없, *kath* 같	1.78%
	VX	Auxiliaries 보조 용언	*cwu* 주, *anh* 않	2.22%
	VCP	Copula 'to be' 긍정 지정사, 서술격 조사 '이다'	*i* 이	1.95%
	VCN	Copula 'not to be' 부정 지정사, 형용사 '아니다'	*ani* 아니	0.20%
Determiner (관형사)	MM	Determiners 관형사	*ku* 그, *i* 이	1.45%
Adverbs (부사)	MAG	Common adverbs 일반 부사	*te* 더, *tto* 또	2.93%
	MAJ	Conjunctive adverbs 접속 부사	*kulena* 그러나, *kuliko* 그리고	0.45%
Interjections (감탄사)	IC	Interjections 감탄사	*kuray* 그래, *ani* 아니	0.27%
Particles (조사)	JKS	Subject particles 주격 조사	*-i* -이, *-ka* -가	2.64%
	JKC	Complement particles 보격 조사	*-i* -이, *-ka* -가	0.24%
	JKG	Adnominal endings 관형형 전성 어미	*-uy* -의, *-u* -으	2.27%
	JKO	Object particles 목적격 조사	*-ul* -을, *-lul* -를	3.58%
	JKB	Adverbial particles 부사격 조사	*-ey* -에, *-ulo* -으로	4.37%
	JKV	Vocative particles 호격 조사	*-a* 아, *-ya* 야	0.02%
	JKQ	Quotation particles 인용격 조사	*-ko* -고, *-lako* -라고	0.08%
	JX	Auxiliary particles 보조사	*-un* -은, *-nun* -는	4.04%
	JC	Conjunctive adverbs 접속 부사	*-kwa* -과, *-wa* -와	0.64%
Endings (어미)	EP	Prefinal endings 선어말 어미	*-ess* -었, *-ass* -았	2.42%
	EF	Final endings 종결 어미	*-ta*, *-nta* -ㄴ다	3.61%

	EC	Connective endings 연결 어미	*-ko* -고, *-e* -어	7.52%
	ETN	Noun conversion endings 명사형 전성 어미	*-ki* -기, *-m* -ㅁ	0.55%
	ETM	Det. conversion endings 관형형 전성 어미	*-n* -ㄴ, *-nun* -는	5.81%
Prefixes (접두사)	XPN	Substantive prefixes 체언 접두사	*cey-* 제-, *pwul-* 불-	0.20%
Suffixes (접미사)	XSN	Noun derivation suffixes 명사 파생 접미사	*-tul* 들, *-cek* 적	1.86%
	XSV	Verb derivation suffixes 동사 파생 접미사	*-ha* -하, *-toe* -되	2.48%
	XSA	Adj. derivation suffixes 형용사 파생 접미사	*-ha* -하, *-sulep* -스럽	0.97%
Roots (어근)	XR	Roots 어근	*ile* 이러, *pisus* 비슷	0.57%
Punctuation Marks (부호)	SF	. ? ! 마침표,물음표,느낌표		3.86%
	SP	, · : / 쉼표,가운뎃점,콜론,빗금		2.04%
	SS	‘ ’ “ ” [ ] ( ) { } – 따옴표,괄호표,줄표		(2.56%)
	SE	… 줄임표		(0.13%)
	SO	- ~ 붙임표(물결,숨김,빠짐)		(0.03%)
	SW	Miscellaneous signs (mathematics, currency) 기타기호 (논리수학기호, 화폐기호)		(0.16%)
Unanalysable Data (분석 불능)	NF	Nouns (assumed) 명사추정범주		0.00%
	NV	Predicates (assumed) 용언추정범주		0.00%
	NA	Unanalysable words 분석불능범주	isolated syll.	(0.01%)

	UNA	Unanalysable words 분석불능범주		(0.39%)
	UNC	Unanalysable words 분석불능범주		(0.40%)
	UNT	Unanalysable words 분석불능범주		(0.36%)
Not Hankul (한글 이외)	SL	Foreign loanwords 외국어	TV, NGO, LG	(0.32%)
	SH	Sinograms 한자	金 *kim*, 李 *i*	(0.29%)
	SN	Numbers 숫자	1, 2, 1980	(1,33%)

Table A.1: Tagset of the Sejong Corpus (written and spoken)

A.2.2 Grammar focus in Korean as a Foreign Language

As for any agglutinative language, the study of Korean language inevitably involves the study of grammatical morphemes. Sohn [2013, p.7] defines Korean morphemes as follows:

> "There are several hundreds of particles and affixes (especially suffixes) in Korean. With constant form and meaning, they agglutinate with each other in a fixed order and are attached to nominal or verbal stems to perform various syntactic and semantic functions."

It is therefore not surprising to find a full collection of particles and affixes in Korean grammars. Sohn follows the traditional classification of words in Korean and considers that particles (called *cosa* 조사) form an independent word class in Korean. He defines them as "postpositional function words which follow a nominal (including a nominalized clause), an adverbial (including an adverbial clause), or a sentence". The term affix is used for morphemes attached to verbs. However, Sohn acknowledges that this distinction has no ground from a morphosyntactic point of view: "[the] grammatical behaviour [of particles] is somewhat similar to that of verbal suffixes" [Sohn, 2013, p.229]. Indeed, inflection in Korean is materialised by both postpositional particles for nouns, and suffixes for verbs. In any case, we

may divide Korean morphemes into two groups: nominal morphemes (attached to nouns) and verbal morphemes (attached to verbs). Both nominal and verbal morphemes are ruled by strict topological rules and are combined in a particular order. This order is presented in Tables A.2 and A.3, borrowed from Chun [2013].

Table A.2 shows that four differents morphemes may be attached to a nominal stem in Korean: namely, the plural suffix *tul* 들, dative particle *eykey/hanthey* 에게/한테, a central morpheme (e.g. the morpheme *man* 만 which means "only") and a final morpheme (e.g. the accusative particle which shows allomorphic variation *ul/lul* 을/를). The word *haksayngtuleykeyman* 학생들에게만 ('only to students') can be segmented into four morphemes *haksayng-tul-eykey-man* 학생-들-에게-만.

Nominal stem	Plural	Dative	Central	Final
	tul	eykey	man	i/ka, (l)ul, (n)un
	들	에게	만	이/가, 을/를, 은/는

Table A.2: Topological structure of the nominal form in Korean

Verbal forms in Korean are also composed of a verbal stem to which different morphemes are attached. The example *ip.hisiesskeysseyo* 입히시었겠어요 (which roughly translates as 'would you have dressed (somebody)?') is segmented into seven morphemes in Figure A.3. The translative morpheme is not required but can only appear at the very ending of a verbal form.

Verbal stem	Verbal morphemes						
	Causative Passive	Honorific	Tense	Aspect	Modes	Honorific	Translative
ip	hi	si	ess	keyss	e	yo	
입	히	시	었	겠	어	요	

Table A.3: Topological structure of the verbal form in Korean

Teaching Korean grammar typically involves teaching how to use these grammatical morphemes, i.e., in which context and for what purpose, as well as how

to combine them. If we examine the table of contents of the *Korean Grammar in Use – Beginning to early Intermediate*[7], we observe that the contributors of the grammar chose to dedicate 20 units out of 24 to verbal endings and a whole unit to present 20 nominal particles.[8] We may divide the latter into two groups, according to their distribution: conjunctive verbal endings (which are used to coordinate two propositions) and final verbal endings (which may appear at the end of sentences).[9] The units concerning endings of *Korean Grammar in Use* may be categorised as follows:

- conjunctive verbal endings: listing and contrast, time expressions, reasons and causes, background information and explanations, conditions and suppositions, quotations;
- final verbal endings: ability and possibility, demands and obligations/permission and prohibition, expressions of hope, making requests and assisting, trying new things and experiences, asking opinions and making suggestions, intentions and plans, conjecture, expressions of state, discovery and surprise, "additional endings";
- morphemes attached to nouns: particles;
- morphemes attached to nouns and verbs: purpose and intentions, confirming information.

Mastering the use of different endings provides a wider range of tools in order to comprehend or to convey implicit (or *connotative*) information along with the literal (or *denotative*) meaning. Indeed, the three sentences in Example 26 have the same literal meaning. However, changing the verbal ending changes the level of politeness and conveys different emotions from the speaker. Example 26a is the most 'neutral' out of the three: the speaker simply expresses their ignorance about something, in a polite and formal way. Example 26b is similar to the previous

[7] Ahn Jean-Myung, Lee Kyung-Ah, Han Hoo-Young. *Korean Grammar in Use – Beginning to early Intermediate*, published in 2010 by Darakwon.

[8] The four remaining units respectively concern tenses (considered as *prefinal* and not final verbal endings), negative expressions, irregular conjugations and changes in parts-of-speech.

[9] This distinction is important in that most POS tagset for Korean do have different tags for conjunctive (`EC` in the Sejong Corpus tagset) and final verbal endings (`EF`), but also prefinal endings (`EP`, which is used for tense morphemes).

example except that the morpheme *-keyss-* -겠- was added. The explicit meaning is the same but this time, the speaker is less assertive and more gentle and polite. As for Example 26c, it still has the same explicit meaning but the level of formality dropped by one degree and the speaker expresses some kind of surprise to an unexpected situation.

(26) a. 저는 잘 모**릅니다**.
ce-nun cal molu-**p-ni-ta**
I-TOP well ignore-**AH-IND-DECL**
'I don't know (well).'

b. 저는 잘 모르**겠습니다**.
ce-nun cal molu-**keyss-sup-ni-ta**
I-TOP well ignore-**may-AH-IND-DECL**
'I don't know (well).'

c. 저는 잘 모르**는데요**.
ce-nun cal molu-**nun-tey-yo**
I-TOP well ignore-**MD-place-POL**
'I don't know (well).'

A.2.3 Table of Grammar Points

In order to study the syntactical structures of Korean language as taught as a foreign language, we transcribed all of the grammar points from textbooks of the first three years of study of Korean as a foreign language.

Table A.4 groups the grammar points seen in textbooks level 1 (1-2) and 2 (2-1 and 2-2) of the Yonsei series, and level 3 (3-1 and 3-2) of the Ewha series. Grammar points are ordered in rows alphabetically and not by level of difficulty assumed in textbooks, but the lesson they were extracted from is indicated in the "level" column.

The majority of these grammar points are endings or suffixes, as shown by the hyphen in front of most of the grammar points. The last ones are not affixes nor verbal endings so they are not attached to any stem, hence the crossed out cells in the "attached to" columns.

Each of the grammar points are analysed according to criteria that we defined

specifically for our research problem. The constitution of such a table was indeed done for the purpose of categorising grammar points taught in Korean as a foreign language with morphological, morphophonological and semantic criteria in order to reach a clearer and more objective view of the way they can potentially be processed by corpus exploration tools. As a matter of fact, we also used those criteria to choose the grammar points we would use in our experiments, i.e., to determine which grammar points are not easily 'concordanceable' and would therefore be interesting to retrieve using our system (see Section 6.2.2). However, it is noteworthy that this table was filled by the author alone, and might contain errors of judgement. Corrections, suggestions and discussions are welcomed.

The 10 columns are described as follows:

1. `ID`: an ID given to the grammar points of this table to identify them.

2. `Grammar Points`: the name of grammar points as they appear in the textbooks: most of the time, grammar points are represented by their constructions directly but sometimes they are named.

   ex: 170. 접속사 (conjunctions)
   This category contains the conjunctions constructed with the verb *kulehta* 그렇다: *kulayto* 그래도 'though', *kulayse* 그래서 'so that', *kulena* 그러나 'but', *kulenikka* 그러니까 'therefore', *kulentey* 그런데 'however', *kulehciman* 그렇지만 'however', *kuliko* 그리고 'and'.

3. `Allomorphy`: this column is filled if the grammar point displays contextual allomorphy, i.e., it has different forms depending on the context and this alternation is *distributional* in that one allomorph does not appear in the same context as the other. This allomorphy is either due to the syllabic structure of the preceding syllable (whether it ends with a vowel (CV) or with a coda (CVC)) or the vowel harmony.

   ex: 63. -(이)나 (suggested choice)

영화	+	-(이)나	=	영화나
yenghwa	+	-(i)na	=	yenghwana
film	+	or	=	'film or'

책	+	-(이)나	=	책이나
cheyk	+	-(i)na	=	cheykina
book	+	or	=	'book or'

4. `Morphological Variation`: this column is filled if the attachment of the grammar point potentially entails morphological variations of the stem, or the grammar point potentially contains a morpheme subject to morphological variation (for instance, a verb). Since an infrasyllabic morpheme is integrated to another syllable, a grammar point composed of an infrasyllabic morpheme automatically entails morphological variations.

   ex: 38. -(으)ㄹ 지도 모르다 (uncertainty)

가	+	-(으)ㄹ 지도 모르다	=	갈 지도 모른다/몰라/모릅니다.
ka	+	-l cito **moluta**	=	***kal** cito **mo.lun.ta/mol.la/mo.lub.ni.ta***
go	+	[uncertainty]	=	'I do not know if I will go'

5. `Infrasyllabic`: this column is filled if the grammar point is composed of at least one 'infrasyllabic' morpheme integrated in the preceding syllable.

6. `Morphological Ambiguity`: filled if the grammar point has at least one homograph, either a homographic morpheme, or a homograph resulting from a fortuitious combinaison of morphemes which incidentally happens to be similar to the grammar point and therefore causes a morphological ambiguity.

   ex: 4. -(으)ㄴ, -는, -(으)ㄹ (adnominal ending)

   Concordancing using `-ㄴ` or `ㄹ` as queries would not work because these morphemes are integrated to the preceding syllable; using either 은, 는 or 을 could retrieve words containing these syllables, but not as adnominal endings: attached to nouns, *-un* -은 /*-nun* -는 are topic markers, and *ul* is the object marker, or is simply part of words such as *kaul* 가**을** 'autumn' or part of other constructions such as (34) -ㄹ/**을**까 봐(서).

7. `Polysemy`: this column is filled if the grammar point has different senses or usages.

   ex: 56. -(으)로 (direction) / 57. -(으)로 (change of state, exchange, transfer) These two usages, as well as other usages, are described at A.2.4.

   ex: 64. -(이)나 (suggested choice) / 65. -(이)나 (unexpected amount)

음악회에나	가	봅시다.		[KGIL_165]
umakhoe-ey-na	ka	po-p-si-ta.		
concert-to-or	go	try-AH-RQ-PR		

'Let's go to a concert or something.'

저는	어제	열	시간이나	잤어요.	[KGIL_165]
ce-nun	ecey	yel	sikan-ina	ca-ss-eyo.	
me-TOP	yesterday	10	hour-as.much	sleep-PST-POL	

'Yesterday I slept (as many as) 10 hours.'

8. `'Concordanceable'`: it is possible to retrieve the grammar point with high precision and recall using a single and *simple* query (not a regular expression) in a concordancer, i.e., the concordance lines only contain the target grammar point (and not homographs) in all or most of its forms and usages. Considering this definition of 'concordanceable', we note that morphological variations, morphological ambiguity and polysemy all imply that the grammar point is not 'concordanceable'.

9. `Attached to`: class of the stem the grammar points are attached to, if any. Expected classes are `A` (predicative adjective), `V` (verb) or `N` (noun). In some cases, a grammar point introduces the construction's different adaptations to verbs, adjectives and nouns in the same lesson.

10. `Level`: lesson and textbook series in which grammar points appear. This column may have two items if the grammar point appears in both Ewha and Yonsei textbooks, with the same form and sense. This contains the following code: textbook series `E/Y` (`E` for Ewha, `Y` for Yonsei), textbook level and lesson number.

    ex: 1. -(ㄴ/는)다면, -(이)라면

    `E3-2_10` = Ewha's level 3-2 textbook, lesson 10.

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
1	-(ㄴ/는)다면		*	*		*		A,V,N	E3-2_10

A.2. Korean Grammar

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
	-(이)라면								
2	-(ㄴ/는)다면서요 N+라면서요		*	*			*	A,V,N	E3-1_6
3	-(는)군요	*					*	A,V	Y1-2_9
4	(으)ㄴ, 는, (으)ㄹ	*	*	*	*	*		V	Y1-2_6
5	-(스/ㅂ)니다만		*	*			*	A,V	Y2-2_6
6	-(으)ㄴ 적이 있다	*	*	*			*	A,V	Y2-1_2
7	-(으)ㄴ 지	*		*				A	Y2-1_1
8	-(으)ㄴ 채로		*	*			*	A,V	E3-2_14
9	-(으)ㄴ 후에	*	*	*				V	Y1-2_9
10	-(으)ㄴ/는 김에		*	*			*	A,V	E3-2_13
11	-(으)ㄴ/는 데다가 N+에다가		*	*			*	A,V,N	E3-2_12
12	-(으)ㄴ/는 반면(에)		*	*			*	A,V	E3-1_5
13	-(으)ㄴ/는 줄 알다/모르다	*	*	*				A,V	E3-1_3
14	-(으)ㄴ/는 줄 알다	*	*	*				A,V	Y2-2_10
15	-(으)ㄴ/는 척하다		*	*				A,V	E3-2_14
16	-(으)ㄴ/는 편이다 N+인 편이다		*	*			*	A,V,N	E3-2_10
17	-(으)ㄴ/는다, -니 ? (반말)	*	*	*	*		*	A,V	Y2-1_4
18	-(으)ㄴ/는데다가		*	*			*	A,V	Y2-2_8
19	-(으)ㄴ/는데도		*	*			*	A,V	E3-2_9
20	-(으)ㄴ/는지 알다/모르다	*	*	*			*	A,V	Y2-1_5
21	-(으)ㄴ/는데	*	*	*	*			A,V	Y1-2_9
22	-(으)ㄴ/는데				*		*	V	Y2-1_1
23	-(으)ㄴ/는데							V	Y2-1_2
24	-(으)ㄴ/는데				*			V	Y2-1_3
25	-(으)ㄴ/는데요	*	*	*	*			A,V	Y1-2_8
26	-(으)ㄹ 거예요	*	*	*				V	Y1-2_8
27	-(으)ㄹ 것 같다	*	*	*				A,V	Y1-2_9
28	-(으)ㄹ 때	*	*	*				A,V	Y1-2_10
29	-(으)ㄹ 때마다		*	*			*	A,V	E3-1_3
30	-(으)ㄹ 만하다		*	*		*		V	E3-1_5
31	-(으)ㄹ 뻔하다		*	*			*	A,V	E3-2_14

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
32	-(으)ㄹ 뿐(만) 아니라		*	*			*	A,V,N	E3-1_3
33	N+뿐(만) 아니라								
34	-(으)ㄹ 수 있다	*	*	*				V	Y1-2_9
35	-(으)ㄹ 수도 있다		*	*			*	A,V	E3-1_1
36	-(으)ㄹ 수밖에 없다		*	*			*	A,V	E3-2_13
37	-(으)ㄹ지 -(으)ㄹ지			*				A,V	Y2-1_4
38	**-(으)ㄹ지도 모르다**		*	*				A,V	E3-2_10
39	-(으)ㄹ 테니까	*		*			*	V	Y2-2_10 E3-2_14
40	-(으)ㄹ 텐데		*	*			*	A,V	E3-1_7
41	-(으)ㄹ걸요		*	*			*	A,V	E3-1_5
42	-(으)ㄹ게요	*	*	*				V	Y1-2_8
43	-(으)ㄹ까 봐(서)		*	*			*	A,V	E3-1_1
44	-(으)ㄹ까 하다	*	*	*			*	V	Y2-1_3
45	-(으)ㄹ래요?	*	*	*				V	Y2-2_7
46	-(으)ㄹ지 모르겠다	*	*	*			*	A,V	Y2-2_6
47	-(으)니까	*		*	*		*	A,V	Y2-1_3 E3-2_11
48	-(으)니까	*				*	*	V	Y1-2_7
49	-(으)냐고 하다		*	*			*	A,V	E3-1_6
50	-(으)라고 하다 (간접인용)	*	*				*	V	Y2-2_7
51	-(으)러 가다	*	*					V	Y1-2_6
52	-(으)려고	*					*	V	Y2-1_1
53	-(으)려다가						*	V	E3-2_15
54	-(으)려던 참이다			*			*	V	E3-2_13
55	-(으)려면						*	V	Y2-1_5
56	**-(으)로**	*			*	*	*	N	Y2-1_5
57	**-(으)로**	*			*	*	*	N	Y1-2_7
58	-(으)로 하다	*	*				*	N	Y2-1_3
59	-(으)면	*	*				*	V	Y1-2_8
60	-(으)면 -(으)ㄹ수록	*	*	*			*	A,V	Y2-2_10
61	-(으)면 안 되다	*	*				*	A,V	Y2-1_2
62	-(으)면서	*	*				*	A,V	Y2-2_10
63	-(이)나	*	*		*		*	N	Y1-2_8

A.2. Korean Grammar

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
64	-(이)나	*	*		*	*		N	Y2-1_4
65	-(이)나	*	*		*	*		N	Y2-1_5
66	-(이)든지	*					*	PRO	E3-1_4 Y2-1_1
67	-(이)라고 하다 -(ㄴ/는)다고 하다		*	*			*	A,V,N	E3-1_6
68	-(이)라도						*	N	E3-1_7
69	-(이)래요, -(ㄴ/는)대요, -네요		*	*			*	A,V,N	E3-2_8
70	-ㄴ/은	*	*	*	*	*		A	Y1-2_6
71	-달라고 하다 (간접인용)		*				*	V	Y2-2_7
72	-거나						*	V	Y2-2_6
73	-거리다		*	*				N	E3-2_8
74	-게				*		*	A,V	Y2-1_2 E3-1_1
75	-게 되다		*				*	V	Y2-2_7
76	-게 하다		*			*	*	A,V	E3-1_4
77	-겠-						*	V	Y1-2_9
78	-겠군요						*	A,V	Y2-1_1
79	-고 있다		*				*	V	Y1-2_9
80	-곤 하다		*	*		*		V	E3-2_15
81	-과/와	*			*	*	*	N	Y1-2_6
82	-기 때문에						*	A,V	Y2-1_1
83	-기 위해서, -(으)ㄹ 해서		*	*			* *	V,N	Y2-2_8
84	-기 전에						*	V	Y1-2_10
85	-기는 하지만						*	A,V	E3-1_2 Y2-1_3
86	-기로 하다		*				*	A,V	Y2-1_4
87	-기에						*	A,V	E3-2_8
88	-나 보다, -(으)ㄴ가 보다		*	*				A,V	Y2-2_8 E3-2_9
89	-나요?					*	*	V	Y2-1_5
90	-느라고						*	V	E3-2_13
91	-는 것보다 -는 게 낫다		*					V	E3-1_2

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
92	-는다고 하다, -(이)라고 하다, -냐고 하다 (간접인용)	*	*	*			*	A,V,N	Y2-2_7
93	-는 대로						*	V	Y2-2_7
94	-는 동안						*	V	Y2-2_6
95	-는 바람에						*	V	E3-1_7
96	-다 보니까						*	V	E3-2_12
97	-다 보면						*	V	E3-2_15
98	-다가						*	V	Y2-1_5
99	-다니						*	V	E3-2_8
100	-답다		*					N	E3-1_1
101	-더군요						*	A,V	Y2-2_9
102	-던					*	*	A,V	Y2-2_9
103	-던데요						*	A,V	E3-1_2
104	-되다		*					N	E3-2_9
105	-만					*	*	N	Y1-2_8
106	-만에						*	N	Y2-2_9
107	-만큼						*	N	Y2-2_9
108	-밖에						*	N	Y2-2_9
109	-받다		*					N	E3-2_11
110	-보다							N	Y1-2_9
111	-부터						*	N	Y2-1_2
112	-스럽다		*					N	E3-1_7
113	-아/어, 이야 (반말)	*	*	*	*			A,V,N	Y2-1_4
114	-아/어 가지고	*	*	*			*	V	Y2-1_4
115	-아/어 버리다	*	*	*				A,V	E3-1_7
116	-아/어 보이다	*	*	*				A	E3-1_2 Y2-1_5
117	-아/어 오다	*	*	*				V	E3-2_15
118	-아/어 있다	*	*	*		*		A,V	Y2-2_6 E3-2_11
119	-아/어 주다	*	*	*				V	Y1-2_6
120	-아/어 지다	*	*	*	*			V	Y2-2_10
121	**-아/어도**	*	*	*	*			A,V	Y2-1_3 E3-1_4

A.2. Korean Grammar

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
122	-아/어도 되다	*	*	*				A,V	Y2-1_2
123	-아/어라	*	*	*				V	Y2-1_4
124	-아/어보다	*	*	*				A,V	Y2-1_2
125	-아/어서	*	*	*	*			V	Y1-2_7
126	-아/어야	*	*	*	*			V	E3-1_1
127	-아/어야 겠다	*	*	*			*	A,V	Y2-2_7
128	-아/어야 하다	*	*	*				A,V	Y2-1_2
129	-아/어지다	*	*	*				A	Y2-1_1 E3-2_9
130	-아/어하다	*	*	*				A	Y2-1_1 E3-1_4
131	-았/었다가	*	*	*			*	V	Y2-1_5
132	-았/었더라면	*	*	*				A	E3-2_10
133	-았/었다가	*	*	*			*	V	Y2-1_5
134	-았/었으면 좋겠다	*	*	*				A,V	E3-2_12 Y2-1_3
135	-아/어 보다	*	*	*	*			V	Y2-1_3
136	-아/어 놓다	*	*	*				V	E3-2_11
137	-아/어야지요	*	*	*			*	V	E3-2_9
138	-아/어/여서 그런지	*	*	*			*	A,V	Y2-2_9
139	-에 쯤						*	N	Y1-2_10
140	-에 대해서						*	N	Y2-2_6
141	-에 비해서						*	N	Y2-2_6
142	-에게						*	N	Y1-2_6
143	-에게서						*	N	Y1-2_8
144	-에다가						*	N	Y2-2_8
145	-에서 -까지						*	N	Y1-2_7
146	-의							N	Y2-2_9
147	-자 (반말)					*	*	V	Y2-1_4
148	-자고 하다 (간접인용)		*					V	Y2-2_7
149	-자고 하다 -(으)라고 하다		*	*			*	A,V	E3-1_6
150	-쟁이						*	N	E3-2_12
151	-쟤요, -(으)래요						*	V	E3-2_8

ID	Grammar Points	Allomorphy	Morph. variations	Infrasyllabic	Morph. Ambiguity	Polysemy	Concordanceable	Attached to	Level
152	-적						*	N	E3-1_5
153	-중에서 제일						*	N	Y1-2_10
154	-지 그래요?						*	V	E3-1_3
155	-지 마 (반말)						*	A,V	Y2-1_4
156	-지 말고						*	V	Y2-2_8
157	-지 말다		*				*	V	Y1-2_7
158	-지 못하다	*	*				*	A,V	Y1-2_10
159	-지 않으면 안 되다		*				*	A,V	Y2-2_6
160	-지만						*	V	Y1-2_6
161	-처럼						*	N	Y2-2_10
162	N 때						*	N	Y1-2_10
163	덕분에						*	N	Y2-2_10
164	동안						*	N	Y1-2_10
165	못						*	/	Y1-2_10
166	만큼						*	N	E3-1_5
167	보고					*	*	/	E3-2_12
168	아무						*	/	Y2-2_8
169	얼마나 -(으)ㄴ/는지 모르다	*	*	*			*	A,V	Y2-2_8
170	접속사							/	Y2-2_10
171	덧-						*	N	E3-1_6
172	맨-					*	*	N	E3-2_14
173	헛-						*	N	E3-2_10
174	단위 명사							/	Y1-2_6
175	ㄷ 동사		*					/	Y1-2_7
176	ㅅ 동사		*					/	Y2-2_8
177	ㅎ 동사		*					/	Y1-2_6
178	르 동사		*					/	Y1-2_7
179	사동사							/	E3-1_4
180	피동사							/	E3-2_11

Table A.4: Characteristics of grammar points extracted from Ewha and Yonsei textbooks

A.2.4 Example of a Polysemous Morpheme: -(으)로 *-(u)lo*

Always attached to nouns, *-(u)lo* -(으)로 is one of the most used case particles in Korean, with more than 320,000 occurrences in the Sejong corpus. This particle has two allomorphs: the form *-ulo* -으로 used with stems ending with a consonant is the 28th most frequent morpheme in Sejong, and the form *-lo* -로 used with stems ending with a vowel is the 30th. As a matter of fact, this particle is one of the few associated with more than one grammatical case, along with *-eyse* -에서/*-eykeyse* -에게서/*-hantheyse* -한테서 (which is used to express both source and dynamic locative) or *-hako* -하고 (for both comitative and conjunctive functions).

Each meaning of the particle can be inferred from the context. The interpretation is therefore unambiguous and relies on co-occurring words, especially the verb. In this section, we describe the main usages of *-(u)lo* -(으)로 and try to define to which extent the context helps interpret the particle.

Main usage: adverbial suffix Due to its various usages, *-(u)lo* -(으)로 is often seen as a common adverbial suffix.

(27)

어	제가	정말**로**	사투리를	빨리	고치드라고요.
e	cey-ka	cengmal-**lo**	sathuri-lul	ppalli	kochi-tu-la-ko-yo.
yeah	I-NM	truth-**ADV**	accent-OBJ	quickly	cover-RT-DC-QT-POL
[6CT_0024]					

'Umm I had really hidden my accent quickly.'

In this example, *-lo* -로 is attached to the noun *cengmal* 정말 to form the adverbial use 'for real' or 'really'. We can also note that *cengmal* 정말 actually has also a non-ambiguous adverbial use in its basic form (the same sentence with *cengmal* 정말 is correct) but is still very often used with *-lo* -로.

However, sorting the various context uses of *-(u)lo* -(으)로 brings to light specific usages, among which the directional, the instrumental and the essive functions, all of which are illustrated below.

Directional function In most cases, when used with motion verbs, *-(u)lo* -(으)로 has the meaning of "toward", "in the direction of" (allative use); but it might

also denote the source of something, especially when used as a compound with ablative particles *-ese* -에서 and *-pwuthe* -부터. In the latter case, it would be then simply translated by "from".

(28) 침실로 간다. [BTEO0324]
chimsil-lo ka-n-ta.
bedroom-towards go-IND-DECL
'Going to/towards the bedroom.'[10]

Change of state, exchange, substitution With verbs denoting a change, *-(u)lo* -(으)로 is always attached to either the manner or the end-point of the process of change, thus in the meaning of "into" or "by".

(29) 천육백 원으로 구월 일일부터 올랐지. [5CT_0013]
chenyukpayk won-**ulo** kwuwel ilil-pwuthe oll-ass-ci.
1600 won-into nine.month one.day-from increase-PST-SUP
'(The price) increased to one thousand six hundred wons starting from the first of September.'

Instrumental function *-(u)lo* -(으)로 is also frequently used as an instrumental particle, denoting either something tangible such as a means, a content or a material, or something more abstract such as a consistency. In both cases we may roughly translate the particle by "*with*". Incidentally, *-(u)lo* -(으)로 is interchangeable with *-(u)losse* -(으)로써, another particle but which is attached exclusively to an instrument. Both appear in similar contexts with that meaning.

Means This usage of *-(u)lo* -(으)로 is very close to the instrumatal function, except that the particle is specifically attached to words denoting means.

(30) 전화하시는 거보다 이메일**로** 하시는 게 더
cenhwaha-si-nun ke-pota imeyil-**lo** ha-si-nun key te
phone-SH-TOP that-than e-mail-SH-**INS** do-SH-TOP that-NM more
편하실 수도 있거든요? [5CT_0047]
phyenha-si-l swu-to iss-ketun-yo?
comfortable-SH-PRS way-also exist-indeed-POL
'Wouldn't it be easier if you send an email instead of making a phone call?'

[10] 'Going to/towards the bedroom' is the literal meaning, but this sentence can also be used to say that one goes to sleep.

Material Similar to the previous usage, this one is very close to the instrumental function, except that in this case, the particle is specifically attached to words referring to materials.

(31) 플라스틱**으로** 만든 샤베트기는 수입품이
phullasuthik-**ulo** mantu-n syapeythuki-nun suipphwum-i
plastic-**INS** make-MD popsicle machine-TOP imports-NM
대부분이다. [BTAA000]
taypwupwun-i-ta.
mainly-be-DECL
'Popsicle machines made of plastic are mostly imported products.'

Frequentative The instrumental particle may be used as a frequentative when attached to time words such as *sikan* 시간 ("hour"), *nal* 날 ("day"), *pam* 밤 ("night") etc. This particular usage of *-(u)lo* -(으)로 might therefore need semantic annotations to be identified automatically.

(32) 배달물은 날**로** 늘어만 간다. [BTAA0005]
paytalmul-un nal-**lo** nul-e-man ka-n-ta.
delivery-TOP day-FQ increase-INF-only go-IND-DECL
'Deliveries keep increasing day by day.'

Essive function The last main usages we identified is the use of *-(u)lo* -(으)로 as an essive particle, denoting status, capacity, position or qualifications when attached to a word referring to a human being ("*as, in the capacity of*"). We mentioned that when used as an instrumental particle *-(u)lo* -(으)로 could be replaced by *-(u)losse* -(으)로써. Likewise, in the essive function, it is interchangeable with *-(u)lose* -(으)로서.

(33) 그때 같은 학교에 교환 학생**으로** 갔었던
kudday kathun hakkyo-ey kyohwan haksayng-**ulo** ka-ss-ess-den
then same school-LOC exchange student-AS go-PST-PST-PST
연대 후배가 있었어요. [6CT_0012]
yenday hupay-ka iss-ess-e-yo.
Yonsei hoobae-NM exist-PST-DECL-POL
'Back then there was a hoobae[11] from Yonsei University who went to the same school as an exchange student.'

[11]In Korean culture, *hoobae* is a word used to refer to people who are have less years of

Among the different usages of the particle *-(u)lo* -(으)로, we note that most of them can be automatically identified by a co-occurring word, such as a verb (motion verbs for the directional function, and verbs denoting changes for the change of state function) or nouns (time for the frequentative function, instruments/means/materials for their respective functions). It might therefore be more interesting to keep lexical words when trying to disambiguate the usages of a morpheme such as *-(u)lo* -(으)로. Using either the predicate of the sentence, or the verb to which *-(u)lo* -(으)로 is attached to, could help our system to group sentences illustrating those usages together: for instance, all sentences containing both *-(u)lo* -(으)로 and *kata* 가다 'go' would illustrate the directional function of the particle. However, in order to retrieve sentences where *-(u)lo* -(으)로 is used as a directional particle but not necessarily with the verb 'go', semantic annotations would be needed.

experience or service at work, at school or in any institution. A *hoobae* can therefore be older as long as he or she arrived after. It is close to the notion of *junior* in English and equivalent to the notion of *kouhai* 後輩 in Japanese.

Appendix B

Scripts

This Appendix shows the scripts that we wrote in order to test the requirement specification of our original corpus exploration function.

B.1 Similarity Measure

This similarity measure script is the core of the experiments conducted in Chapter 6, and, as a matter of fact, calls the second script shown in B.2 in order to compute minimum edit distance.

Apart from the edit distance function which is called and not defined in this script, and the Jaccard similarity measure which was called from the `sklearn` library, all other functions are defined either at the beginning of this script or within the main function. These functions allow to test the various configurations described in 6.3: the variable `raw_query` is modified to take one or several sentences as queries, the similarity measures are called respectively in the functions `jaccard_similarity`, `dice_coefficient` and `edit_distance`, the function `remove_lexical_item` is used to remove lexical items from both the input and the corpus, the function `to_bigram` is used to take word order into consideration when computing the Jaccard distance and the genre is defined by the user when they specify the corpus name as an argument to the programme.

This script takes three arguments as input: (1) the name of the corpus (the name of the sample from Sejong, or bnc for the English adaptation), (2) the name of the target grammar point (either *eto* 어도, *ulo* 으로 or *ljitomoluta* ㄹ지도모르

다, for more details, see 6.2.2)[1], as well as (3) the number of the desired search mode (1 for the default mode, 2 for the distributional analysis search mode and 3 for the different usages search mode, all of which are thoroughly described in 5.3.4).

```
import os, sys, re, operator
import distance
import numpy as np
from collections import OrderedDict
from scipy import stats
from sklearn.metrics import jaccard_similarity_score
from edit_distance import *

reload(sys)
sys.setdefaultencoding('utf8')

def to_sejong_tagset(x):
  """ uses a preconfigured table to transform the kkma tagset into
      the Sejong tagset before computing any measure:
    tags all become less precise (from a subcategory to a more
        general category)
  """
  with open(dirname+'/scripts/kkmatosejong') as kts:
    for line in kts:
      tags = line.split('\t')
      x = re.sub(tags[0],tags[1].rstrip(),x)
  return x

def common_elements(list1, list2):
  return [element for element in list1 if element in list2]

def get_pos(x):
  found = ''
  for item in x:
    try:
      found += re.search('/(.+)', item).group(1)
    except AttributeError:
    # if regex not found in the original string
      found = ''
  return found

def ranking(nb,dic,mode):
  """ gives the ranking of the top 10 most similar (closest to 0)
      sentences to queries
    with or without the query itself if it is part of the corpus
```

[1]Another grammar point also appears in the script, *unikka*, but this grammar point is not studied in this dissertation.

```
  """
  if mode == 3:
    sorted_dic = sorted(dic.items(), key=operator.itemgetter(1),
        reverse=True) # option 3
  else:
    dic = {k: v for k, v in dic.iteritems() if v != 1.0}
    sorted_dic = sorted(dic.items(), key=operator.itemgetter(1)) #
        options 1 and 2

  res = ''
  sorted_list = []
  extended_list = ''

  if len(sorted_dic) < 51 :
    limit = len(sorted_dic)
  else:
    limit = 50

  for i in range(0,limit): # change to nb,nb+10 to leave the similar
      sentence out of this ranking; or 0,nb+10 to keep it
    sorted_list = list(sorted_dic[i])
    if i < 10:
      res += str(i+1)+'. '+sorted_list[0]+' ('+str(sorted_list[1])+')
          \n'
    extended_list += str(i+1)+'. '+sorted_list[0]+'\n'

  with open(dirname+'/'+outfile+'.ext','w') as ext_list:
    ext_list.write(extended_list)

  return res

def remove_lexical_item(x):
  """ removes word form that are not relevant for syntactic (
      typically, lexical items) """
  for i in range(0,len(x)):
    if re.search(r'(NNG|NNP|NR|NP|VV|VA|MM)',x[i]) is not None:
      x[i] = re.sub(r'.+?/([A-Z]+)',r'\1',x[i])
  return x

def get_hmeans(x,y):
  """ gets harmonic means between similarity measures (by pairs) """
  if not x:
    """ Initialisation of jaccard dictionary used to compare results
        from different queries """
    x = y
  else:
    for key in y.keys():
      if not x.has_key(key):  print key,'is not in the main
          dictionary...'
```

```
80        elif y[key] == 0.0: x[key] = y[key]
          elif x[key] != 0.0: x[key] = stats.hmean([x[key], y[key]])
    return x

def to_bigram(a, b):
    """ a more orthodox and robust implementation from :
      https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/
         Dice's_coefficient#Python
      dice coefficient 2nt/na + nb.
    """
    if not len(a) or not len(b): return 0.0
    if len(a) == 1: a=a+u'.'
    if len(b) == 1: b=b+u'.'

    a_bigram_list=[]
    for i in range(len(a)-1):
      a_bigram_list.append(a[i]+' '+a[i+1])
    b_bigram_list=[]
    for i in range(len(b)-1):
      b_bigram_list.append(b[i]+' '+b[i+1])

    a_bigrams = set(a_bigram_list)
    b_bigrams = set(b_bigram_list)

    return (a_bigrams, b_bigrams)

def dice_coefficient(x):
    (a_bigrams, b_bigrams) = x

    overlap = len(a_bigrams & b_bigrams)
    dice_coeff = overlap * 2.0/(len(a_bigrams) + len(b_bigrams))
    return 1-dice_coeff # inverse of dice coefficient in order to
       compare with Jaccard and Levenshtein

def jaccard_similarity(a,b):
    return distance.jaccard(a,b)

if __name__ == '__main__':
    content = []

    res = ''
    dirname = os.getcwd()

    corpus = sys.argv[1]
    outfile = sys.argv[2]
    mode = int(sys.argv[3])

    split_map = {1:'1 - same words in similar contexts',
         2:'2 - similar words (based on POS) in similar contexts',
```

```
      3:'3 - same words in different contexts'
    }

if mode not in [1,2,3]:
  raise ValueError(u'Wrong input for mode: should be either 1, 2 or
      3\n')

# Get corpus
with open(dirname+'/'+corpus) as i:
  for line in i:
    if not line.strip(): continue
    content.append(line.strip().decode('utf-8').split(' '))

o = open(dirname+'/'+outfile, 'w')

print 'Successfully fetched parsed sentences from '+corpus+' !\n'

# Get input sentence(s) and target grammatical construction
if "어도" in outfile:
  # 어도 from Ewha 3-1
  raw_query = ['저/NP 는/JX 피곤/NNG 하/XSV 어도/ECD 아침/NNG 운동/NNG
      은/JX 꼭/MAG 하/VV 어요/EFN ./SF','아무리/MAG 바쁘/VA 아도/ECD
      아침/NNG 식사/NNG 는/JX 꼭/MAG 하/VV 세요/EFN ./SF','문제/NNG
      가/JKS 어렵/VA 어도/ECD 끝/NNG 까지/JX 푸/VV ㄹ/ETD 거/NNB 이/VCP
      에요/EFN ./SF']
  raw_target = ["어도/ECD"]
elif "으니까" in outfile:
  # 으()니까 from Yonsei 1-2
  raw_query = ['오늘/NNG 을/JKO 일/NNG 이/JKS 많/VA 으니까/ECD 내일/NNG
      만나/VV ㅂ시다/EFA ./SF','날씨/NNG 가/JKS 춥/VA 니까/ECD 안/NNG
      으로/JKM 들어가/VV 세요/EFN ./SF','담배/NNG 는/JX 건강/NNG 에/JKM
      나쁘/VV 니까/ECD 피우/VV 지/ECD 말/VXV 시/EPH ㅂ시오/EFN ./SF','
      오늘/NNG 은/JX 눈/NNG 이/JKS 많이/MAG 오/VV 니까/ECD 자동차/NNG
      를/JKO 운전/NNG 하/XSV 지/ECD 마/VV 세요/EFN ./SF']
  raw_target = ["니까/ECD"]
elif "으로" in outfile:
  # 으()로 from Yonsei 1-2
  raw_query = ['젓가락/NNG 으로/JKM 먹/VV 습니다/EFN ./SF','한국말/NNG
      로/JKM 말하/VV 시/EPH ㅂ시오/EFN ./SF','버스/NNG 로/JKM 오/VV
      았/EPT 습니다/EFN ./SF','연필/NNG 로/JKM 쓰/VV ㅂ니다/EFN ./SF','
      김치/NNG 는/JX 배추/NNG 로/JKM 만들/VV ㅂ니다/EFN ./SF']
  raw_target = ["로/JKM"]
elif "ㄹ지도모르다" in outfile:
  # ㄹ지도모르다  from Ewha 3-2
  raw_query = ['50/NR 년/NNM 후/NNG 에/JKM 는/JX 사람/NNG 대신/NNP
      에/JKM 로봇/NNG 이/JKS 일/NNG 을/JKO 하/VV ㄹ/ETD 지도/NNG
      몰/VV ㄹ라요/EFN ./SF','정말/MAG 그러/VV ㄹ지/ECD 도/JX 모르/VV
      아요/EFN ./SF','화성/NNG 에/JKM 외계인/NNG 이/JKS 살/VV ㄹ지/ECD
      도/JX 모르/VV 아요/EFN ./SF','내일/NNG 은/JX 맑/VA 을지/ECS 도/JX
      모르/VV ㅂ니다/EFN ./SF','그/MDT 사람/NNG 말/NNG 이/JKS 사실/NNG
      일지/NNG 도/JX 모르/VV 아/ECS ./SF']
  raw_target = ["ㄹ지/ECD","도/JX","모르/VV"]
```

```
  else:
    raise ValueError('Wrong argument name, should be either 어도,
        으니까, 으로 or ㄹ지도모르다')

  print 'Current query is :'
  query = []
  target = []

  o.write('Automatically generated file!\n')
  o.write('\nData:\n')

  # Convert tagset from KKMA to Sejong for the input sentence(s) and
      the target construction
  for sentence in raw_query:
    if 'sejong' in corpus:
      sentence = to_sejong_tagset(sentence)
    query.append(sentence.decode('utf-8').split(' '))
    o.write(sentence+"\n")
    print sentence

  for form in raw_target:
    if 'sejong' in corpus:
      form = to_sejong_tagset(form)
    target.append(form)
  print 'Target:',target
  o.write('\t(b) Target: '+str(target)+'\n')

  o.write('\t(c) Mode: '+split_map[mode]+'\n')

  print '\nSimilarity measure in process...\n'

  biHmeans = {}
  uniHmeans = {}
  levHmeans = {}
  biNLHmeans = {}
  uniNLHmeans = {}
  levNLHmeans = {}

  # For each sentence used as input...
  for sample in query:
    count = 0

    bigram = {}
    unigram = {}
    levenshtein = {}
    biNL = {}
    uniNL = {}
    levenshteinNL = {}
```

```
    # For each sentence in the corpus...
    for x in content:
      common = common_elements(x, target)

      if mode == 2:
      # use this to get different words but same context (option 2)
      # abort if the sequence of POS does not match or if the
          sequence of word is the same
        if not get_pos(target) in get_pos(x) or ''.join(target) in ''
            .join(common): continue


      else:
      # use this to get same words (options 1 and 3)
        if not ''.join(target) in ''.join(common): continue

      count += 1
      # Computes the similarity between the query and each sentence
          of the corpus
      unigram[' '.join(x)] = jaccard_similarity(sample,x)
      levenshtein[' '.join(x)] = edit_distance(sample,x)
      bigram[' '.join(x)] = dice_coefficient(to_bigram(sample,x))

      # Same but with only POS of lexical items in both query and
          corpus
      clr_x = x[:]
      clr_sample = sample[:]

      uniNL[' '.join(x)] = jaccard_similarity(remove_lexical_item(
          clr_sample),remove_lexical_item(clr_x))
      levenshteinNL[' '.join(x)] = edit_distance(remove_lexical_item(
          clr_sample),remove_lexical_item(clr_x))
      biNL[' '.join(x)] = dice_coefficient(to_bigram(
          remove_lexical_item(clr_sample),remove_lexical_item(clr_x)))

    biHmeans = get_hmeans(biHmeans,bigram)
    biNLHmeans = get_hmeans(biNLHmeans,biNL)
    uniHmeans = get_hmeans(uniHmeans,unigram)
    uniNLHmeans = get_hmeans(uniNLHmeans,uniNL)
    levHmeans = get_hmeans(levHmeans,levenshtein)
    levNLHmeans = get_hmeans(levNLHmeans,levenshteinNL)

  if count == 0: sys.exit('No sentence matched!')

  # Print file
  print count,'sentences matched',''.join(target)

  o.write('\nNumber of matched sentences: '+str(count))
  o.write('\n\n10 most similar sentences according to the Jaccard/
```

```
    Dice distance using bigrams (measure is Jaccard\'s):\n')
o.write('\n\tWord forms + POS\n'+ranking(len(query),biHmeans,mode))
o.write('\n\tWord forms (except lexical items) + POS\n'+ranking(len
    (query),biNLHmeans,mode))

o.write('\n10 most similar sentences according to the Jaccard/Dice
    distance using unigrams only:\n')
o.write('\n\tWord forms + POS\n'+ranking(len(query),uniHmeans,mode)
    )
o.write('\n\tWord forms (except lexical items) + POS\n'+ranking(len
    (query),uniNLHmeans,mode))

o.write('\n10 closest sentences using the Levenshtein distance:\n')
o.write('\n\tWord forms + POS\n'+ranking(len(query),levHmeans,mode)
    )
o.write('\n\tWord forms (except lexical items) + POS\n'+ranking(len
    (query),levNLHmeans,mode))

o.close()
```

B.2 Edit Distance

The following script is the programme used to compute minimum edit distance for both experiments on Korean (first function `edit_distance`), and their adaptation to the English language (function `edit_distance_en`). Those functions are directly called in the similarity measure script presented in Section B.1.

The weight were adjusted to Korean data (Sejong tagset) and English data (CLAWS5 tagset) from a simple implementation in Python of the minimum edit distance provided by Isabelle Tellier. The details of the adapted weighting are given in Section 6.3.3 for Korean and Section 6.4 for English.

```
import re

def edit_distance(mot1,mot2):

    table = [[0 for j in range(len(mot2)+1)] for i in range(len(mot1)
        +1)]
    for i in range(0,len(mot1)+1):
        table[i][0]=i
    for j in range(0,len(mot2)+1):
        table[0][j]=j
    for i in range(1,len(mot1)+1):
```

```
            for j in range(1,len(mot2)+1):
                if mot1[i-1] == mot2[j-1]:
                    cout = 0
                else:
            # replacing modifiers does not cost much (XR, XSA and XVA are
                verbs when used in adnominals, thus being modifiers, not
                head of the sentence)
            # not a single POS for adverbs and adjectives, but rather
                constructions such as XR XSA EC for an adnominal
            if re.search(r'(MAG|MAJ|MM|XR|XSA|XSV)', mot1[i-1]) is not
                None: #
              cout = 0.5
            # replacing interjections, prefixes, suffixed morphological
                morphemes costs the least
            elif re.search(r'(IC|XPN|XSN)', mot1[i-1]) is not None:
              cout = 0.1
            # replacing the head of the sentence
            elif re.search(r'(VV|VA|VX)', mot1[i-1]) is not None:
              cout = 1.5
            else:
              cout = 1
          # add
                table[i][j] = min(table[i-1][j]+1,table[i][j-1]+1,table[i
                    -1][j-1]+cout)

    return table[len(mot1)][len(mot2)]

def edit_distance_en(mot1,mot2):

    table = [[0 for j in range(len(mot2)+1)] for i in range(len(mot1)
        +1)]
    for i in range(0,len(mot1)+1):
        table[i][0]=i
    for j in range(0,len(mot2)+1):
        table[0][j]=j
    for i in range(1,len(mot1)+1):
        for j in range(1,len(mot2)+1):
            if mot1[i-1] == mot2[j-1]:
                cout = 0
            else:
            # replacing modifiers
            if re.search(r'(AJ0|AJC|AJS|AT0|AV0|AVP|CRD)', mot1[i-1]) is
                not None: #
              cout = 0.5
            # replacing interjections
            elif re.search(r'(ITJ)', mot1[i-1]) is not None:
              cout = 0.1
            # replacing the head of the sentence
            elif re.search(r'(VBB|VBD|VBG|VBI|VBN|VBZ|VDB|VDD|VDG|VDI|VDN
```

```
            |VDZ|VHB|VHD|VHG|VHI|VHN|VHZ|VM0|VVB|VVD|VVG|VVI|VVN|VVZ)'
            , mot1[i-1]) is not None:
          cout = 1.5
        else:
          cout = 1
55          table[i][j] = min(table[i-1][j]+1,table[i][j-1]+1,table[i
                -1][j-1]+cout)

    return table[len(mot1)][len(mot2)]
```

Appendix C

Output files

The results of our experiments are listed in this Appendix, and organised as follows:

- number of input sentences;
- type of input;
- similarity measure;
- genre.

Each section represents a parameter that was tested, and for each parameter, two sets of experiments were run, one for the 'default mode' (searching for the same construction in the same context) and one for the 'distributional analysis mode' (searching for a different construction in the same context). Details on these modes are given in 5.3.4.

This organisation is the same as in Section 6.3 where the results are analysed.

C.1 Number of Input

C.1.1 Mode 1 – Default

Single input

-(u)lo **-(으)로**

```
Data:
  (a) Query: 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 9666

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (0.384615384615)
2. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (0.384615384615)
3. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (0.384615384615)
4. 피/NNG 는/JX 머리/NNG 에서/JKB 얼굴/NNG 로/JKB 흘러내리/VV 었/EP 다/EF ./SF
    (0.466666666667)
5. .../SE .../SE 발톱/NNG 의/JKG 길이/NNG 는/JX 얼마/NNG 로/JKB 하/VV ㄹ까/EF ?/SF (0.5)
6. 강도/NNG 살인/NNG 혐의/NNG 는/JX 조사/NNG 과정/NNG 에서/JKB 과실/NNG 치사/NNG 로/JKB 바
    뀌/VV 었/EP 습니다/EF ./SF (0.5)
7. 밥/NNG 어미/NNG 는/JX 구덩이/NNG 로/JKB 내려가/VV 지/EC 않/VX 았/EP 다/EF ./SF (0.5)
8. 남자/NNG 는/JX 마루/NNG 로/JKB 올라가/VV 아서/EC 아기/NNG 를/JKO 안/VV 았/EP 다/EF ./SF
    (0.529411764706)
9. 비/NNG 는/JX 진눈깨비/NNG 로/JKB 변하/VV 아/EC 가/VX 고/EC 있/VX 었/EP 다/EF ./SF
    (0.529411764706)
10. 어머니/NNG 는/JX 머리/NNG 를/JKO 젓/VV 으며/EC 뒤/NNG 로/JKB 물러앉/VV 았/EP 다/EF ./
    SF (0.529411764706)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 테/NNG 를/JKO 나무/NNG 로/JKB 두르/VV ㄴ/ETM ./SF (2)
2. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (2)
3. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (2)
4. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (2)
5. 붉/VA 은/ETM 피/NNG 로/JKB 쓰/VV ㄴ/ETM ./SF (3)
6. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (3)
7. 자기/NP 자리/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (3)
8. 꼬리/NNG 도/JX 볼멘소리/NNG 로/JKB 투덜거리/VV 었/EP 습니다/EF ./SF (3)
9. 출입문/NNG 도/JX 위아래/NNG 로/JKB 덜거덕거리/VV 었/EP 다/EF ./SF (3)
10. 한참/NNG 걷/VV 어서/EC 기관실/NNG 로/JKB 가/VV ㄴ다/EF ./SF (3)
```

-ato/eto -아도/어도

```
Data:
  (a) Query: 문제/NNG 가/JKS 어렵/VA 어도/EC 끝/NNG 까지/JX 푸/VV ㄹ/ET 거/NNB 이/VCP 에
    요/EF ./SF
  (b) Target: [ " 어도/ECD "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 977

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. "/SS 그러/VV 어도/EC 하/VV 아/EC 보/VX ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF "/SS
    (0.727272727273)
```

```
2. "/SS 그거/NP 만/JX 보이/VV 어/EC 주/VX 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF
   (0.739130434783)
3. 커피/NNG 잔/NNG 은/JX 그/MM 뒤/NNG 에/JKB 치우/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에
   요/EF ./SF (0.75)
4. 자리/NNG 가/JKS 없/VA 으면/EC ,/SP 돌아오/VV 고/EC 싶/VX 어도/EC 어려워하/VV ㄹ/ETM 거/
   NNB 이/VCP 에요/EF ./SF (0.76)
5. "/SS 나/NP ㄴ /JX 친구/NNG 없/VA 어도/EC ,/SP 가늘/VA 고/EC 길/VA 게/EC 살/VV ㄹ/ETM
   거/NNB 이/VCP 에요/EF ./SF "/SS (0.785714285714)
6. 그런데/MAJ 아무리/MAG 기다리/VV 어도/EC 전구/NNG 를/JKO 갈/VV 아/EC 주/VX 러/EC 오/VV 지/
   EC ㄹ/JKO 않/VX 았/EP 던/ETM 거/NNB 이/VCP 에요/EF ./SF (0.8)
7. 왜냐하면/MAG .../SE .../SE 왜냐하면/MAG .../SE .../SE 때리/VV 구/EC 싶/VX 어도/EC 때리/VV
   ㄹ/ETM 수/NNB 가/JKS 없/VA 기/ETN 때문/NNB 이/VCP 에요/EF ./SF (0.857142857143)
8. "/SS 그러/VV 어도/EC 누구/NP 이/VCP ㄴ가/EC 는/JX 오랫동안/NNG 묵묵히/MAG 경복궁/NNP 돌담/
   NNG 을/JKO 따르/VV 아/EC 서/VV 어/EC 있/VX 던/ETM 오래/MAG 되/XSV ㄴ/ETM 가죽나무/NNG
   들/XSN 을/JKO 떠올리/VV 고/EC 는/JX 하/VX ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF
   (0.860465116279)
9. 내/NP 가/JKS 숨/NNG 을/JKO 쉬/VV 어/EC 지/VX ㄹ/ETM 못/MAG 하/XSV 고/EC 데굴데굴/MAG 길
   바닥/NNG 을/JKO 구르/VV 어도/EC 사람/NNG 들/XSN 은/JX 태연/NNG 하/XSA ㄴ/ETM 얼굴/NNG
   로/JKB 나/NP 의/JKG 앞/NNG 을/JKO 지나가/VV 는/ETM 거/NNB 이/VCP 에요/EF ./SF
   (0.863636363636)
10. 그러/VV 어도/EC 그이/NP 는/JX 어렵/VA 게/EC 벌/VV ㄴ/ETM 돈/NNG 을/JKO 늦/VA 게/EC 나
    마/JX 제대로/MAG 쓰/VV 고/EC 있/VX 는/ETM 셈/NNB 이/VCP 에요/EF ./SF (0.875)

  Word forms (except lexical items) + POS
1. 자리/NNG 가/JKS 없/VA 으면/EC ,/SP 돌아오/VV 고/EC 싶/VX 어도/EC 어려워하/VV ㄹ/ETM 거/
   NNB 이/VCP 에요/EF ./SF (0.6)
2. "/SS 우리/NP 가/JKS 이렇/VA 게/EC 물러서/VV 려고/EC 오늘/NNG 까지/JX 오/VV ㄴ/ETM 거/NNB
   이/VCP 냐/EF ?/SF 지금/MAG 외롭/VA 고/EC 힘들/VA 어도/EC 참/VV 았/EP 어야지/EF ./SF
   (0.705882352941)
3. "/SS 나/NP ㄴ/JX 친구/NNG 없/VA 어도/EC ,/SP 가늘/VA 고/EC 길/VA 게/EC 살/VV ㄹ/ETM
   거/NNB 이/VCP 에요/EF ./SF "/SS (0.714285714286)
4. "/SS 그러/VV 어도/EC 하/VV 아/EC 보/VX ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF "/SS
   (0.727272727273)
5. 그런데/MAJ 아무리/MAG 기다리/VV 어도/EC 전구/NNG 를/JKO 갈/VV 아/EC 주/VX 러/EC 오/VV 지/
   EC ㄹ/JKO 않/VX 았/EP 던/ETM 거/NNB 이/VCP 에요/EF ./SF (0.733333333333)
6. 늦/VA 어도/EC 내일/NNG 까지/JX 는/JX 데스크/NNG 에/JKB 제출/NNG 하/XSV 아야/EC 하/VX ㄴ
   다/EF ./SF (0.739130434783)
7. "/SS 그거/NP 만/JX 보이/VV 어/EC 주/VX 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF
   (0.739130434783)
8. 커피/NNG 잔/NNG 은/JX 그/MM 뒤/NNG 에/JKB 치우/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에
   요/EF ./SF (0.75)
9. 예쁘/VA 어도/EC 권태/NNG 롭/XSA 고/EC 못생기/VA 어도/EC 권태/NNG 롭/XSA 다/EF ./SF
   (0.777777777778)
10. 뭐/IC ,/SP 총각/NNG 이/VCP 라고/EC 부르/VV 는/ETM 거/NNB 보다/JKB 야/JX 아저씨/NNG 가/
    JKS 그렇/VA 어도/EC 낫/VA 지/EF ./SF (0.777777777778)

10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. "/SS 그러/VV 어도/EC 하/VV 아/EC 보/VX ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF "/SS (8.5)
2. 안/MAG 들리/VV 어도/EC 그만/MAG 이/VCP 다/EF ./SF (9)
3. 「/SS 그러/VV 어도/EC 마찬가지/NNG 이/VCP 야/EF ./SF (9)
4. "/SS 언제/MAG 듣/VV 어도/EC 좋/VA 은/ETM 곡/NNG 이/VCP 야/EF ./SF (9.5)
5. 내일/NNG 쯤/XSN 부터/JX 먹/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 야/EF ./SF (9.5)
6. 애/NP 가/JKS 이렇/VA 어/EC 뵈/VV 어도/EC 대학/NNG 중퇴/NNG 이/VCP 라구요/EF ./SF (9.5)
7. 크/VV 어도/EC 마찬가지/NNG 이/VCP 었/EP 다/EF ./SF (10)
8. 그러/VV 어도/EC 너무/MAG 자책/NNG 하/XSV 지/EC 말/VX 아요/EF ./SF (10)
9. 언제/MAG 듣/VV 어도/EC 씩씩/XR 하/XSA ㄴ/ETM 목소리/NNG 이/VCP 었/EP 다/EF ./SF (10)
10. "/SS 그러/VV 어도/EC 아직/MAG 일교차/NNG 가/JKS 심하/VA ㅂ니다/EF ./SF (10)

  Word forms (except lexical items) + POS
1. 밤/NNG 이/JKS 깊/VA 어도/EC 어미/NNG 는/JX 오/VV 지/EC 않/VX 았/EP 다/EF ./SF (6)
```

```
2. 내일/NNG 쯤/XSN 부터/JX 먹/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 야/EF ./SF (7)
3. 자리/NNG 가/JKS 없/VA 으면/EC ,/SP 돌아오/VV 고/EC 싶/VX 어도/EC 어려워하/VV ㄹ/ETM 거/
   NNB 이/VCP 에요/EF ./SF (7)
4. 밥/NNG 은/JX 굶/VV 어도/EC 술/NNG 은/JX 마시/VV 어야/EC 하/VX 았/EP 으니까/EF ./SF
   (7.5)
5. 커피/NNG 잔/NNG 은/JX 그/MM 뒤/NNG 에/JKB 치우/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에
   요/EF ./SF (7.5)
6. 아줌마/NNG 가/JKS 백/NR 번/NNB 을/JKO 죽/VV 어도/EC 안/MAG 되/XSV ㄴ다구/EF ./SF (8)
7. 얘/NP 가/JKS 이렇/VA 어/EC 뵈/VV 어도/EC 대학/NNG 중퇴/NNG 이/VCP 라구요/EF ./SF (8)
8. 「/SS 그러/VV 어도/EC 마찬가지/NNG 이/VCP 야/EF ./SF (8)
9. 차장/NNG 급/NNG 이하/NNG 직원/NNG 은/JX 그러/VV 어도/EC 낫/VA 았/EP 다/EF ./SF (8)
10. 그러/VV 어도/EC 더위/NNG 는/JX 사라지/VV 지/EC 않/VX 았/EP 다/EF ./SF (8)
```

Multiple input

-(u)lo -(으)로

```
Data:
  (a) Query:
    - 젓가락/NNG 으로/JKB 먹/VV 습니다/EF ./SF
    - 한국말/NNG 로/JKB 말하/VV 시/EP ㅂ시오/EF ./SF
    - 버스/NNG 로/JKB 오/VV 았/EP 습니다/EF ./SF
    - 연필/NNG 로/JKB 쓰/VV ㅂ니다/EF ./SF
    - 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 48330

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (0.476393024245)
2. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (0.491012713722)
3. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (0.491012713722)
4. 피/NNG 는/JX 머리/NNG 에서/JKB 얼굴/NNG 로/JKB 흘러내리/VV 였/EP 다/EF ./SF
   (0.568977195755)
5. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (0.576923076923)
6. 침실/NNG 로/JKB 가/VV ㄴ다/EF ./SF (0.576923076923)
7. 선장실/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (0.576923076923)
8. K/SL 는/JX 대합실/NNG 로/JKB 들어가/VV 았/EP 다/EF ./SF (0.578811369509)
9. 나/NP 는/JX 동사무소/NNG 로/JKB 가/VV 았/EP 다/EF ./SF (0.578811369509)
10. 강도/NNG 살인/NNG 혐의/NNG 는/JX 조사/NNG 과정/NNG 에서/JKB 과실/NNG 치사/NNG 로/JKB 바
    뀌/VV 었/EP 습니다/EF ./SF (0.582588546839)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (1.84615384615)
2. 침실/NNG 로/JKB 가/VV ㄴ다/EF ./SF (1.84615384615)
3. 선장실/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (1.84615384615)
4. 침대/NNG 로/JKB 다가가/VV 았/EP 다/EF ./SF (2.34146341463)
5. 테/NNG 를/JKO 나무/NNG 로/JKB 두르/VV ㄴ/ETM ./SF (2.52631578947)
```

```
6. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (2.615803814 71)
7. 자기/NP 자리/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (2.66666666667)
8. 벌컥/MAG 모/NNG 로/JKB 돌아눕/VV 는다/EF ./SF (2.66666666667)
9. "/SS 공중전화/NNG 로/JKB 해보/VV 지/EF ./SF (2.66666666667)
10. 내일/NNG 오하이오/NNP 로/JKB 떠나/VV 아/EF ./SF (2.66666666667)
```

-ato/eto -아도/어도

```
Data:
  (a) Query:
    - 저/NP 는/JX 피곤/NNG 하/XSV 어도/EC 아침/NNG 운동/NNG 은/JX 꼭/MA 하/VV 어요/EF ./SF
    - 아무리/MA 바쁘/VA 아도/EC 아침/NNG 식사/NNG 는/JX 꼭/MA 하/VV 세요/EF ./SF
    - 문제/NNG 가/JKS 어렵/VA 어도/EC 끝/NNG 까지/JX 푸/VV ㄹ/ET 거/NNB 이/VCP 에요/EF ./
      SF
  (b) Target: [ " 어도/ECD "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 2931

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. "/SS 그러/VV 어도/EC 하/VV 아/EC 보/VX ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF "/SS
    (0.842105263158)
2. "/SS 그거/NP 만/JX 보이/VV 어/EC 주/VX 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF
    (0.85)
3. 커피/NNG 잔/NNG 은/JX 그/MM 뒤/NNG 에/JKB 치우/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에
    요/EF ./SF (0.857142857143)
4. 자리/NNG 가/JKS 없/VA 으면/EC ,/SP 돌아오/VV 고/EC 싶/VX 어도/EC 어려워하/VV ㄹ/ETM 거/
    NNB 이/VCP 에요/EF ./SF (0.863636363636)
5. "/SS 나/NP ㄴ/JX 친구/NNG 없/VA 어도/EC ,/SP 가늘/VA 고/EC 길/VA 게/EC 살/VV ㄹ/ETM
    거/NNB 이/VCP 에요/EF ./SF "/SS (0.88)
6. 그런데/MAJ 아무리/MAG 기다리/VV 어도/EC 전구/NNG 를/JKO 갈/VV 아/EC 주/VX 러/EC 오/VV 지/
    EC ㄹ/JKO 않/VX 았/EP 던/ETM 거/NNB 이/VCP 에요/EF ./SF (0.888888888889)
7. 왜냐하면/MAG .../SE .../SE 왜냐하면/MAG .../SE .../SE 때리/VV 구/EC 싶/VX 어도/EC 때리/VV
    ㄹ/ETM 수/NNB 가/JKS 없/VA 기/ETN 때문/NNB 이/VCP 에요/EF ./SF (0.923076923077)
8. "/SS 그러/VV 어도/EC 누구/NP 이/VCP ㄴ가/EC 는/JX 오랫동안/NNG 묵묵히/MAG 경복궁/NNP 돌담/
    NNG 을/JKO 따르/VV 아/EC 서/VV 어/EC 있/VX 던/ETM 오래/MAG 되/XSV ㄴ/ETM 가죽나무/NNG
    들/XSN 을/JKO 떠올리/VV 고/EC 는/JX 하/VX ㄹ/ETM 거/NNB 이/VCP 에요/EF ./SF (0.925)
9. 내/NP 가/JKS 숨/NNG 을/JKO 쉬/VV 어/EC 지/VX ㄹ/ETM 못/MAG 하/XSV 고/EC 데굴데굴/MAG 길
    바닥/NNG 을/JKO 구르/VV 어도/EC 사람/NNG 들/XSN 은/JX 태연/NNG 하/XSA ㄴ/ETM 얼굴/NNG
    로/JKB 나/NP 의/JKG 앞/NNG 을/JKO 지나가/VV 는/ETM 거/NNB 이/VCP 에요/EF ./SF
    (0.926829268293)
10. 그러/VV 어도/EC 그이/NP 는/JX 어렵/VA 게/EC 별/VV ㄴ/ETM 돈/NNG 을/JKO 늦/VA 게/EC 나
    마/JX 제대로/MAG 쓰/VV 고/EC 있/VX 는/ETM 셈/NNB 이/VCP 에요/EF ./SF
    (0.933333333333)

  Word forms (except lexical items) + POS
1. 자리/NNG 가/JKS 없/VA 으면/EC ,/SP 돌아오/VV 고/EC 싶/VX 어도/EC 어려워하/VV ㄹ/ETM 거/
    NNB 이/VCP 에요/EF ./SF (0.75)
2. 늦/VA 어도/EC 내일/NNG 까지/JX 는/JX 데스크/NNG 에/JKB 제출/NNG 하/XSV 아야/EC 하/VX ㄴ
    다/EF ./SF (0.790697674419)
3. 커피/NNG 잔/NNG 은/JX 그/MM 뒤/NNG 에/JKB 치우/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 에
    요/EF ./SF (0.805369127517)
4. 그러/VV 어도/EC 아이/NNG 는/JX 모닥불/NNG 에/JKB 눈길/NNG 한번/NNG 안/MAG 보내/VV 었/EP
    어요/EF ./SF (0.806597379123)
```

```
5. 씨티/NNG 투어/NNG 버스/NNG 는/JX 그러/VV 어도/EC 지구/NNG 는/JX 돌/VV ㄴ다/EF ./SF
    (0.821515892421)
6. "/SS 우리/NP 가/JKS 이렇/VA 게/EC 물러서/VV 려고/EC 오늘/NNG 까지/JX 오/VV ㄴ/ETM 거/NNB
    이/VCP 냐/EF ?/SF 지금/MAG 외롭/VA 고/EC 힘들/VA 어도/EC 참/VV 았/EP 어야지/EF ./SF
    (0.827586206897)
7. 얘/NP 가/JKS 이렇/VA 어/EC 뵈/VV 어도/EC 대학/NNG 중퇴/NNG 이/VCP 라구요/EF ./SF
    (0.829268292683)
8. "/SS 나/NP ㄴ/JX 친구/NNG 없/VA 어도/EC ,/SP 가늘/VA 고/EC 길/VA 게/EC 살/VV ㄹ/ETM
    거/NNB 이/VCP 에요/EF ./SF "/SS (0.833333333333)
9. 그런데/MAJ 아무리/MAG 기다리/VV 어도/EC 전구/NNG 를/JKO 갈/VV 아/EC 주/VX 러/EC 오/VV 지/
    EC ㄹ/JKO 않/VX 았/EP 던/ETM 거/NNB 이/VCP 에요/EF ./SF (0.833558863329)
10. 언제/MAG 먹/VV 어도/EC 유리/NNG 에서/JKB 는/JX 어미/NNG 의/JKG 눈물/NNG 냄새/NNG 가/JKS
    나/VV 았/EP 다/EF ./SF (0.834728033473)

10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. 그러/VV 어도/EC 아무/MM 대꾸/NNG 를/JKO 안/MAG 하/VV ㄴ다/EF ./SF (9.0365448505)
2. 안/MAG 들리/VV 어도/EC 그만/MAG 이/VCP 다/EF ./SF (9.12)
3. 「/SS 그러/VV 어도/EC 마찬가지/NNG 이/VCP 야/EF ./SF (9.12)
4. 크/VV 어도/EC 마찬가지/NNG 이/VCP 었/EP 다/EF ./SF (9.60674157303)
5. 그러/VV 어도/EC 모자라/VV 았/EP 다/EF ./SF (9.60674157303)
6. 그러/VV 어도/EC 움직이/VV 지/EC 않/VX 았/EP 다/EF ./SF (9.60674157303)
7. 언제/MAG 듣/VV 어도/EC 좋/VA 은/ETM 소리/NNG ./SF (9.60674157303)
8. "/SS 그러/VV 어도/EC 하/VV 아야/EC 되/VV 어/EF ./SF (9.60674157303)
9. 그러/VV 어도/EC 가/VV 아야지/EF ./SF (9.60674157303)
10. "/SS 그러/VV 어도/EC 좀/MAG 낫/VA 지/EF ./SF (9.60674157303)

  Word forms (except lexical items) + POS
1. 밤/NNG 이/JKS 깊/VA 어도/EC 어미/NNG 는/JX 오/VV 지/EC 않/VX 았/EP 다/EF ./SF
    (7.34693877551)
2. "/SS 그러/VV 어도/EC 안내/NNG 방송/NNG 을/JKO 잘/MAG 귀담아듣/VV 으세요/EF ./SF
    (7.54838709677)
3. "/SS 사진/NNG 안/MAG 붙이/VV 어도/EC 되/VV 어요/EF ./SF (8.02547770701)
4. 얘/NP 가/JKS 이렇/VA 어/EC 뵈/VV 어도/EC 대학/NNG 중퇴/NNG 이/VCP 라구요/EF ./SF
    (8.08988764045)
5. 「/SS 그러/VV 어도/EC 마찬가지/NNG 이/VCP 야/EF ./SF (8.11940298507)
6. 살/NNG 은/JX 찌/VV 었/EP 어도/EC 말/NNG 은/JX 바로/MAG 하/VV 아/EF ./SF
    (8.29841897233)
7. 내일/NNG 쯤/XSN 부터/JX 먹/VV 어도/EC 되/VV ㄹ/ETM 거/NNB 이/VCP 야/EF ./SF
    (8.30070921986)
8. 아무리/MAG 교육/NNG 시키/VV 어도/EC 잘/MAG 안/MAG 되/VV ㅂ니다/EF ./SF (8.32369942197)
9. 씨티/NNG 투어/NNG 버스/NNG 는/JX 그러/VV 어도/EC 지구/NNG 는/JX 돌/VV ㄴ다/EF ./SF
    (8.32369942197)
10. 차장/NNG 급/NNG 이하/NNG 직원/NNG 은/JX 그러/VV 어도/EC 낫/VA 았/EP 다/EF ./SF
    (8.35494880546)
```

C.1.2 Mode 2 – Distributional Analysis

Single input

```
Data:
  (a) Query: 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB " ]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 48722

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. 바다/NNP 는/JX 자신/NNG 이/JKS 한국인/NNG 이/VCP 라고/EC 너무나/MAG 자랑/NNG 스럽/XSA 게/
    EC 말/NNG 하/XSV ㄴ/ETM 것/NNB 에/JKB 비하/VV 아/EC 한국말/NNG 이/JKS 서투르/VA 었/EP
    고/EC ,/SP 새우/NNG 가/JKS 자신/NNG 을/JKO 노려보/VV 는/ETM 것/NNB 같/VA 아/EC 새우젓/
    NNG 이/JKS 들어가/VV ㄴ/ETM 김치/NNG 는/JX 먹/VV 을/ETM 수/NNB 없/VA 다는/ETM 고백/NNG
    을/JKO 하/VV 며/EC 무안/NNG 하/XSA 아/EC 하/VX 는/ETM 표정/NNG 을/JKO 짓/VV 기/ETN
    도/JX 하/VX 았/EP 다/EF ./SF (0.970149253731)
2. 다른/MM 여자/NNG 랑/JKB 자/VV 는/ETM 것/NNB 보다/JKB 안/MAG 낫/VA 니/EF ?/SF (1.0)
3. 처음/NNG 에/JKB 오/VV ㄴ/ETM 사람/NNG 은/JX 비닐하우스촌/NNG 에서/JKB 데려오/VV ㄴ/ETM 남
    자/NNG 아이/NNG 이/VCP 었/EP 고/EC ,/SP 두/MM 번/NNB 째/XSN 는/JX 귀/NNG 가/JKS 전혀/
    MAG 들리/VV 지/EC 않/VX 는/ETM 할아버지/NNG 이/VCP 었/EP 다/EF ./SF (1.0)
4. 말/NNG 은/JX 자주/MAG 생각/NNG 으로부터/JKB 의/JKG 독립/NNG 을/JKO 꿈꾸/VV ㄴ다/EF ./SF
    (1.0)
5. 그리/MAG 하/XSV 어도/EC 그중/NNG 에/JKB 화롯불/NNG 을/JKO 가져오/VV ㄴ/ETM 계집애/NNG 만/
    JX 은/JX 저희/NP 들/XSN 축/NNB 에서/JKB 좁/MAG 쫄리/VV 어/EC 지내/VV 는지/EC 한풀/NNG
    이/JKS 죽/VV 어서/EC 떠들/VV 는/ETM 꼴/NNG 만/JX 웃/VV 으며/EC 가만히/MAG 바라보/VV 고/
    EC 앉/VV 았/EP 다/EF ./SF (1.0)
6. 어젯밤/NNG 에/JKB 언뜻/MAG 지나치/VA 면서/EC 보/VX 았/EP 을/ETM 때/NNG 는/JX 한/MM 삼천/
    NR 평/NNB 쯤/XSN 되/VV 는/ETM 줄/NNB 알/VV 았/EP 는데/EC ,/SP 낮/NNG 에/JKB 보/VV
    니/EC 훨씬/MAG 더/MAG 넓/VA 은/ETM 것/NNB 같/VA 다/EF ./SF (1.0)
7. 나/NP 의/JKG 집/NNG 에서/JKB 아이/NNG 들/XSN 둘/NR 이/JKS 잠자/VV 고/EC 있/VX 다/EF ./
    SF (1.0)
8. 어머니/NNG 가/JKS 아버지/NNG 의/JKG 선산/NNG 에/JKB 묻히/VV 던/ETM 날/NNG 에/JKB 도/JX
    아버지/NNG 는/JX 손수건/NNG 을/JKO 꺼내/VV 어/EC 눈가/NNG 를/JKO 닦/VV 았/EP 다/EF ./
    SF (1.0)
9. 창/NNG 쪽/NNB 에/JKB 붙/VV 어/EC 있/VX 는/ETM 3/SN 인/NNG 용/XSN 소파/NNG 위/NNG 에/
    JKB 여자/NNG 가/JKS 있/VV 었/EP 다/EF ./SF (1.0)
10. 아무리/MAG 상상/NNG 은/JX 자유/NNG 이/VCP 라지만/EC 로라/NNP 는/JX 자네/NP 하고/JKB 너
    무/MAG 멀리/MAG 있/VV 어/EF ./SF (1.0)

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (0.666666666667)
2. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (0.666666666667)
3. 아이/NNG 는/JX 학교/NNG 에/JKB 들어가/VV 았/EP 을까/EF ?/SF (0.692307692308)
4. 효과/NNG 는/JX 뜻밖/NNG 으로/JKB 크/VA 었/EP 다/EF ./SF (0.692307692308)
5. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (0.692307692308)
6. 언니/NNG 는/JX 집/NNG 에/JKB 없/VA 어/EF ./SF "/SS (0.692307692308)
7. 피/NNG 는/JX 흙/NNG 속/NNG 으로/JKB 스미/VV ㄴ다/EF ./SF (0.692307692308)
8. 먹이/NNG 는/JX 다음/NNG 에/JKB 주/VV 자/EF ./SF "/SS (0.692307692308)
9. 니코틴/NNG 냄새/NNG 는/JX 예상/NNG 보다/JKB 심하/VA 았/EP 다/EF ./SF (0.714285714286)
10. 그것/NP 으로/JKB 수사/NNG 는/JX 끝/NNG 이/VCP 었/EP 다/EF ./SF (0.714285714286)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (2)
```

```
2. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (2)
3. 담뱃가게/NNG 옆/NNG 에/JKB 있/VV 잖아요/EF ./SF (3)
4. 밖/NNG 에/JKB 는/JX 비/NNG 가/JKS 오/VV 아/EF ./SF (3)
5. 사람/NNG 에게/JKB 점수/NNG 를/JKO 매기/VV 다니/EF ./SF (3)
6. 신청서/NNG 는/JX 여기/NP 에/JKB 있/VV 습니다/EF ./SF (3)
7. 취중/NNG 에/JKB 택시/NNG 를/JKO 잡/VV 다가/EF ./SF (3)
8. 큰오빠/NNG 한테/JKB 여자/NNG 가/JKS 있/VV 대/EF ./SF (3)
9. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (3)
10. 낙엽/NNG 들/XSN 비/NNG 에/JKB 젖/VV 는다/EF ./SF (3)
```

Multiple input

```
Data:
  (a) Query:
    - 젓가락/NNG 으로/JKB 먹/VV 습니다/EF ./SF
    - 한국말/NNG 로/JKB 말하/VV 시/EP ㅂ시오/EF ./SF
    - 버스/NNG 로/JKB 오/VV 았/EP 습니다/EF ./SF
    - 연필/NNG 로/JKB 쓰/VV ㅂ니다/EF ./SF
    - 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB " ]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 48722

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. "/SS 약고개/NNP 에서/JKB 오/VV 았/EP 습니다/EF ./SF "/SS (0.878048780488)
2. 지나/VV ㄴ/ETM 일요일/NNG 저녁/NNG 에/JKB 오/VV 았/EP 습니다/EF ./SF (0.893141945774)
3. "/SS 저기/NP 동쪽/NNG 에서/JKB 오/VV 았/EP 습니다/EF ./SF "/SS (0.893141945774)
4. 시장/NNG 에/JKB 가/VV 았/EP 습니다/EF ./SF (0.908108108108)
5. "/SS 그거/NP ,/SP 국무성/NNG 에서/JKB 통지/NNG 오/VV 았/EP 습니다/EF ./SF "/SS
    (0.914285714286)
6. 나/NP 는/JX 오늘/NNG 아침/NNG 공항/NNG 에/JKB 가/VV 았/EP 다/EC 오/VV 았/EP 습니다/EF
    ./SF (0.928409947249)
7. "/SS 그/MM 숲/NNG 에/JKB 다녀오/VV 았/EP 습니다/EF ./SF (0.929032258065)
8. 별님이/NNP 도/JX 새댁/NNG 옆/NNG 에/JKB 앉/VV 았/EP 습니다/EF ./SF (0.936280884265)
9. "/SS 저/NP 는/JX 여기/NP 서/JKB 태어나/VV 았/EP 습니다/EF ./SF (0.936280884265)
10. 태양/NNG 은/JX 여전히/MAG 동쪽/NNG 에서/JKB 떠오르/VV 았/EP 습니다/EF ./SF
    (0.936280884265)

  Word forms (except lexical items) + POS
1. 할머니/NNG 는/JX 새댁/NNG 보다/JKB 더/MAG 기뻐하/VV 았/EP 습니다/EF ./SF
    (0.757780784844)
2. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (0.774193548387)
3. 피/NNG 는/JX 흙/NNG 속/NNG 으로/JKB 스미/VV ㄴ다/EF ./SF (0.794952681388)
4. 밥/NNG 어미/NNG 는/JX 방/NNG 으로/JKB 들어가/VV 았/EP 다/EF ./SF (0.79746261894)
5. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (0.8)
6. 아이/NNG 는/JX 학교/NNG 에/JKB 들어가/VV 았/EP 을까/EF ?/SF (0.801781737194)
7. 엄마/NNG 는/JX 신경질/NNG 적/XSN 으로/JKB 말/NNG 하/XSV 았/EP 습니다/EF ./SF
    (0.804387568556)
8. 효과/NNG 는/JX 뜻밖/NNG 으로/JKB 크/VA 었/EP 다/EF ./SF (0.808988764045)
9. 끈기/NNG 는/JX 신념/NNG 을/JKO 바탕/NNG 으로/JKB 하/VV ㄴ다/EF ./SF (0.812182741117)
10. 남/NNP 경사/NNG 는/JX 진심/NNG 으로/JKB 묻/VV 었/EP 다/EF ./SF (0.812182741117)

10 closest sentences using the Levenshtein distance:

```

```
  Word forms (except lexical items) + POS
1. 밖/NNG 으로/JKB 나오/VV 세요/EF ./SF (2.66666666667)
2. 뱃간/NNG 으로/JKB 가/VV ㄴ다/EF ./SF (2.66666666667)
3. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (2.69662921348)
4. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (2.73504273504)
5. 시장/NNG 에/JKB 가/VV 았/EP 습니다/EF ./SF (2.77456647399)
6. 뒤/NNG 에/JKB 타/VV 라/EF ./SF (2.90909090909)
7. 비/NNG 에/JKB 젖/VV 어/EF ./SF (2.90909090909)
8. 천장/NNG 에서/JKB 나/VV 아요/EF ./SF (2.90909090909)
9. 토요일/NNG 에/JKB 보/VV 자/EF ./SF (2.90909090909)
10. 저녁/NNG 에/JKB 가/VV 죠/EF ./SF (2.90909090909)
```

C.2 Type of Input

C.2.1 Mode 1 – Default

-(u)lcito moluta -(으)ㄹ지도 모르다

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 259

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. 미치/VV ㄹ지/EC 도/JX 모르/VV 아/EC (0.818181818182)
2. 그렇/VA ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.833333333333)
3. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 지/EF ./SF (0.846153846154)
4. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아/EF ./SF (0.846153846154)
5. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아요/EF ./SF (0.846153846154)
6. 어쩌면/MAG 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 습니다/EF ./SF (0.857142857143)
7. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 군요/EF ./SF (0.857142857143)
8. 괜히/MAG 화풀이/NNG 당하/XSV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (0.857142857143)
9. 혹시/MAG 차/NNG 가/JKS 오/VV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (0.866666666667)
10. 그것/NP ㄴ/JX 사실/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.866666666667)

  Word forms (except lexical items) + POS
1. 그러면/MAJ 조금/NNG 은/JX 쉽/VA 어/EC 지/VX ㄹ지/EC 도/JX 모르/VV 아/EC 하/VV 고/EC 말/
    NNG 이/VCP 야/EF ./SF (0.727272727273)
2. 어쩌면/MAG 남편/NNG 은/JX 저/NP 를/JKO 죽이/VV ㄹ지/EC 도/JX 모르/VV 아요/EF ./SF
    (0.764705882353)
3. 변명/NNG 은/JX 진실/NNG 의/JKG 다른/MM 얼굴/NNG 이/VCP ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF
    (0.777777777778)
4. 게다가/MAG 타협/NNG 은/JX 생각/NNG 보다/JKB 길/VA 어/EC 지/VX ㄹ지/EC 도/JX 모르/ㄴ다VV/
    EF ./SF (0.789473684211)
5. 아니/IC 어쩌면/MAG 한순간/NNG 의/JKG 실수/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV 아/EC ,/SP 사
    랑/NNG 은/JX ./SF (0.8)
6. 그것/ㄴNP/JX 그렇/VA 게/EC 중요/NNG 하/ㄴXSA/ETM 사실/NNG 은/JX 아니/VCN ㄹ지/EC 도/JX
    모르/ㄴ다VV/EF ./SF (0.809523809524)
7. 미치/VV ㄹ지/EC 도/JX 모르/VV 아/EC (0.818181818182)
```

```
8. 이곳/NP 비/ㄴVV/ETM 공장/NNG 은/JX 어쩌면/MAG 나/NP 의/JKG 도약/NNG 의/JKG 발판/NNG 이/
    JKC 되/VV ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (0.826086956522)
9. 그렇/VA ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (0.833333333333)
10. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 지/EF ./SF (0.846153846154)

10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 지/EF ./SF (5)
2. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아/EF ./SF (5)
3. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아요/EF ./SF (5)
4. 그렇/VA ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (5)
5. 괜히/MAG 화풀이/NNG 당하/XSV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (5.5)
6. 어쩌면/MAG 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 습니다/EF ./SF (6)
7. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 군요/EF ./SF (6)
8. 미치/VV ㄹ지/EC 도/JX 모르/VV 아/EC (6)
9. 혹시/MAG 차/NNG 가/JKS 오/VV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (6.5)
10. 정말/MAG 그것/NP 은/JX 운명/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (6.5)

  Word forms (except lexical items) + POS
1. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 지/EF ./SF (4)
2. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아/EF ./SF (4)
3. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아요/EF ./SF (4)
4. 그렇/VA ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (4)
5. 어쩌면/MAG 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 습니다/EF ./SF (5)
6. "/SS 그래/IC 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (5)
7. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 군요/EF ./SF (5)
8. 괜히/MAG 화풀이/NNG 당하/XSV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (5)
9. 혹시/MAG 차/NNG 가/JKS 오/VV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (5.5)
10. 그것/ㄴNP/JX 사실/NNG 이/VCP ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (6)
```

-(u)lo -(으)로

```
Data:
  (a) Query: 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 48330

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. 모래/NNG 로/JKB 만들/VV ㄴ/ETM 벽/NNG ./SF (0.818181818182)
2. 죽음/NNG 이/JKS 그/NP 들/XSN 을/JKO 예술가/NNG 로/JKB 만들/VV ㄴ다/EF ./SF
    (0.866666666667)
3. 그냥/MAG 심심풀이/NNG 로/JKB 만들/VV 어/EC 보/VX ㄴ/ETM 거/NNB 이/VCP 에요/EF ./SF
    (0.875)
4. 「/SS 이것/NP ㄴ/JX 진짜/NNG 금실/NNG 로/JKB 만들/VV ㄴ/ETM 옷/NNG 이/VCP 예요/EF ./SF
    (0.882352941176)
5. 나/NP 를/JKO 바보/NNG 로/JKB 만들/VV 는/ETM 극도/NNG 의/JKG 너/NP 의/JKG 예민/XR 성/XSN
    ./SF (0.888888888889)
6. 잘나/VA ㄴ/ETM 놈/NNB 들/XSN 이/JKS 일/NNG 은/JX 왜/MAG 이따위/NP 로/JKB 만들/VV 냐/EF
    ./SF (0.888888888889)
7. 그렇/VA 다면/EC 그녀/NP 를/JKO 나/NP 의/JKG 노예/NNG 로/JKB 만들/VV 고/EC 싶/VX 다/EC
    .../SE .../SE ./SF (0.9)
```

```
8. 이것/NP ㄴ/JX 모두/MAG 제/NP 가/JKS 심심풀이/NNG 로/JKB 만들/VV 어/EC 보/VX ㄴ/ETM 것/
    NNB 이/VCP ㅂ니다/EF ./SF (0.9)
9. 「/SS 진짜/NNG .../SE .../SE 금실/NNG 로/JKB 만들/VV ㄴ/ETM .../SE .../SE 옷/NNG 이/VCP
    라서/EC 그렇/VA 어요/EF ./SF (0.9)
10. </SS 희수/NNP >/SS 의/JKG 한자/NNG 를/JKO </SS 喜囚/SH >/SS 로/JKB 만들/VV 면/EC 어
    떻/VA ㄹ까/EF ?/SF (0.9)

  Word forms (except lexical items) + POS
1. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (0.384615384615)
2. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (0.384615384615)
3. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (0.384615384615)
4. 피/NNG 는/JX 머리/NNG 에서/JKB 얼굴/NNG 로/JKB 흘러내리/VV 었/EP 다/EF ./SF
    (0.466666666667)
5. .../SE .../SE 발톱/NNG 의/JKG 길이/NNG 는/JX 얼마/NNG 로/JKB 하/VV ㄹ까/EF ?/SF (0.5)
6. 강도/NNG 살인/NNG 혐의/NNG 는/JX 조사/NNG 과정/NNG 에서/JKB 과실/NNG 치사/NNG 로/JKB 바
    뀌/VV 었/EP 습니다/EF ./SF (0.5)
7. 밥/NNG 어미/NNG 는/JX 구덩이/NNG 로/JKB 내려가/VV 지/EC 않/VX 았/EP 다/EF ./SF (0.5)
8. 남자/NNG 는/JX 마루/NNG 로/JKB 올라가/VV 아서/EC 아기/NNG 를/JKO 안/VV 았/EP 다/EF ./SF
    (0.529411764706)
9. 비/NNG 는/JX 진눈깨비/NNG 로/JKB 변하/VV 아/EC 가/VX 고/EC 있/VX 었/EP 다/EF ./SF
    (0.529411764706)
10. 어머니/NNG 는/JX 머리/NNG 를/JKO 젓/VV 으며/EC 뒤/NNG 로/JKB 물러앉/VV 았/EP 다/EF ./
    SF (0.529411764706)
10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. 그것/NP 도/JX 남자/NNG 로/JKB ./SF (5)
2. 그것/NP 도/JX 한국말/NNG 로/JKB ./SF (5)
3. 국내/NNG 로/JKB 세계/NNG 로/JKB ./SF (5)
4. 모래/NNG 로/JKB 만들/VV ㄴ/ETM 벽/NNG ./SF (5)
5. 산/NNG 으로/JKB 들/NNG 로/JKB ./SF (5)
6. 빨갛/VA ㄴ/ETM 것/NNB 로/JKB ./SF (5)
7. 그런/MM 거/NNB 로/JKB 요/JX ./SF (5)
8. "/SS 어디/NP 로/JKB 돌아가/VV 나/EF ./SF (5.5)
9. 붉/VA 은/ETM 피/NNG 로/JKB 쓰/VV ㄴ/ETM ./SF (5.5)
10. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (5.5)

  Word forms (except lexical items) + POS
1. 테/NNG 를/JKO 나무/NNG 로/JKB 두르/VV ㄴ/ETM ./SF (2)
2. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (2)
3. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (2)
4. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (2)
5. 붉/VA 은/ETM 피/NNG 로/JKB 쓰/VV ㄴ/ETM ./SF (3)
6. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (3)
7. 자기/NP 자리/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (3)
8. 꼬리/NNG 도/JX 볼멘소리/NNG 로/JKB 투덜거리/VV 었/EP 습니다/EF ./SF (3)
9. 출입문/NNG 도/JX 위아래/NNG 로/JKB 덜거덕거리/VV 었/EP 다/EF ./SF (3)
10. 한참/NNG 걷/VV 어서/EC 기관실/NNG 로/JKB 가/VV ㄴ다/EF ./SF (3)
```

C.2.2 Mode 2 – Distributional Analysis

-(u)lcito moluta -(으)ㄹ지도 모르다

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 1520

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.714285714286)
2. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (0.733333333333)
3. 아직/MAG 막히/VV 었/EP 을지/EC 도/JX 모르/VV 겠/EP 어/EF ./SF (0.733333333333)
4. 너/NP 의/JKG 말/NNG 이/JKS 맞/VV 을지/EC 도/JX 모르/VV 아/EF ./SF (0.75)
5. "/SS 오늘/NNG 은/JX 좀/MAG 늦/VV 을지/EC 도/JX 모르/VV ㄹ걸/EF !/SF (0.75)
6. 아마/MAG 울/VV 고/EC 있/VX 을지/EC 도/JX 모르/VV 았/EP 다/EF ./SF (0.75)
7. 그리고/MAJ 결국/NNG 에/JKB ㄴ/JX 빈손/NNG 만/JX 남/VV 을지/EC 도/JX 모르/VV ㄴ다구/EF ./
    SF (0.777777777778)
8. "/SS 그래서/MAJ ,/SP 통일감/NNG 이/JKS 떨어지/VV 었/EP 을지/EC 도/JX 모르/VV 아요/EF ./
    SF (0.777777777778)
9. 그녀/NP 는/JX 어쩌면/MAG 집/NNG 에/JKB 내려가/VV 았/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./
    SF (0.777777777778)
10. 어젯밤/NNG 나/NP 에게/JKB 전화/NNG 를/JKO 하/VV 았/EP 을지/EC 도/JX 모르/VV 겠/EP 다/EF
    ./SF (0.789473684211)

  Word forms (except lexical items) + POS
1. "/SS 오늘/NNG 은/JX 좀/MAG 늦/VV 을지/EC 도/JX 모르/VV ㄹ걸/EF !/SF (0.625)
2. 어쩌면/MAG 다시/MAG 박송미/NNP 에게/JKB 연락/NNG 하/XSV ㄹ/ETM 수/NNB 없/VA 을지/EC 도/
    JX 모르/VV 았/EP 다/EF ./SF (0.714285714286)
3. "/SS 뭐/IC ,/SP 더/MAG 이상/NNG 상종/NNG 하/XSV ㄹ/ETM 필요/NNG 없/VA 을지/EC 도/JX 모
    르/VV 아요/EF ./SF (0.714285714286)
4. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.714285714286)
5. 어쩌면/MAG 우리/NP 가/JKS 영영/MAG 헤어지/VV ㄹ/ETM 수/NNB 없/VA 을지/EC 도/JX 모르/VV ㄴ
    다는/ETM 예감/NNG 때문/NNB 에/JKB ?/SF (0.727272727273)
6. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (0.733333333333)
7. 시간/NNG 은/JX 더디/VA 게/EC 만/JX 흐르/VV 었/EP 다/EF ./SF (0.733333333333)
8. 아직/MAG 막히/VV 었/EP 을지/EC 도/JX 모르/VV 겠/EP 어/EF ./SF (0.733333333333)
9. 석고/NNG 를/JKO 떼/VV 어/EC 내/VX ㄹ/ETM 때/NNG 도/JX 아프/VA ㄹ/ETM 터/NNB 이/VCP 고/
    EC .../SE .../SE 하지만/MAJ 맨/XPN 얼굴/NNG 이/JKC 아니/VCN 니까/EC ,/SP 조금/NNG 은/
    JX 낫/VA 을지/EC 도/JX 모르/VV 겠/EP 습니다/EF ./SF "/SS (0.736842105263)
10. 그러나/MAJ 그/NP 는/JX 그것/NP 이/JKS 차라리/MAG 낫/VA 을지/EC 도/JX 모르/VV ㄴ다는/ETM
    생각/NNG 을/JKO 하/VV 았/EP 다/EF ./SF (0.739130434783)

10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4.5)
2. 권태기/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (5)
3. 「/SS 그렇/VA ㄹ는지/EC 도/JX 모르/VV 죠/EF ./SF (5)
4. 할미/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 죠/EF ./SF (5)
5. 그래서/MAJ 이/VCP 었/EP 는지/EC 도/JX 모르/VV ㄴ다/EF ./SF (5.5)
6. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (5.5)
7. 정신/NNG 병원/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 아/EF ./SF (5.5)
8. 아직/MAG 막히/VV 었/EP 을지/EC 도/JX 모르/VV 겠/EP 어/EF ./SF (5.5)
9. 너/NP 의/JKG 말/NNG 이/JKS 맞/VV 을지/EC 도/JX 모르/VV 아/EF ./SF (6.5)
```

```
10. 아니/IC 어쩌면/MAG 필연/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 아/EF ./SF (6.5)

  Word forms (except lexical items) + POS
1. 권태기/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4)
2. 「/SS 그렇/VA ㄹ는지/EC 도/JX 모르/VV 죠/EF ./SF (4)
3. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4)
4. 할미/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 죠/EF ./SF (4)
5. 시간/NNG 은/JX 더디/VA 게/EC 만/JX 흐르/VV 었/EP 다/EF ./SF (4)
6. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (4.5)
7. 정신/NNG 병원/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 아/EF ./SF (4.5)
8. 그렇/VA 기/EC ㄴ/JX 하/VV 지요/EF ./SF (5)
9. "/SS 그렇/VA 기/EC ㄴ/JX 하/VV 아요/EF ./SF (5)
10. 그렇/VA 게/EC 는/JX 하/VV 다/EF ./SF (5)
```

-(u)lo -(으)로

```
Data:
  (a) Query: 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB " ]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 48722

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. 바다/NNP 는/JX 자신/NNG 이/JKS 한국인/NNG 이/VCP 라고/EC 너무나/MAG 자랑/NNG 스럽/XSA 게/
    EC 말/NNG 하/XSV ㄴ/ETM 것/NNB 에/JKB 비하/VV 아/EC 한국말/NNG 이/JKS 서투르/VA 었/EP
    고/EC ,/SP 새우/NNG 가/JKS 자신/NNG 을/JKO 노려보/VV 는/ETM 것/NNB 같/VA 아/EC 새우젓/
    NNG 이/JKS 들어가/VV ㄴ/ETM 김치/NNG 는/JX 먹/VV 을/ETM 수/NNB 없/VA 다는/ETM 고백/NNG
     을/JKO 하/VV 며/EC 무안/NNG 하/XSA 아/EC 하/VX 는/ETM 표정/NNG 을/JKO 짓/VV 기/ETN
    도/JX 하/VX 았/EP 다/EF ./SF (0.970149253731)
2. 다른/MM 여자/NNG 랑/JKB 자/VV 는/ETM 것/NNB 보다/JKB 안/MAG 낫/VA 니/EF ?/SF (1.0)
3. 처음/NNG 에/JKB 오/VV ㄴ/ETM 사람/NNG 은/JX 비닐하우스촌/NNG 에서/JKB 데려오/VV ㄴ/ETM 남
    자/NNG 아이/NNG 이/VCP 었/EP 고/EC ,/SP 두/MM 번/NNB 째/XSN 는/JX 귀/NNG 가/JKS 전혀/
    MAG 들리/VV 지/EC 않/VX 는/ETM 할아버지/NNG 이/VCP 었/EP 다/EF ./SF (1.0)
4. 말/NNG 은/JX 자주/MAG 생각/NNG 으로부터/JKB 의/JKG 독립/NNG 을/JKO 꿈꾸/VV ㄴ다/EF ./SF
    (1.0)
5. 그리/MAG 하/XSV 어도/EC 그중/NNG 에/JKB 화롯불/NNG 을/JKO 가져오/VV ㄴ/ETM 계집애/NNG 만/
    JX 은/JX 저희/NP 들/XSN 축/NNB 에서/JKB 좀/MAG 쫄리/VV 어/EC 지내/VV 는지/EC 한풀/NNG
    이/JKS 죽/VV 어서/EC 떠들/VV 는/ETM 꼴/NNG 만/JX 웃/VV 으며/EC 가만히/MAG 바라보/VV 고/
    EC 앉/VV 았/EP 다/EF ./SF (1.0)
6. 어젯밤/NNG 에/JKB 언뜻/MAG 지나치/VA 면서/EC 보/VX 았/EP 을/ETM 때/NNG 는/JX 한/MM 삼천/
    NR 평/NNB 쯤/XSN 되/VV 는/ETM 줄/NNB 알/VV 았/EP 는데/EC ,/SP 낮/NNG 에/JKB 보/VV
    니/EC 훨씬/MAG 더/MAG 넓/VA 은/ETM 것/NNB 같/VA 다/EF ./SF (1.0)
7. 나/NP 의/JKG 집/NNG 에서/JKB 아이/NNG 들/XSN 둘/NR 이/JKS 잠자/VV 고/EC 있/VX 다/EF ./
    SF (1.0)
8. 어머니/NNG 가/JKS 아버지/NNG 의/JKG 선산/NNG 에/JKB 묻히/VV 던/ETM 날/NNG 에/JKB 도/JX
    아버지/NNG 는/JX 손수건/NNG 을/JKO 꺼내/VV 어/EC 눈가/NNG 를/JKO 닦/VV 았/EP 다/EF ./
    SF (1.0)
9. 창/NNG 쪽/NNB 에/JKB 붙/VV 어/EC 있/VX 는/ETM 3/SN 인/NNG 용/XSN 소파/NNG 위/NNG 에/
    JKB 여자/NNG 가/JKS 있/VV 었/EP 다/EF ./SF (1.0)
10. 아무리/MAG 상상/NNG 은/JX 자유/NNG 이/VCP 라지만/EC 로라/NNP 는/JX 자네/NP 하고/JKB 너
    무/MAG 멀리/MAG 있/VV 어/EF ./SF (1.0)

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (0.666666666667)
```

```
2. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (0.666666666667)
3. 아이/NNG 는/JX 학교/NNG 에/JKB 들어가/VV 았/EP 을까/EF ?/SF (0.692307692308)
4. 효과/NNG 는/JX 뜻밖/NNG 으로/JKB 크/VA 었/EP 다/EF ./SF (0.692307692308)
5. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (0.692307692308)
6. 언니/NNG 는/JX 집/NNG 에/JKB 없/VA 어/EF ./SF "/SS (0.692307692308)
7. 피/NNG 는/JX 흙/NNG 속/NNG 으로/JKB 스미/VV ㄴ다/EF ./SF (0.692307692308)
8. 먹이/NNG 는/JX 다음/NNG 에/JKB 주/VV 자/EF ./SF "/SS (0.692307692308)
9. 니코틴/NNG 냄새/NNG 는/JX 예상/NNG 보다/JKB 심하/VA 았/EP 다/EF ./SF (0.714285714286)
10. 그것/NP 으로/JKB 수사/NNG 는/JX 끝/NNG 이/VCP 었/EP 다/EF ./SF (0.714285714286)

10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. 신청서/NNG 는/JX 여기/NP 에/JKB 있/VV 습니다/EF ./SF (5.5)
2. 나/NP 는/JX 아파트/NNG 에서/JKB 살/VV ㄴ다/EF ./SF (5.5)
3. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (5.5)
4. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (5.5)
5. 그/NP 는/JX 어디/NP 에/JKB 있/VV 을까/EF ./SF (5.5)
6. 적어도/MAG 나/NP 한테/JKB 는/JX ./SF (6)
7. 상상/NNG 으로/JKB 말/NNG 이/VCP 다/EF ./SF (6)
8. 강이/NNP 에게/JKB 조차/JX 도/JX ./SF (6)
9. 특히/MAG 사무실/NNG 에서/JKB ./SF (6)
10. 마음/NNG 에/JKB 안/MAG 들/VV 어요/EF ./SF (6)

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (2)
2. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (2)
3. 담뱃가게/NNG 옆/NNG 에/JKB 있/VV 잖아요/EF ./SF (3)
4. 밖/NNG 에/JKB 는/JX 비/NNG 가/JKS 오/VV 아/EF ./SF (3)
5. 사람/NNG 에게/JKB 점수/NNG 를/JKO 매기/VV 다니/EF ./SF (3)
6. 신청서/NNG 는/JX 여기/NP 에/JKB 있/VV 습니다/EF ./SF (3)
7. 취중/NNG 에/JKB 택시/NNG 를/JKO 잡/VV 다가/EF ./SF (3)
8. 큰오빠/NNG 한테/JKB 여자/NNG 가/JKS 있/VV 대/EF ./SF (3)
9. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (3)
10. 낙엽/NNG 들/XSN 비/NNG 에/JKB 젖/VV 는다/EF ./SF (3)
```

C.3 Similarity Measures

C.3.1 Mode 1 – Default

```
Data:
  (a) Query: 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 9666

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (0.384615384615)
2. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (0.384615384615)
3. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (0.384615384615)
4. 피/NNG 는/JX 머리/NNG 에서/JKB 얼굴/NNG 로/JKB 흘러내리/VV 었/EP 다/EF ./SF
    (0.466666666667)
5. .../SE .../SE 발톱/NNG 의/JKG 길이/NNG 는/JX 얼마/NNG 로/JKB 하/VV ㄹ까/EF ?/SF (0.5)
```

```
6. 강도/NNG 살인/NNG 혐의/NNG 는/JX 조사/NNG 과정/NNG 에서/JKB 과실/NNG 치사/NNG 로/JKB 바꿔/VV 었/EP 습니다/EF ./SF (0.5)
7. 밥/NNG 어미/NNG 는/JX 구덩이/NNG 로/JKB 내려가/VV 지/EC 않/VX 았/EP 다/EF ./SF (0.5)
8. 남자/NNG 는/JX 마루/NNG 로/JKB 올라가/VV 아서/EC 아기/NNG 를/JKO 안/VV 았/EP 다/EF ./SF (0.529411764706)
9. 비/NNG 는/JX 진눈깨비/NNG 로/JKB 변하/VV 아/EC 가/VX 고/EC 있/VX 었/EP 다/EF ./SF (0.529411764706)
10. 어머니/NNG 는/JX 머리/NNG 를/JKO 젓/VV 으며/EC 뒤/NNG 로/JKB 물러앉/VV 았/EP 다/EF ./SF (0.529411764706)

10 most similar sentences according to the Jaccard/Dice distance using unigrams only:

  Word forms (except lexical items) + POS
1. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (0.375)
2. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (0.375)
3. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (0.375)
4. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (0.428571428571)
5. 펑크/NNG 나/VV ㄴ/ETM 혼/NNG 사이/NNG 로/JKB ./SF (0.428571428571)
6. 모래/NNG 로/JKB 만들/VV ㄴ/ETM 벽/NNG ./SF (0.428571428571)
7. 침실/NNG 로/JKB 가/VV ㄴ다/EF ./SF (0.428571428571)
8. 인문/NNG 학부/NNG 로/JKB 걸어가/VV 는/ETM 길/NNG ./SF (0.428571428571)
9. 선장실/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (0.428571428571)
10. 강도/NNG 살인/NNG 혐의/NNG 는/JX 조사/NNG 과정/NNG 에서/JKB 과실/NNG 치사/NNG 로/JKB 바꿔/VV 었/EP 습니다/EF ./SF (0.444444444444)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 테/NNG 를/JKO 나무/NNG 로/JKB 두르/VV ㄴ/ETM ./SF (2)
2. 이/MM 차/NNG 는/JX 하늘/NNG 로/JKB 올라가/VV ㅂ니다/EF ./SF (2)
3. 거지/NNG 는/JX 뒤/NNG 로/JKB 나자빠지/VV 었/EP 다/EF ./SF (2)
4. 여자/NNG 는/JX 개찰구/NNG 로/JKB 뛰어나가/VV 았/EP 다/EF ./SF (2)
5. 붉/VA 은/ETM 피/NNG 로/JKB 쓰/VV ㄴ/ETM ./SF (3)
6. 복도/NNG 로/JKB 나서/VV ㄴ다/EF ./SF (3)
7. 자기/NP 자리/NNG 로/JKB 올라가/VV ㄴ다/EF ./SF (3)
8. 꼬리/NNG 도/JX 볼멘소리/NNG 로/JKB 투덜거리/VV 었/EP 습니다/EF ./SF (3)
9. 출입문/NNG 도/JX 위아래/NNG 로/JKB 덜거덕거리/VV 었/EP 다/EF ./SF (3)
10. 한참/NNG 걷/VV 어서/EC 기관실/NNG 로/JKB 가/VV ㄴ다/EF ./SF (3)
```

C.3.2 Mode 2 – Distributional Analysis

```
Data:
  (a) Query: 김치/NNG 는/JX 배추/NNG 로/JKB 만들/VV ㅂ니다/EF ./SF
  (b) Target: [ " 로/JKB " ]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 48722

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (0.666666666667)
2. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (0.666666666667)
3. 아이/NNG 는/JX 학교/NNG 에/JKB 들어가/VV 았/EP 을까/EF ?/SF (0.692307692308)
4. 효과/NNG 는/JX 뜻밖/NNG 으로/JKB 크/VA 었/EP 다/EF ./SF (0.692307692308)
5. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (0.692307692308)
6. 언니/NNG 는/JX 집/NNG 에/JKB 없/VA 어/EF ./SF "/SS (0.692307692308)
7. 피/NNG 는/JX 흙/NNG 속/NNG 으로/JKB 스미/VV ㄴ다/EF ./SF (0.692307692308)
8. 먹이/NNG 는/JX 다음/NNG 에/JKB 주/VV 자/EF ./SF "/SS (0.692307692308)
9. 니코틴/NNG 냄새/NNG 는/JX 예상/NNG 보다/JKB 심하/VA 았/EP 다/EF ./SF (0.714285714286)
10. 그것/NP 으로/JKB 수사/NNG 는/JX 끝/NNG 이/VCP 었/EP 다/EF ./SF (0.714285714286)

10 most similar sentences according to the Jaccard/Dice distance using unigrams
    only:

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (0.5)
2. 피/NNG 는/JX 흙/NNG 속/NNG 으로/JKB 스미/VV ㄴ다/EF ./SF (0.5)
3. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (0.5)
4. 밖/NNG 에/JKB 는/JX 비/NNG 가/JKS 오/VV 아/EF ./SF (0.555555555556)
5. 신청서/NNG 는/JX 여기/NP 에/JKB 있/VV 습니다/EF ./SF (0.555555555556)
6. 끈기/NNG 는/JX 신념/NNG 을/JKO 바탕/NNG 으로/JKB 하/VV ㄴ다/EF ./SF (0.555555555556)
7. '/SS 학교/NNG 에/JKB 는/JX 다니/VV ㄹ까/EF ./SF '/SS (0.5555555555556)
8. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (0.555555555556)
9. 나/NP 는/JX 아파트/NNG 에서/JKB 살/VV ㄴ다/EF ./SF (0.555555555556)
10. 밥/NNG 어미/NNG 는/JX 방/NNG 으로/JKB 들어가/VV 았/EP 다/EF ./SF (0.555555555556)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 역사/NNG 는/JX 소걸음/NNG 으로/JKB 움직이/VV ㄴ다/EF ./SF (2)
2. 남자/NNG 는/JX 형/NNG 에게/JKB 묻/VV 는다/EF ./SF (2)
3. 담뱃가게/NNG 옆/NNG 에/JKB 있/VV 잖아요/EF ./SF (3)
4. 밖/NNG 에/JKB 는/JX 비/NNG 가/JKS 오/VV 아/EF ./SF (3)
5. 사람/NNG 에게/JKB 점수/NNG 를/JKO 매기/VV 다니/EF ./SF (3)
6. 신청서/NNG 는/JX 여기/NP 에/JKB 있/VV 습니다/EF ./SF (3)
7. 취중/NNG 에/JKB 택시/NNG 를/JKO 잡/VV 다가/EF ./SF (3)
8. 큰오빠/NNG 한테/JKB 여자/NNG 가/JKS 있/VV 대/EF ./SF (3)
9. 엄마/NNG 는/JX 유방암/NNG 에/JKB 걸리/VV 었/EP 다/EF ./SF (3)
10. 낙엽/NNG 들/XSN 비/NNG 에/JKB 젖/VV 는다/EF ./SF (3)
```

C.4 Genres

C.4.1 Mode 1 – Default

Book

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 259

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 그러면/MAJ 조금/NNG 은/JX 쉽/VA 어/EC 지/VX ㄹ지/EC 도/JX 모르/VV 아/EC 하/VV 고/EC 말/
    NNG 이/VCP 야/EF ./SF (0.727272727273)
2. 어쩌면/MAG 남편/NNG 은/JX 저/NP 를/JKO 죽이/VV ㄹ지/EC 도/JX 모르/VV 아요/EF ./SF
    (0.764705882353)
3. 변명/NNG 은/JX 진실/NNG 의/JKG 다른/MM 얼굴/NNG 이/VCP ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF
    (0.777777777778)
4. 게다가/MAG 타협/NNG 은/JX 생각/NNG 보다/JKB 길/VA 어/EC 지/VX ㄹ지/EC 도/JX 모르/ㄴ다VV/
    EF ./SF (0.789473684211)
5. 아니/IC 어쩌면/MAG 한순간/NNG 의/JKG 실수/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV 아/EC ,/SP 사
    랑/NNG 은/JX ./SF (0.8)
6. 그것/ㄴNP/JX 그렇/VA 게/EC 중요/NNG 하/ㄴXSA/ETM 사실/NNG 은/JX 아니/VCN ㄹ지/EC 도/JX
    모르/ㄴ다VV/EF ./SF (0.809523809524)
7. 미치/VV ㄹ지/EC 도/JX 모르/VV 아/EC (0.818181818182)
8. 이곳/NP 비/ㄴVV/ETM 공장/NNG 은/JX 어쩌면/MAG 나/NP 의/JKG 도약/NNG 의/JKG 발판/NNG 이/
    JKC 되/VV ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (0.826086956522)
9. 그렇/VA ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (0.833333333333)
10. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 지/EF ./SF (0.846153846154)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 지/EF ./SF (4)
2. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아/EF ./SF (4)
3. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 아요/EF ./SF (4)
4. 그렇/VA ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (4)
5. 어쩌면/MAG 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 습니다/EF ./SF (5)
6. "/SS 그래/IC 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (5)
7. "/SS 그렇/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 군요/EF ./SF (5)
8. 괜히/MAG 화풀이/NNG 당하/XSV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (5)
9. 혹시/MAG 차/NNG 가/JKS 오/VV ㄹ지/EC 도/JX 모르/VV 니까요/EF ./SF (5.5)
10. 그것/ㄴNP/JX 사실/NNG 이/VCP ㄹ지/EC 도/JX 모르/ㄴ다VV/EF ./SF (6)
```

Journal

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
```

```
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 56

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 그렇/VA ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.833333333333)
2. 어쩌면/MAG 35/SN 억/NR 년/NNB 동안/NNG 무사히/MAG 자라/VV 아/EC 오/VX ㄴ/ETM 온/MM 생명/
    NNG 은/JX 몇/MM 백/NR 년/NNB 내/NNB 에/JKB 멸망/NNG 하/XSV ㄹ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (0.866666666667)
3. 그것/NP 이/JKS 외압/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (0.875)
4. 이대로/MAG 이/VCP 라면/EC 재미/NNG 가/JKS 덜/MAG 하/XSA ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./
    SF (0.888888888889)
5. 또한/MAG 허상/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다고/EC 생각/NNG 하/XSV 았/EP 다/EF ./
    SF (0.888888888889)
6. 아름답/VA ㅁ/ETN 에서/JKB 이란/JX 어쩌면/MAG 그리움/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (0.888888888889)
7. 다시/MAG 들어가/VV 고/EC 싶/VX 은/ETM 마음/NNG 이/JKS 생기/VV ㄹ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (0.894736842105)
8. 주인공/NNG 플라즈모/NNP 의/JKG 천진난만/NNG 하/XSA ㅁ/ETN 은/JX 어리/VA ㄹ/ETM 때/NNG 인
    상/NNG 적/XSN 으로/JKB 보/VV 았/EP 던/ETM </SS 아스트로/NNP 보이/NNG >/SS 에서/JKB 영
    향/NNG 받/VV 은/ETM 바/NNB 가/JKS 크/VA ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF
    (0.897435897436)
9. 어쩜/MAG 이/MM 첫/MM 목록/NNG 은/JX 부재/NNG 하/XSV 았/EP 던/ETM 자신/NNG 의/JKG 목소리/
    NNG 를/JKO 내/VV 는/ETM 이/NP 들/XSN 의/JKG 열정/NNG 을/JKO 담/VV 은/ETM 것/NNB 으로/
    JKB 만족/NNG 하/XSA 아야/EC 하/VX 는/ETM 작품집/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (0.90243902439)
10. 혹시/MAG 꿈/NNG 에/JKB 그리/VV 던/ETM 사랑/NNG 과/JKB 접속/NNG 하/XSV ㄹ지/EC 도/JX 모
    르/VV 니까요/EF ./SF "/SS (0.904761904762)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 그렇/VA ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4)
2. 그것/NP 이/JKS 외압/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (7)
3. 이대로/MAG 이/VCP 라면/EC 재미/NNG 가/JKS 덜/MAG 하/XSA ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./
    SF (8.5)
4. 또한/MAG 허상/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다고/EC 생각/NNG 하/XSV 았/EP 다/EF ./
    SF (9)
5. 아름답/VA ㅁ/ETN 에서/JKB 이란/JX 어쩌면/MAG 그리움/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (9)
6. 다시/MAG 들어가/VV 고/EC 싶/VX 은/ETM 마음/NNG 이/JKS 생기/VV ㄹ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (9.5)
7. 전신/NNG 마비/NNG 같/VA 은/ETM 합병증/NNG 이/JKS 오/VV ㄹ지/EC 도/JX 모르/VV ㄹ/ETM 일/
    NNG 이/VCP 었/EP 다/EF ./SF (11)
8. 혹시/MAG 꿈/NNG 에/JKB 그리/VV 던/ETM 사랑/NNG 과/JKB 접속/NNG 하/XSV ㄹ지/EC 도/JX 모
    르/VV 니까요/EF ./SF "/SS (11.5)
9. 당신/NP 은/JX 나/NP 의/JKG 기분/NNG 이/JKS 어떻/VA ㄹ지/EC 조금/NNG 도/JX 헤아리/VV ㄹ/
    ETM 줄/NNB 모르/VV 는군요/EF ./SF "/SS (12)
10. 대학/NNG 동창/NNG 이/VCP 니/EC 여고/NNG 동창/NNG 과/JKB 다르/VA 아서/EC 좀/MAG 어색/XR
    하/XSA ㄹ지/EC 도/JX 모르/VV 았/EP 다/EF ./SF (13)
```

Newspaper

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
```

```
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 1 - same words in similar contexts

Number of matched sentences: 42

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 좌고우면/NNG 하/XSV 다가/EC 는/JX 귀중/XR 하/XSA ㄴ/ETM 시간/NNG 을/JKO 놓치/VV 어/EC 민
    심/NNG 은/JX 멀리/MAG 떠나/VV ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.846153846154)
2. 모든/MM 성장/NNG 은/JX 결국/NNG 자신/NNG 이/JKS 기대/VV 고/EC ,/SP 자신/NNG 을/JKO 억/
    VV 누르/VV 어/EC 오/VX ㄴ/ETM 것/NNB 들/XSN 에/JKB 대하/VV ㄴ/ETM 배반/NNG 이/VCP ㄹ
    지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.882352941176)
3. 뒷날/NNG 청문회/NNG 에/JKB 서/VV ㄹ지/EC 도/JX 모르/VV ㄴ다는/ETM 마음가짐/NNG 은/JX 관리/
    NNG 들/XSN 한테/JKB 한결/MAG 깨끗/XR 하/XSA 고/EC 떳떳/XR 하/XSA ㄴ/ETM 일/NNG 처리/
    NNG 를/JKO 주문/NNG 하/XSV ㄹ/ETM 것/NNB 이/VCP 다/EF ./SF (0.885714285714)
4. "/SS 박정희/NNP 기념관/NNG 을/JKO 짓/VV 는/ETM 데/NNB 국고/NNG 를/JKO 지원/NNG 하/XSV
    ㄴ다/EC "/SS 는/JX '/SS 슬프/VA ㄴ/ETM 대한민국/NNP '/SS 의/JKG 현실/NNG 은/JX 빈곤/
    NNG 하/XSA ㄴ/ETM 현대사/NNG 연구/NNG 의/JKG 결과물/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ
    다/EF ./SF (0.9)
5. 그리고/MAJ '/SS 음반/NNG 시장/NNG 활황/NNG '/SS 까지/JX 만들/VV 어/EC 내/VX ㄹ지/EC 도/
    JX 모르/VV ㄴ다/EF ./SF (0.9)
6. 이/NP 처럼/JKB 협상/NNG 이/JKS 지지부진/NNG 하/XSV ㄴ/ETM 채/NNB 로/JKB 시간/NNG 만/JX
    끌/VV 게/EC 되/VV ㄹ/ETM 경우/NNG 야/NNG 권/XSN 통합/NNG 은/JX 실패/NNG 하/XSV ㄹ지/EC
    도/JX 모르/VV ㄴ다는/ETM 우려/NNG 가/JKS 점점/MAG 짙/VA 어/EC 가/VX 는/ETM 시점/NNG
    이/VCP 다/EF ./SF (0.904761904762)
7. 연애/NNG 열풍/NNG 과/JC 섹스/NNG 에/JKB 대하/VV ㄴ/ETM 집착/NNG 이/JKS 인간관계/NNG 의/
    JKG 모든/MM 것/NNB 으로/JKB 얘기/NNG 되/XSV 는/ETM 오늘날/NNG ,/SP 새롭/VA ㄴ/ETM 관
    계/NNG 의/JKG 가능/NNG 성/XSN 에/JKB 대하/VV ㄴ/ETM 질문/NNG 은/JX 이제/NNG 부터/JX 시
    작/NNG 되/XSV 는/ETM 것/NNB 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다는/ETM 생각/NNG 을/JKO
    하/VV 아/EC 보/VX ㄴ다/EF ./SF (0.914893617021)
8. 서파티/NNP 는/JX 유럽/NNP 주둔/NNG 나토/NNP 병력/NNG 은/JX 2/SN 백/NR 10/SN 만/NR 명/
    NNB 에서/JKB 1/SN 백/NR 30/SN 만/NR 명/NNB 으로/JKB 감축/NNG 되/XSV ㄹ/ETM 것/NNB
    이/VCP 며/EC 동유럽/NNP 병력/NNG 은/JX 이/NP 보다/JKB 훨씬/MAG 더/MAG 낮/VA 은/ETM 수
    준/NNG 으로/JKB 줄어들/VV ㄹ지/EC 도/JX 모르/VV ㄴ다고/EC 내다보/VV 았/EP 다/EF ./SF
    (0.914893617021)
9. 베컴/NNP 과/JC 김남일/NNP 을/JKO 한/MM 광고/NNG 에서/JKB 보/VV 는/ETM 날/NNG 이/JKS 오/
    VV ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (0.916666666667)
10. "/SS 동네/NNG 의원/NNG 에서/JKB 우연히/MAG 자궁암/NNG 검사/NNG 를/JKO 받/VV 았/EP 는데/
    EC 혹시/MAG 암/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다고/EC 하/VV 아서요/EF ./SF "/SS
    (0.925925925926)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 그리고/MAJ '/SS 음반/NNG 시장/NNG 활황/NNG '/SS 까지/JX 만들/VV 어/EC 내/VX ㄹ지/EC 도/
    JX 모르/VV ㄴ다/EF ./SF (11.5)
2. 베컴/NNP 과/JC 김남일/NNP 을/JKO 한/MM 광고/NNG 에서/JKB 보/VV 는/ETM 날/NNG 이/JKS 오/
    VV ㄹ지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (14.5)
3. 좌고우면/NNG 하/XSV 다가/EC 는/JX 귀중/XR 하/XSA ㄴ/ETM 시간/NNG 을/JKO 놓치/VV 어/EC 민
    심/NNG 은/JX 멀리/MAG 떠나/VV ㄹ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (15.5)
4. "/SS 동네/NNG 의원/NNG 에서/JKB 우연히/MAG 자궁암/NNG 검사/NNG 를/JKO 받/VV 았/EP 는데/
    EC 혹시/MAG 암/NNG 이/VCP ㄹ지/EC 도/JX 모르/VV ㄴ다고/EC 하/VV 아서요/EF ./SF "/SS
    (18.5)
5. 이/MM 후보/NNG 의/JKG 지지율/NNG 정체/NNG 로/JKB 새롭/VA 게/EC 불거지/VV ㄹ지/EC 도/JX 모
    르/VV ㄹ/ETM 한나라당/NNP 내부/NNG 의/JKG 틈새/NNG 와/JC 분열/NNG 을/JKO 노리/VV ㄴ/ETM
    것/NNB 이/VCP 다/EF ./SF (21)
6. 카리스마/NNG 가득/MAG 넘치/VV 는/ETM 목소리/NNG 로/JKB 그동안/NNG 의/JKG 구조/NNG 적/XSN
    이/VCP ㄴ/ETM 문제/NNG 들/XSN 을/JKO 한꺼번에/MAG 혁파/NNG 하/XSV 아/EC 내/VX ㄹ지/EC
    도/JX 모르/VV ㄴ다/EF ./SF (21.5)
7. '/SS 문민/NNG 독재/NNG '/SS 를/JKO 강화/NNG 하/XSV 는/ETM 방편/NNG 으로/JKB 검찰/NNG 인
    사/NNG 를/JKO 단행/NNG 하/XSV ㄹ지/EC 도/JX 모르/VV ㄴ다는/ETM 우려/NNG 가/JKS 그것/NP
```

```
    이/VCP 다/EF ./SF (21.5)
8. 그것/NP 은/JX 우리/NP 가/JKS 어떻/VA 게/EC 채우/VV 어야/EC 하/VX ㄹ지/EC 도/JX 모르/VV
    는/ETM 공간/NNG 이나/JC 시간/NNG 으로/JKB 이루/VV ㄹ/ETM 수/NNB 있/VV 는/ETM 것/NNB
    이/JKC 아니/VCN 다/EF ./SF (22)
9. 어차피/MAG 받/VV 아/EC 들이/VV ㄹ/ETM 일/NNG 을/JKO 지금/NNG 까지/JX 버티/VV 어/EC 오/VX
    ㄴ/ETM 데/NNB 대하/VV 아/EC 내심/NNG 불쾌/XR 하/XSA 게/EC 여기/VV ㄹ지/EC 도/JX 모르/
    VV ㄴ다/EF ./SF (22.5)
10. 모든/MM 성장/NNG 은/JX 결국/NNG 자신/NNG 이/JKS 기대/VV 고/EC ,/SP 자신/NNG 을/JKO 억/
    VV 누르/VV 어/EC 오/VX ㄴ/ETM 것/NNB 들/XSN 에/JKB 대하/VV ㄴ/ETM 배반/NNG 이/VCP ㄹ
    지/EC 도/JX 모르/VV ㄴ다/EF ./SF (23.5)
```

C.4.2 Mode 2 – Distributional Analysis

Book

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 1520

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms + POS
1. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.714285714286)
2. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (0.733333333333)
3. 아직/MAG 막히/VV 었/EP 을지/EC 도/JX 모르/VV 겠/EP 어/EF ./SF (0.733333333333)
4. 너/NP 의/JKG 말/NNG 이/JKS 맞/VV 을지/EC 도/JX 모르/VV 아/EF ./SF (0.75)
5. "/SS 오늘/NNG 은/JX 좀/MAG 늦/VV 을지/EC 도/JX 모르/VV ㄹ걸/EF !/SF (0.75)
6. 아마/MAG 울/VV 고/EC 있/VX 을지/EC 도/JX 모르/VV 았/EP 다/EF ./SF (0.75)
7. 그리고/MAJ 결국/NNG 에/JKB ㄴ/JX 빈손/NNG 만/JX 남/VV 을지/EC 도/JX 모르/VV ㄴ다구/EF ./
    SF (0.777777777778)
8. "/SS 그래서/MAJ ,/SP 통일감/NNG 이/JKS 떨어지/VV 었/EP 을지/EC 도/JX 모르/VV 아요/EF ./
    SF (0.777777777778)
9. 그녀/NP 는/JX 어쩌면/MAG 집/NNG 에/JKB 내려가/VV 았/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./
    SF (0.777777777778)
10. 어젯밤/NNG 나/NP 에게/JKB 전화/NNG 를/JKO 하/VV 았/EP 을지/EC 도/JX 모르/VV 겠/EP 다/EF
    ./SF (0.789473684211)

  Word forms (except lexical items) + POS
1. "/SS 오늘/NNG 은/JX 좀/MAG 늦/VV 을지/EC 도/JX 모르/VV ㄹ걸/EF !/SF (0.625)
2. 어쩌면/MAG 다시/MAG 박송미/NNP 에게/JKB 연락/NNG 하/XSV ㄹ/ETM 수/NNB 없/VA 을지/EC 도/
    JX 모르/VV 았/EP 다/EF ./SF (0.714285714286)
3. "/SS 뭐/IC ,/SP 더/MAG 이상/NNG 상종/NNG 하/XSV ㄹ/ETM 필요/NNG 없/VA 을지/EC 도/JX 모
    르/VV 아요/EF ./SF (0.714285714286)
4. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.714285714286)
5. 어쩌면/MAG 우리/NP 가/JKS 영영/MAG 헤어지/VV ㄹ/ETM 수/NNB 없/VA 을지/EC 도/JX 모르/VV ㄴ
    다는/ETM 예감/NNG 때문/NNB 에/JKB ?/SF (0.727272727273)
6. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (0.733333333333)
7. 시간/NNG 은/JX 더디/VA 게/EC 만/JX 흐르/VV 었/EP 다/EF ./SF (0.733333333333)
8. 아직/MAG 막히/VV 었/EP 을지/EC 도/JX 모르/VV 겠/EP 어/EF ./SF (0.733333333333)
9. 석고/NNG 를/JKO 떼/VV 어/EC 내/VX ㄹ/ETM 때/NNG 도/JX 아프/VA ㄹ/ETM 터/NNB 이/VCP 고/
    EC .../SE .../SE 하지만/MAJ 맨/XPN 얼굴/NNG 이/JKC 아니/VCN 니까/EC ,/SP 조금/NNG 은/
    JX 낫/VA 을지/EC 도/JX 모르/VV 겠/EP 습니다/EF ./SF "/SS (0.736842105263)
10. 그러나/MAJ 그/NP 는/JX 그것/NP 이/JKS 차라리/MAG 낫/VA 을지/EC 도/JX 모르/VV ㄴ다는/ETM
    생각/NNG 을/JKO 하/VV 았/EP 다/EF ./SF (0.739130434783)

10 closest sentences using the Levenshtein distance:

  Word forms + POS
1. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4.5)
2. 권태기/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (5)
3. 「/SS 그렇/VA ㄹ는지/EC 도/JX 모르/VV 죠/EF ./SF (5)
4. 할미/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 죠/EF ./SF (5)
5. 그래서/MAJ 이/VCP 었/EP 는지/EC 도/JX 모르/VV ㄴ다/EF ./SF (5.5)
6. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (5.5)
7. 정신/NNG 병원/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 아/EF ./SF (5.5)
8. 아직/MAG 막히/VV 었/EP 을지/EC 도/JX 모르/VV 겠/EP 어/EF ./SF (5.5)
9. 너/NP 의/JKG 말/NNG 이/JKS 맞/VV 을지/EC 도/JX 모르/VV 아/EF ./SF (6.5)
```

```
10. 아니/IC 어쩌면/MAG 필연/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 아/EF ./SF (6.5)

  Word forms (except lexical items) + POS
1. 권태기/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4)
2. 「/SS 그렇/VA ㄹ는지/EC 도/JX 모르/VV 죠/EF ./SF (4)
3. 어쩌면/MAG 그렇/VA 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (4)
4. 할미/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 죠/EF ./SF (4)
5. 시간/NNG 은/JX 더디/VA 게/EC 만/JX 흐르/VV 었/EP 다/EF ./SF (4)
6. 「/SS 내출혈/NNG 이/JKS 있/VV 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF (4.5)
7. 정신/NNG 병원/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 아/EF ./SF (4.5)
8. 그렇/VA 기/EC ㄴ/JX 하/VV 지요/EF ./SF (5)
9. "/SS 그렇/VA 기/EC ㄴ/JX 하/VV 아요/EF ./SF (5)
10. 그렇/VA 게/EC 는/JX 하/VV 다/EF ./SF (5)
```

Journal

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 414

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. "/SS 다이아몬드/NNG 가/JKS 붙/VV 을지/EC 도/JX 모르/VV 지/EF ./SF "/SS (0.75)
2. 핀란드/NNP 의/JKG 색깔/NNG 은/JX 어쩌면/MAG 하얀색/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV ㄴ다/
    EF ./SF (0.777777777778)
3. 이/MM 문장/NNG 을/JKO 꼭/MAG 15/SN 세기/NNG 북구/NNP 유럽/NNP 풍/XSN 이/VCP 라고/EC 말/
    NNG 하/XSV ㄹ/ETM 수/NNB 는/JX 없/VA 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF
    (0.785714285714)
4. 어쩌면/MAG "/SS 미안/NNG 하/XSA 아/EC ,/SP 여보/IC "/SS 이/VCP 라던/ETM 아내/NNG 의/
    JKG 말/NNG 은/JX 진심/NNG 이/VCP 었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF
    (0.785714285714)
5. 돈/NNG 은/JX 없/VA 다가/EC 도/JX 벌/VV 면/EC 또/MAG 생기/VV 지만/EC ,/SP 한번/NNG 흘러
    가/VV ㄴ/ETM 시간/NNG 은/JX 다시/MAG 돌이키/VV ㄹ/ETM 수/NNB 없/VA 다/EF ./SF
    (0.785714285714)
6. 베풀/VV 고/EC 도/JX 잊어버리/VV ㄹ/ETM 수/NNB 있/VV 는/ETM 사람/NNG 은/JX 행복/NNG 하/
    XSA 다/EF ./SF (0.8)
7. 화살표/NNG 가/JKS 지시/NNG 하/XSV 는/ETM 방향/NNG 을/JKO 그대/NP 가/JKS 따르/VV 지/EC
    않/VX 는다면/EC 그대/NP 는/JX 영원히/MAG 돌아오/VV ㄹ/ETM 수/NNB 없/VA 을지/EC 도/JX 모
    르/VV ㄴ다/EF ./SF (0.806451612903)
8. 윤희/NNP 언니/NNG 같/VA 으면/EC 얼마/NNG 이/VCP 든지/EC 그렇/VA ㄹ/ETM 수/NNB 있/VV 을
    지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.818181818182)
9. 나/NP 도/JX 서른/NR 살/NNB 이/JKS 넘/VV 었/EP 으면/EC 이런/MM 영화/NNG 못/MAG 만들/VV
    었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.826086956522)
10. 알리/NNP 족장/NNG 과/JKB 의/JKG 진하/VA ㄴ/ETM 우정/NNG 은/JX 동성애/NNG 를/JKO 암시/
    NNG 하/XSV 고/EC 도/JX 남/VV 는다/EF ./SF (0.826086956522)

10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 하긴/MAJ 그렇/VA 기/EC 도/JX 하/VV 겠/EP 네요/EF ./SF (5)
2. "/SS 다이아몬드/NNG 가/JKS 붙/VV 을지/EC 도/JX 모르/VV 지/EF ./SF "/SS (5.5)
3. 이제/MAG 오/VV 아서/EC 야/JX 깨닫/VV 는다/EF ./SF (6)
4. 끝내/MAG 망가뜨리/VV 고/EC 말/VV ㄹ는지/EC 도/JX 모르/VV ㄴ다/EF ./SF (6.5)
```

```
5. 기대/NNG 만큼/JKB 되/VV ㄹ지/EC 는/JX 모르/VV 겠/EP 지만/EF ./SF (6.5)
6. 끝내/MAG 망가뜨리/VV 고/EC 말/VX ㄹ는지/EC 도/JX 모르/VV ㄴ다/EF ./SF (6.5)
7. 서울/NNP 사람/NNG 만/JX 모이/VV 는지/EC 도/JX 모르/VV 았/EP 다/EF ./SF (6.5)
8. "/SS 나/NP 도/JX 늦/VA 게/EC 야/JX 알아차리/VV 었/EP 다/EF ./SF (7)
9. 그래서/MAJ 행복/NNG 이/JKS 값지/VA ㄴ/ETM 것/NNB 이/VCP ㄴ지/EC 도/JX 모르/VV ㄴ다/EF
   ./SF (7)
10. 영화/NNG 를/JKO 만들/VV 려면/EC 적어도/MAG 저렇/VA 게/EC 는/JX 만들/VV 어야지/EF ./SF
   (7)
```

Newspaper

```
Data:
  (a) Query: 내일/NNG 은/JX 맑/VA 을지/EC 도/JX 모르/VV ㅂ니다/EF ./SF
  (b) Target: [ " ㄹ지/ECD ", " 도/JX ", " 모르/VV "]
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 271

10 most similar sentences according to the Jaccard/Dice distance using bigrams (
    measure is Jaccard's):

  Word forms (except lexical items) + POS
1. 하마터면/MAG 우리/NP 항공/NNG 산업/NNG 은/JX 물론/MAG 나라/NNG 의/JKG 체면/NNG 과/JC 대
   외/NNG 신인/NNG 도/NNG 에/JKB 치명/NNG 적/XSN 이/VCP ㄴ/ETM 손상/NNG 을/JKO 입히/VV
   었/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.806451612903)
2. 청정기/NNG 방식/NNG 은/JX 크/VA 게/EC 필터식/NNG ,/SP 집진판식/NNG ,/SP 해파/NNG 필터식/
   NNG 등/NNB 으로/JKB 나뉘/VV 는데/EC 각기/MAG 장단점/NNG 이/JKS 있/VV 으므로/EC 어떤/MM
   방식/NNG 이/JKS 적합/NNG 하/XSA ㄴ지/EC 도/JX 따지/VV 어/EC 보/VX 아야/EC 하/VX ㄴ다/EF
   ./SF (0.833333333333)
3. 물론/MAG 전문/NNG 적/XSN 이/VCP ㄴ/ETM 분석/NNG 기법/NNG 을/JKO 동원/NNG 하/XSV ㄴ/ETM
   건설/NNG 부/NNG 의/JKG 이/NP 같/VA 은/ETM 진단/NNG 은/JX 상당/NNG 하/XSA ㄴ/ETM 타당/
   XR 성/XSN 을/JKO 갖/VV 고/EC 있/VX 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF
   (0.837837837838)
4. 그러나/MAJ 책/NNG 은/JX 어렵/VA ㄴ/ETM 말/NNG 은/JX 제치/VV 어/EC 놓/VX 고/EC 도/JX 보/
   VV ㄹ/ETM 수/NNB 있/VV 으므로/EC 나쓰메/NNP 소우세키/NNP 나/JC 도쿠도미/NNP 로우카/NNP
   등/NNB 의/JKG 소설/NNG ,/SP 잡문/NNG 등/NNB 을/JKO 다소/MAG 애독/NNG 하/XSV 았/EP 다/
   EF ./SF (0.846153846154)
5. 심야/NNG 의/JKG 인터뷰/NNG 가/JKS 끝나/VV ㄹ/ETM 때/NNG 쯤/XSN 그/NP 의/JKG 목/NNG 은/
   JX 신기/XR 하/XSA 게/EC 도/JX 풀리/VV 어/EC 있/VX 었/EP 다/EF ./SF (0.851851851852)
6. 이런/MM 점/NNG 에서/JKB 노조/NNG 의/JKG 정치/NNG 활동/NNG 은/JX 민주주의/NNG 실현/NNG 을/
   JKO 위하/VV ㄴ/ETM 기본/NNG 요건/NNG 이/VCP 라고/EC 도/JX 하/VV ㄹ/ETM 수/NNB 있/VV
   다/EF ./SF (0.857142857143)
7. 만약/NNG 이번/NNG 페르시아/NNP 만/NNG 위기/NNG 가/JKS 냉전/NNG 구조/NNG 아래/NNG 에서/JKB
   발생/NNG 하/XSV 았/EP 다면/EC 미/NNP -/SS 소/NNP 대립/NNG 으로/JKB 유엔/NNP 기능/NNG
   은/JX 마비/NNG 되/XSV 고/EC 이라크/NNP 행위/NNG 는/JX 기정/NNG 사실/NNG 화/XSN 되/XSV
   었/EP 거나/EC 미/NNP -/SS 소/NNP 대립/NNG 을/JKO 더욱/MAG 증폭/NNG 시키/XSV 는/ETM 계
   기/NNG 로/JKB 작용/NNG 하/XSV 았/EP 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF
   (0.872340425532)
8. 아니/MAG 어쩌면/MAG 비극/NNG 적/XSN 창조/NNG 의/JKG 포로/NNG 들/XSN 이/VCP ㄴ/ETM 서구
   인/NNG 들/XSN 에게/JKB 창조/NNG 의/JKG 비극/NNG 성/XSN 을/JKO 보/VV 기/ETN 란/JX 쉽/
   VA 지/EC 않/VX 을지/EC 도/JX 모르/VV ㄴ다/EF ./SF (0.875)
9. 오히려/MAG 잘/MAG 되/VV 었/EP 는지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (0.875)
10. 김영삼/NNP 대통령/NNG 은/JX 4/SN 일/NNB "/SS 우리/NP 는/JX 북한/NNP 에/JKB 대하/VV 아/
   EC 군사/NNG 적/XSN 으로/JKB 어떤/MM 경우/NNG 가/JKS 있/VV 을지/EC 도/JX 모르/VV ㄴ다는/
   ETM 우발/NNG 적/XSN 이/VCP ㄴ/ETM 상황/NNG 까지/JX 도/JX 고려/NNG 하/XSV 아/EC 철저히/
   MAG 대비/NNG 하/XSV 아야/EC 하/VX ㄴ다/EC "/SS 고/JKQ 강조/NNG 하/XSV 았/EP 다/EF ./
   SF (0.877551020408)
```

```
10 closest sentences using the Levenshtein distance:

  Word forms (except lexical items) + POS
1. 오히려/MAG 잘/MAG 되/VV 었/EP 는지/EC 도/JX 모르/VV 겠/EP 다/EF ./SF (7.5)
2. "/SS 고향/NNG 이/JKS 북쪽/NNG 이/VCP ㄴ지/EC 도/JX 모르/VV 지/EF ./SF "/SS (7.5)
3. 그것/NP 은/JX 또한/MAG 효율/NNG 적/XSN 이/VCP 지/EC 도/JX 않/VV 다/EF ./SF (7.5)
4. 북한/NNP 의/JKG 이런/MM 회담/NNG 자세/NNG 가/JKS 낯설/VA 지/EC 는/JX 않/VV 다/EF ./SF
   (8)
5. 하지만/MAJ 그것/NP 은/JX 역설/NNG 이/JKC 아니/VCN 었/EP 는지/EC 도/JX 모르/VV ㄴ다/EF ./
   SF (8.5)
6. 아니/IC 그것/NP 이/JKS 바로/MAG 우리/NP 인간/NNG 의/JKG 참모습/NNG 이/VCP ㄴ지/EC 도/JX
   모르/VV ㄴ다/EF ./SF (10.5)
7. 그렇/VA 어야/EC 만/JX 살아남/VV 을/ETM 수/NNB 있/VV 기/ETN 때문/NNB 이/VCP 다/EF ./SF
   (10.5)
8. 국회/NNG 의/JKG 특위/NNG 는/JX 지방/NNG 자치/NNG 의/JKG 구현/NNG 을/JKO 다루/VV 리라고/
   EC 도/JX 하/VV ㄴ다/EF ./SF (11.5)
9. 그렇/VA 다고/EC 인간/NNG 의/JKG 삶/NNG 이/JKS 그리/MAG 단순히/MAG 흡수/NNG 용해/NNG 되/
   XSV 지/EC 도/JX 않/VV 는다/EF ./SF (12.5)
10. 마음/NNG 약하/VA ㄴ/ETM 정식/NNP 은/JX 어떻/VA 게/EC 든/JX 하/VV 아/EC 보/VX 겠/EP 다
   고/EC 다시/MAG 약속/NNG 하/XSV ㄴ다/EF ./SF (13)
```

C.5 English

C.5.1 Mode 1 – Default

Had been V-ing

```
Data:
  (a) Query: I_PNP 'd_VHD been_VBN working_VVG hard_AV0 all_SENT day_NN1
  (b) Target: ["'d_VHD", 'been_VBN', 'working_VVG']
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 38986

10 most similar sentences according to Jaccard index/Dice coefficient using
    bigrams (measure is Jaccard's):

  Word forms + POS
1. I_PNP 'd_VHD forgotten_VVN all_DT0 about_PRP them_PNP (0.818181818182)
2. If_CJS only_AV0 I_PNP 'd_VHD been_VBN 10_CRD years_NN2 of_PRF age_NN1 again_AV0
     I_PNP 'd_VM0 have_VHI loved_VVN the_AT0 Pirates_NN2 Club_NN1 (0.818181818182)
3. How_AVQ did_VDD you_PNP know_VVI I_PNP 'd_VHD met_VVN him_PNP (0.846153846154)
4. I_PNP 'd_VHD never_AV0 thought_VVN of_PRF that_DT0 replied_VVD Mcduff_NP0
    (0.846153846154)
5. We_PNP 'd_VHD been_VBN in_PRP our_DPS first_ORD house_NN1 for_PRP just_AV0
    over_AV0 five_CRD years_NN2 and_CJC I_PNP 'd_VHD served_VVN my_DPS d-i-y_NN1
    apprenticeship_NN1 then_AV0 improving_VVG slightly_AV0 as_CJS each_DT0
    room_NN1 was_VBD restored_VVN-AJ0 and_CJC decorated_VVD-AJ0 (0.882352941176)
6. I_PNP jumped_VVD out_AVP of_PRF an_AT0 aeroplane_NN1 and_CJC discovered_VVD
    I_PNP 'd_VHD forgotten_VVN my_DPS parachute_NN1 (0.8888888888889)
7. If_CJS I_PNP 'd_VHD known_VVN what_DTQ you_PNP were_VBD going_VVG to_TO0 do_VDI
     I_PNP 'd_VM0 never_AV0 have_VHI caught_VVN him_PNP (0.904761904762)
8. Checking_VVG his_DPS watch_NN1 he_PNP found_VVD that_CJT he_PNP 'd_VHD been_VBN
     out_AVP of_PRF the_AT0 cabin_NN1 for_PRP a_AT0 little_AV0 over_PRP three_CRD
    hours_NN2 (0.916666666667)
```

```
9. We_PNP 'd_VHD been_VBN prepared_AJ0 to_TO0 buy_VVI houses_NN2 with_PRP
    flaws_NN2 invisible_AJ0 to_PRP the_AT0 naked_AJ0 eye_NN1 but_CJC now_AV0
    we_PNP 'd_VHD fallen_VVN for_PRP one_PNI-CRD with_PRP all_DT0 its_DPS
    flaws_NN2 only_AV0 too_AV0 obviously_AV0 visible_AJ0 (0.941176470588)
10. She_PNP wrote_VVD During_PRP the_AT0 past_AJ0 two_CRD years_NN2 I_PNP have_VHB
    been_VBN working_VVG with_PRP Mr_NP0 Wilmott_NP0 's_POS co-operation_NN1
    towards_PRP the_AT0 publication_NN1 of_PRF a_AT0 Flora_NN0 of_PRF the_AT0
    Outer_AJ0 Islands_NN2 concerning_PRP which_DTQ there_EX0 is_VBZ a_AT0
    considerable_AJ0 scattered_AJ0 literature_NN1 but_CJC no_AT0 comprehensive_AJ0
    publication_NN1 (0.953488372093)

  Word forms (except lexical items) + POS
1. You_PNP 've_VHB been_VBN behaving_VVG suspiciously_AV0 these_DT0 past_PRP
    few_DT0 weeks_NN2 (0.714285714286)
2. GETTING_VVG THERE_AV0 (0.714285714286)
3. So_AV0 Trent_NP0 had_VHD been_VBN correct_AJ0 (0.8)
4. I_PNP had_VHD on_PRP my_DPS shoes_NN2 (0.8)
5. I_PNP 've_VHB been_VBN watching_VVG you_PNP (0.8)
6. This_DT0 repair_NN1 and_CJC replacement_NN1 necessitated_VVD breaking_VVG
    away_AV0 the_AT0 concrete_NN1 into_PRP which_DTQ the_AT0 track_NN1 had_VHD
    been_VBN set_VVN (0.8)
7. Endill_NP0-NN1 had_VHD been_VBN punished_VVN also_AV0 (0.8)
8. Penal_AJ0 taxes_NN2 had_VHD been_VBN abolished_VVN (0.8)
9. If_CJS only_AV0 I_PNP 'd_VHD been_VBN 10_CRD years_NN2 of_PRF age_NN1 again_AV0
    I_PNP 'd_VM0 have_VHI loved_VVN the_AT0 Pirates_NN2 Club_NN1 (0.809523809524)
10. I_PNP wished_VVD again_AV0 that_CJT I_PNP had_VHD been_VBN at_PRP B.P._NP0
    with_PRP Angela_NP0 and_CJC Anne_NP0 and_CJC Wendy_NP0 and_CJC my_DPS
    other_AJ0 comrades_NN2 (0.809523809524)

10 closest sentences using Levenshtein's:

  Word forms + POS
1. I_PNP 've_VHB been_VBN watching_VVG you_PNP (5.0)
2. I_PNP 've_VHB been_VBN searching_VVG for_PRP ages_NN2 (5.0)
3. I_PNP been_VBN (5)
4. I_PNP 'd_VHD forgotten_VVN all_DT0 about_PRP them_PNP (5.0)
5. I_PNP need_VVB your_DPS help_NN1 (5.5)
6. I_PNP feel_VVB better_AJC already_AV0 (5.5)
7. I_PNP Miguelito_NP0-NN1 (5.5)
8. I_PNP can_VM0 tell_VVI (5.5)
9. I_PNP 've_VHB been_VBN looking_VVG for_PRP you_PNP everywhere_AV0 (5.5)
10. I_PNP 'm_VBB not_XX0 sure_AJ0 (5.5)

  Word forms (except lexical items) + POS
1. I_PNP 've_VHB been_VBN watching_VVG you_PNP (4.0)
2. I_PNP 've_VHB been_VBN looking_VVG for_PRP you_PNP everywhere_AV0 (4.0)
3. I_PNP 've_VHB been_VBN searching_VVG for_PRP ages_NN2 (4.0)
4. He_PNP had_VHD been_VBN correct_AJ0 about_PRP the_AT0 pain_NN1 (4.0)
5. cancelling_VVG your_DPS holiday_NN1 (4.5)
6. I_PNP need_VVB your_DPS help_NN1 (4.5)
7. ERECTING_VVG A_AT0 KIT_NN1 GARAGE_NN1 (4.5)
8. Trent_NP0 had_VHD no_AT0 option_NN1 (4.5)
9. Positioning_VVG the_AT0 steam_NN1 engine_NN1 (4.5)
10. Recognising_VVG and_CJC rewarding_AJ0 success_NN1 (4.5)
```

Like

```
Data:
  (a) Query: I_PNP left_VVD everything_PNI like_PRP it_PNP was_VBD
  (b) Target: ['like_PRP']
  (c) Mode: 2 - similar words (based on POS) in similar contexts

Number of matched sentences: 27271

10 most similar sentences according to Jaccard index/Dice coefficient using
    bigrams (measure is Jaccard's):

  Word forms + POS
1. Eventually_AV0 it_PNP was_VBD published_VVN in_PRP Nature_NN1 (0.8)
2. After_PRP dinner_NN1 it_PNP was_VBD Question_NN1 Time_NN1 (0.8)
3. For_PRP Trent_NP0 it_PNP was_VBD a_AT0 sizing-up_NN1-AJ0 process_NN1-VVB
    (0.818181818182)
4. In_PRP fact_NN1 it_PNP was_VBD sufficiently_AV0 sensitive_AJ0 for_PRP A.T_NP0
    (0.833333333333)
5. Then_AV0 it_PNP was_VBD time_NN1 for_PRP the_AT0 Club_NN1 cabaret_NN1
    (0.833333333333)
6. For_PRP those_DT0 times_NN2 it_PNP was_VBD an_AT0 environmentally_AV0
    sensitive_AJ0 organisation_NN1 (0.846153846154)
7. This_DT0 time_NN1 it_PNP was_VBD a_AT0 team_NN1 from_PRP Southampton_NP0
    University_NN1 (0.846153846154)
8. Indeed_AV0 it_PNP was_VBD cause_NN1 for_PRP pride_NN1 in_PRP this_DT0 area_NN1
    (0.846153846154)
9. On_PRP 24_CRD July_NP0 it_PNP was_VBD the_AT0 turn_NN1 of_PRF The_AT0 Times_NN2
    (0.857142857143)
10. However_AV0 it_PNP was_VBD now_AV0 all_DT0 systems_NN2 go_VVB for_PRP the_AT0
    future_NN1 (0.857142857143)

  Word forms (except lexical items) + POS
1. So_AV0 I_PNP applied_VVD for_PRP my_DPS discharge_NN1 and_CJC it_PNP was_VBD
    granted_VVN (0.714285714286)
2. When_CJS I_PNP arrived_VVD back_AVP at_PRP B.P._NP0 it_PNP was_VBD to_PRP
    further_AJC billeting_NN1 problems_NN2 (0.75)
3. I_PNP came_VVD with_PRP my_DPS pain_NN1 (0.777777777778)
4. I_PNP came_VVD to_PRP the_AT0 cathedral_NN1 (0.777777777778)
5. I_PNP loved_VVD a_AT0 man_NN1 with_PRP spirit_NN1 (0.8)
6. Eventually_AV0 it_PNP was_VBD published_VVN in_PRP Nature_NN1 (0.8)
7. Grabbing_VVG the_AT0 nettle_NN1 I_PNP went_VVD on_AVP-PRP (0.8)
8. After_PRP dinner_NN1 it_PNP was_VBD Question_NN1 Time_NN1 (0.8)
9. Then_AV0 I_PNP went_VVD back_AVP to_PRP work_NN1 (0.8)
10. For_PRP Trent_NP0 it_PNP was_VBD a_AT0 sizing-up_NN1-AJ0 process_NN1-VVB
    (0.818181818182)

10 closest sentences using Levenshtein's:

  Word forms + POS
1. Forgot_VVD all_DT0 about_PRP it_PNP (5)
2. I_PNP agree_VVB with_PRP Juvenal_NP0 (5)
3. I_PNP 'm_VBB talking_VVG to_PRP you_PNP (5.5)
4. I_PNP am_VBB off_PRP-AVP food_NN1 hunting_NN1 (5.5)
5. I_PNP 'm_VBB safe_AJ0 in_PRP here_AV0 (5.5)
6. Nobody_PNI will_VM0 know_VVI about_PRP it_PNP (5.5)
7. His_DPS hands_NN2 shook_VVD with_PRP it_PNP (5.5)
8. I_PNP came_VVD to_PRP the_AT0 cathedral_NN1 (5.5)
9. The_AT0 Zodiac_NN1 swung_VVN-VVD behind_PRP it_PNP (5.5)
10. I_PNP can_VM0 look_VVI after_PRP myself_PNX (5.5)

  Word forms (except lexical items) + POS
1. Forgot_VVD all_DT0 about_PRP it_PNP (4)
2. I_PNP came_VVD to_PRP the_AT0 cathedral_NN1 (4)
```

```
3. I_PNP came_VVD with_PRP my_DPS pain_NN1 (4)
4. I_PNP loved_VVD a_AT0 man_NN1 with_PRP spirit_NN1 (4.5)
5. Germany_NP0 started_VVD in_PRP 1984_CRD (5)
6. He_PNP looked_VVD into_PRP the_AT0 living-room_NN1 (5)
7. She_PNP looked_VVD up_AVP at_PRP him_PNP (5)
8. Work_NN1-VVB started_VVD in_PRP autumn_NN1 1810_CRD (5)
9. His_DPS hands_NN2 shook_VVD with_PRP it_PNP (5)
10. Air_NN1 escaped_VVD into_PRP his_DPS mouth_NN1 (5)
```

References

Al Tamimi, Abdel Karim, Jaradat, Manar, Al-Jarrah, Nuha, and Ghanem, Sahar (2014). Aari: automatic arabic readability index. *International Arab Journal of Information Technology*, 11(4):370–378. 217

Albero, Brigitte (2000a). L'autoformation dans les dispositifs de formation ouverte et à distance: instrumenter le développement de l'autonomie dans les apprentissages. In *Les TIC au cœur de l'enseignement supérieur*, pages 139–159. Laboratoire Paragraphe, Université Paris VIII-Vincennes-St Denis. 41

Albero, Brigitte (2000b). *L'autoformation en contexte institutionnel: du paradigme de l'instruction au paradigme de l'autonomie*. Editions L'Harmattan. 41

Amini, Massih-Reza and Gaussier, Eric (2013). *Recherche d'information: applications, modèles et algorithmes*. Editions Eyrolles. 88

Anthony, Laurence (2013). A critical look at software tools in corpus linguistics. *Linguistic Research*, 30(2):141–161. 72, 76

Anthony, Laurence (2014). Antconc (version 3.4.3). Tokyo, Japan: Waseda University. [Computer Software]. 76

Aston, Guy (1997). Small and large corpora in language learning. In *PALC*, volume 97, pages 51–62. 73

Augustinus, Liesbeth, Vandeghinste, Vincent, and Van Eynde, Frank (2012). Example-based treebank querying. In *Proceedings of eighth international con-*

ference on Language Resources and Evaluation (LREC'2012), pages 3161–3167. 102, 137

Augustinus, Liesbeth, Vandeghinste, Vincent, and Vanallemeersch, Tom (2016). Poly-gretel: cross-lingual example-based querying of syntactic constructions. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'2016)*, pages 3549–3554. ELRA. 102

Bally, Charles (1921). *Traité de stylistique française*, volume 1. Carl Winter's Universitatsbuchhandlung. 7

Bandyopadhyay, Sanghamitra and Saha, Sriparna (2012). *Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications*. Springer Science & Business Media. 142, 145

Baranes, Marion (2015). *Normalisation orthographique de corpus bruités*. PhD thesis, Université Paris-Diderot-Paris VII. 52

Behrens, Heike (2006). The input–output relationship in first language acquisition. *Language and Cognitive Processes*, 21(1-3):2–24. 28

Blin, Françoise (2016). The theory of affordances. In Caws, Catherine and Hamel, Marie-Josée, editors, *Language-Learner Computer Interactions: Theory, methodology and CALL applications*, volume 2 of *Language Studies, Science and Engineering*, pages 41–64. John Benjamins Publishing Company. 92

Bloomfield, Leonard (1935). *Language*. London: George Allen and Unwin. 19, 21, 55, 56, 65

Boch, Françoise and Buson, Laurence (2012). Orthographe & grammaire à l'université. quels besoins? quelles démarches pédagogiques? *Scripta*, 16(30):31–51. 217

Boulton, Alex (2009). Testing the limits of data-driven learning: language proficiency and training. *ReCALL*, 21(1):37–54. 40

Boulton, Alex (2012). Beyond concordancing: Multiple affordances of corpora in university language degrees. *Procedia-Social and Behavioral Sciences*, 34:33–38. 42, 91

Brazil, David (1995). *A Grammar of Speech*. Describing English Language. Oxford University Press. 6

Brown, H. Douglas (2006). *Principles of Language Learning and Teaching*. Pearson Education, 5th edition edition. 30, 31

Bybee, Joan (2012). Where does grammar come from? In Rickerson, E. M. and Hilton, B., editors, *The 5-Minute Linguist. Bite-sized essays on Language and Languages*, pages 60–63. Equinox. 5

Bybee, Joan L (2006). From usage to grammar: The mind's response to repetition. *Language*, 82(4):711–733. 8

Carton, Francis (1995). L'apprentissage différencié des quatre aptitudes. *Verbum*, 1:63–74. 5

Chaker, Salem (2004). Kabylie: La langue. présentation générale. *Encyclopédie berbère*, (26):4055–4066. 22

Chandler, Brian and Tribble, Chris (1989). Longman mini-concordancer. Harlow UK: Longman Press. [Computer Software]. 75

Cheng, Winnie, Greaves, Chris, and Warren, Martin (2006). From n-gram to skipgram to concgram. *International journal of corpus linguistics*, 11(4):411–433. 82, 84

Choi, Jihun, Youn, Jonghem, and Lee, Sang-goo (2016). A grapheme-level approach for constructing a korean morphological analyzer without linguistic knowledge. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 3872–3879. IEEE. 65

Chomsky, Noam (1965). *Aspects of the Theory of Syntax*. MIT Press. 30

Chun, Jihye (2013). *Interface syntaxe-topologie et amas verbal en coréen et en français*. PhD thesis, Université Paris Ouest Nanterre La Défense. xiv, 228

Ciekanski, Maud (2014). Les corpus: de nouvelles perspectives pour l'apprentissage des langues en autonomie? *Recherches en didactique des langues et des cultures-Les Cahiers de l'Acedle*, 11(1):111–134. 41

Collins-Thompson, Kevyn (2014). Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135. 217

Constant, Matthieu (2012). *Mettre les expressions multi-mots au coeur de l'analyse automatique de textes: sur l'exploitation de ressources symboliques externes*. Habilitation à diriger des recherches, Université Paris-Est. 62

Cook, Vivian (1999). Going beyond the native speaker in language teaching. *TESOL quarterly*, 33(2):185–209. 35

Corbin, Pierre (1980). De la production des données en linguistique introspective. *Théories linguistiques et traditions grammaticales*, pages 121–179. 46

Daowadung, Patcharanut and Chen, Yaw-Huei (2011). Using word segmentation and svm to assess readability of thai text for primary school students. In *Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on*, pages 170–174. IEEE. 217

Desagulier, Guillaume (2017). *Corpus Linguistics and Statistics with R*. Springer. 130

Ellis, Rod (1997). *Second Language Acquisition*. Oxford Introduction to Language Study. Oxfort University Press. 16, 20

Firth, John R (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*. 131

Flora, Stephen Ray (2004). *The power of reinforcement*. SUNY Press. 29

Francis, W Nelson and Kucera, Henry (1979). *Brown Corpus Manual, Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers.* Brown University. 136

François, Thomas (2014). An analysis of a french as a foreign language corpus for readability assessment. *NEALT Proceedings Series*, 22:13. 217

Fries, Charles Carpenter and Traver, Alice Aileen (1940). *English word lists: a study of their adaptability for instruction.* American Council on Education. 39

Gibson, James Jerome (1977). The theory of affordances. In Shaw, Robert and Bransford, John, editors, *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, chapter The Theory of Affordances. Lawrence Erlbaum Associates. 91

Goddijn, Simo and Binnenpoorte, Diana (2003). Assessing manually corrected broad phonetic transcriptions in the spoken dutch corpus. In *Proceedings of ICPhS*, pages 1361–1364. 51

Goldstone, Robert L (1994). Influences of categorization on perceptual discrimination. *Journal of Experimental Psychology: General*, 123(2):178. 144

Goodman, Nelson (1972). Seven strictures on similarity. In Goodman, Nelson, editor, *Problems and Projects*, pages 437–447. New York: Bobbs-Merrill. 143

Greaves, Chris (2009). *ConcGram 1.0: A phraseological search engine.* John Benjamins Publishing Company. 84

Greaves, Chris and Warren, Martin (2007). Concgramming: A computer driven approach to learning the phraseology of english. *ReCALL*, 19:287–306. 83, 84, 85

Hamamura, Takeshi and Xu, Yi (2015). Changes in chinese culture as examined through changes in personal pronoun usage. *Journal of Cross-Cultural Psychology*, 46(7):930–941. 82

Han, Bo and Baldwin, Timothy (2011). Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of*

the Association for Computational Linguistics: Human Language Technologies, volume 1, pages 368–378. Association for Computational Linguistics. 52

Holec, Henri (1990). Qu'est-ce qu'apprendre à apprendre ? *Mélanges Pédagogiques*, 20:75–97. 40

Hou, Zhide (2016). A corpus-driven analysis of media representations of the chinese dream. *International Journal of English Linguistics*, 6(1):142–149. 85

Houdé, Olivier, Kayser, Daniel, Koenig, Olivier, Proust, Joëlle, and Rastier, François, editors (2004). *Dictionary of cognitive science: neuroscience, psychology, artificial intelligence, linguistics, and philosophy*. Routledge. 46

Huang, Xuedong, Alleva, Fileno, Hon, Hsiao-Wuen, Hwang, Mei-Yuh, Lee, Kai-Fu, and Rosenfeld, Ronald (1993). The sphinx-ii speech recognition system: an overview. *Computer Speech & Language*, 7(2):137–148. 83

Hunston, Susan (2002). *Corpora in applied linguistics*. Cambridge University Press. 72

Im, Ho-Bin, Hong, Kyung-Pyo, and Chang, Suk-In (2012). *Korean grammar for international learners*. Yonsei University Press. xvi

Jacques, Marie-Paule (2005). Pourquoi une linguistique de corpus. In Williams, Geoffrey, editor, *La Linguistique de Corpus*, pages 21–29. Rennes: Presses Universitaires de Rennes. 46

Jensen, Per Anker and Vikner, Carl (2004). The english pre-nominal genitive and lexical semantics. In *Possessives and Beyond: Semantics and Syntax*, pages 3–27. GLSA Publications. 85

Johns, Tim (1991). Should you be persuaded: Two samples of data-driven learning materials. *Classroom Concordancing: English Language Research Journal*, 4:1–16. 41

Kahane, Sylvain (2008). Le rôle des structures et représentations dans l'évolution des théories syntaxiques. In *Conférence de l'École Doctorale "Connaissance, Langage, Modélisation"*, pages 1–17. 158

Kardkovács, Zsolt T and Tikk, Domonkos (2007). On the transformation of sentences with genitive relations to sql queries. *Data & Knowledge Engineering*, 61(3):406–416. 85

Kennedy, Claire and Miceli, Tiziana (2001). An evaluation of intermediate students' approaches to corpus investigation. *Language learning & technology*, 5(3):77–90. 215, 216

Kettemann, Bernhard and Marko, Georg (2011). Data-driving critical discourse analysis. In Kübler, Natalie, editor, *Corpora, Language, Teaching, and Resources: From Theory to Practice*, volume 12 of *Etudes contrastives*, pages 20–48. Peter Lang. 41

Kim, Jong-Bok, Yang, Jaehyung, and Choi, Incheol (2004). Capturing and parsing the mixed properties of light verb constructions in a typed feature structure grammar. In *Proceedings of the 18th Pacific Asia Conference on Language Information and Computation (PACLIC'04)*. 67

Kim, Myoung-Cheol and Choi, Key-Sun (1999). A comparison of collocation-based similarity measures in query expansion. *Information processing & management*, 35(1):19–30. 68

Kim, Soohee (2013). When keeping up means falling behind: The dear price of stressing "correct" orthography in teaching korean as a heritage language. *Korean Language in America*, pages 71–91. 66, 179

Klein, Wolfgang (1989). *L'acquisition de langue étrangère*. Armand Colin. 26

Kraif, Olivier and Diwersy, Sascha (2012). Le Lexicoscope: un outil pour l'étude de profls combinatoires et l'extraction de constructions lexico-syntaxiques. In *Actes de la 19e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2012)*, pages 399–406. 84

Krashen, D. Stephen (1981a). *Second Language Acquisition and Second Language Learning*. Oxford: Pergamon Press. xii, 25, 30, 32

Krashen, Stephen D (1981b). Bilingual education and second language acquisition theory. *Schooling and language minority students: A theoretical framework*, pages 51–79. 31

Kuhl, Patricia K, Tsao, Feng-Ming, and Liu, Huei-Mei (2003). Foreign-language experience in infancy: Effects of short-term exposure and social interaction on phonetic learning. *Proceedings of the National Academy of Sciences*, 100(15):9096–9101. 18

Lacheret, Anne, Kahane, Sylvain, Beliao, Julie, Dister, Anne, Gerdes, Kim, Goldman, Jean-Philippe, Obin, Nicolas, Pietrandrea, Paola, and Tchobanov, Atanas (2014). Rhapsodie: a prosodic-syntactic treebank for spoken french. In *Ninth Language Resources and Evaluation Conference (LREC'14)*, pages 295–301. 70

Lanvers, Ursula (1999). Lexical growth patterns in a bilingual infant: The occurrence and significance of equivalents in the bilingual lexicon. *International Journal of Bilingual Education and Bilingualism*, 2(1):30–52. 19

Laurent, Maurice (2004). *Les jeunes, la langue, la grammaire : d'une grammaire implicite à une grammaire explicite. Catégories de mots et Constituants de la phrase.*, volume 1. Une Éducation Pour Demain. 217

Lebart, Ludovic, Salem, André, and Berry, Lisette (1997). *Exploring textual data*, volume 4. Springer Science & Business Media. 59

Lee, Dong-Joo, Yeon, Jong-Heum, Hwang, In-Beom, and Lee, Sang-Goo (2010). KKMA: A tool for utilizing Sejong Corpus based on Relational Database. *Journal of KIISE: Computing Practices and Letters*, 16(11):1046–1050. 181

Lee, Jeong-Tae, Cheon, Min-Ah, and Kim, Jae-Hoon (2015). 세종 말뭉치로부터 용언연어 추출 (verbal collocation extraction from sejong tagged corpus). In *Proceedings of the 27th Annual Conference on Human and Cognitive Language Technology (한글 및 한국어 정보처리 학술대회, HCLT'15)*, pages 121–123. 67

Lin, Yuri, Michel, Jean-Baptiste, Aiden, Erez Lieberman, Orwant, Jon, Brockman, Will, and Petrov, Slav (2012). Syntactic annotations for the google books ngram

corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174. Association for Computational Linguistics. 82

Livingston, Kenneth R, Andrews, Janet K, and Harnad, Stevan (1998). Categorical perception effects induced by category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(3):732–753. 141

Luhn, Hans Peter (1960). Key word-in-context index for technical literature (kwic index). *Journal of the Association for Information Science and Technology*, 11(4):288–295. 131

Lukoff, Fred (1982). *An introductory course in Korean*. Yonsei University Press. 171

Magistry, Pierre (2013). *Unsupervised word segmentation and wordhood assessment: the case for mandarin chinese*. PhD thesis, Paris 7. 55

Magistry, Pierre, Fabre, Murielle, and Goudin, Yoann (2017). Indices phonologiques des sinogrammes: de l'étude de l'acquisition à la modélisation pour l'apprentissage. *Traitement Automatique des Langues*, 57(3):39–63. 9

Marcellesi, Christiane (1976). Norme et enseignement du français. *Cahiers de linguistique sociale*, (1):1–9. 5

Martin, Samuel Elmo (1954). *Korean Morphophonemics*. Linguistic Society of America. 171

McEnery, Tony and Hardie, Andrew (2012). *Corpus linguistics: Method, theory and practice*. Cambridge University Press. 47, 48, 74, 76, 77, 78, 141

McNeil, Mary Charles, Polloway, Edward A, and Smith, J David (1984). Feral and isolated children: Historical review and analysis. *Education and Training of the Mentally Retarded*, pages 70–79. 20

Medin, Douglas L, Goldstone, Robert L, and Gentner, Dedre (1993). Respects for similarity. *Psychological review*, 100(2):254–278. 143

Mélanie-Becquet, Frédérique and Fuchs, Catherine (2011). Elaboration d'une base de données d'exemples de structures comparatives: de la grille d'annotation au systeme d'interrogation. *Corpus*, (10):273–295. 101

Meurers, Walt Detmar and Müller, Stefan (2009). Corpora and syntax. *Corpus linguistics: An international handbook*, 2:920–933. 107, 112, 114

Michel, Jean-Baptiste, Shen, Yuan Kui, Aiden, Aviva Presser, Veres, Adrian, Gray, Matthew K, Pickett, Joseph P, Hoiberg, Dale, Clancy, Dan, Norvig, Peter, Orwant, Jon, et al. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182. 82

Mikolov, Tomas, Yih, Wen-tau, and Zweig, Geoffrey (2013). Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, pages 746–751. 130

Nation, Ian SP (2001). *Learning vocabulary in another language*. Ernst Klett Sprachen. 141

Navarro, Gonzalo (2001). A guided tour to approximate string matching. *ACM Computing Surveys (CSUR)*, 33(1):31–88. 152

Nebhi, Kamel, Goldman, Jean-Philippe, and Laenzlinger, Christopher (2010). FipsColor: grammaire en couleur interactive pour l'apprentissage du français. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2010)*, Actes de TALN 2010. ID: unige:24086. 217

Niemytzki, Viktor Vladimirovich (1927). On the "third axiom of metric space". *Transactions of the American Mathematical Society*, 29(3):507–513. 145

Norman, Donald Arthur (1988). *The Psychology of Everyday Things*. New York: Basic Books. 92, 97

O'Donnell, Matthew Brook, Scott, Mike, Mahlberg, Michaela, and Hoey, Michael (2012). Exploring text-initial words, clusters and concgrams in a newspaper corpus. *Corpus Linguistics and Linguistic Theory*, 8(1):73–101. 70

O'Keeffe, Anne, McCarthy, Michael, and Carter, Ronald (2007). *From corpus to classroom: Language use and language teaching*. Cambridge University Press. 45

Pae, Hye K (2011). Is korean a syllabic alphabet or an alphabetic syllabary. *Writing Systems Research*, 3(2):103–115. 151

Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab. 133

Pallier, Christophe (2007). Critical periods in language acquisition and language attrition. In Köpke, Barbara, Schmid, Monika S, Keijzer, Merel, and Dostert, Susan, editors, *Language attrition: Theoretical perspectives*, volume 33 of *Studies in Bilingualism*, pages 155–168. John Benjamins Publishing. 26

Pallier, Christophe, Dehaene, Stanislas, Poline, Jean-Baptiste, LeBihan, Denis, Argenti, Anne-Marie, Dupoux, Emmanuel, and Mehler, Jacques (2003). Brain imaging of language plasticity in adopted adults: Can a second language replace the first? *Cerebral cortex*, 13(2):155–161. 26

Palmer, Harold Edward (1933). *Second interim report on English collocations*. Institute for Research in English Teaching, Tokyo. 39

Panckhurst, Rachel (2010). Texting in three european languages: does the linguistic typology differ. In *Actes du Colloque i-Mean 2009 Issues in Meaning in Interaction*. 52

Park, Eunjeong L. and Cho, Sungzoon (2014). KoNLPy: Korean natural language processing in Python. In *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*, Chuncheon, Korea. 178

Pearson, Barbara Zurer, Fernández, Sylvia C, and Oller, D Kimbrough (1995). Cross-language synonyms in the lexicons of bilingual infants: One language or two? *Journal of child language*, 22:345–368. 19

Pechenick, Eitan Adam, Danforth, Christopher M, and Dodds, Peter Sheridan (2015). Characterizing the google books corpus: Strong limits to inferences of socio-cultural and linguistic evolution. *PloS one*, 10(10):e0137041. 82

Piattelli-Palmarini, Massimo, editor (1980). *Language and learning: the debate between Jean Piaget and Noam Chomsky*. London: Routledge and Keagan Paul. 30

Poudat, Céline and Landragin, Frédéric (2017). *Explorer un corpus textuel: Méthodes-pratiques-outils*. De Boeck Superieur. 78

Rappaport, Gilbert C (2004). The syntax of possessors in the nominal phrase: Drawing the lines and deriving the forms. In *Possessives and Beyond: Semantics and Syntax*, volume 29, pages 243–261. GLSA Publications. 85

Resnik, Philip and Elkiss, Aaron (2005). The linguist's search engine: An overview. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 33–36, Stroudsburg, PA, USA. Association for Computational Linguistics. 107

Rose, Daniel E and Levinson, Danny (2004). Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM. 93, 133

Rosenshine, Barak and Meister, Carla (1992). The use of scaffolds for teaching higher-level cognitive strategies. *Educational leadership*, 49(7):26–33. 31

Sag, Ivan A, Baldwin, Timothy, Bond, Francis, Copestake, Ann, and Flickinger, Dan (2002). Multiword expressions: A pain in the neck for NLP. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15. Springer. 61, 62

Santorini, Beatrice (1991). Part-of-speech tagging guidelines for the penn treebank project. Technical report, University of Pennsylvania. (3rd. 137

Sato, Satoshi, Matsuyoshi, Suguru, and Kondoh, Yohsuke (2008). Automatic assessment of japanese text readability based on a textbook corpus. In *Proceedings*

of the Sixth International Conference on Language Resources and Evaluation (LREC'08). 217

Schaeffer-Lacroix, Eva (2015). Analyse de trois systèmes de gestion de corpus pour l'enseignement-apprentissage des langues étrangères. *Alsic. Apprentissage des Langues et Systèmes d'Information et de Communication*, 18(1):1–23. 112

Schmid, Helmut (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK. 136

Schwering, Angela (2008). Approaches to semantic similarity measurement for geo-spatial data: A survey. *Transactions in GIS*, 12(1):5–29. 141

Scott, Mike (2017). Wordsmith tools help. Stroud: Lexical Analysis Software. [Computer Software]. 76

Seliger, Herbert (1996). Primary language attrition in the context of bilingualism. *Handbook of second language acquisition*, pages 605–625. 22

Selinker, Larry (1972). Interlanguage. *IRAL-International Review of Applied Linguistics in Language Teaching*, 10(1-4):209–232. 35

Shin, Hyopil (2008). The 21st sejong project: with a focus on building of the selk (sejong electronic lexicon of korean) and the knc (korean national corpus). Invited Talk at the Third International Joint Conference on Natural Language Processing (IJCNLP'08). 67

Shin, Sung-Ock (1988). *Tense and aspect in Korean*. PhD thesis. 171

Sinclair, John (2005). Corpus and text-basic principles. *Developing linguistic corpora: A guide to good practice*, pages 1–16. 44

Sinclair, John, Hanks, Patrick, Fox, Gwyneth, Moon, Rosamund, Stock, Penny, et al. (1987). *Collins COBUILD English language dictionary*. Collins London. 39

Singleton, David Michael and Ryan, Lisa (2004). *Language acquisition: The age factor*, volume 9. Multilingual Matters. 20

Skinner, Burrhus Frederic (1957). *Verbal behavior*. The Century Psychology Series. New York: Appleton-Century-Crofts. 29

Slavin, Robert E. (2005). *Educational Psychology: Theory and Practice*. Allyn & Bacon. 30

Sohn, Ho-Min (2013). *Korean*. KLEAR and RILI studies in Korean Language and Linguistics. Korea University Press. xvi, 11, 67, 171, 172, 174, 175, 199, 222, 227

Taylor, Insup and Olson, David R (1995). *Scripts and literacy: Reading and learning to read alphabets, syllabaries, and characters*, volume 7. Springer Science & Business Media. 151

Teknomo, Kardi (2015). Similarity measurement. `http://people.revoledu.com/kardi/tutorial/Similarity`. 145

Thorndike, Edward Lee et al. (1932). *Teacher's word book of the twenty thousand words found most frequently and widely in general reading for children and young people*. Teachers college, Columbia university. 38

Thorndike, Edward Lee and Lorge, Irving (1944). *The teacher's wordbook of 30,000 words*. New York: Columbia University, Teachers College. x, 38

Tono, Yukio (2011). Talc in action: recent innovations in corpus-based english language teaching in japan. In Frankenberg-Garcia, Ana, Flowerdew, Lynne, and Aston, Guy, editors, *New Trends in Corpora and Language Learning*, chapter 1, pages 3–25. Bloomsbury Academic. 39

Tournadre, Nicolas (2014). *Le Prisme des langues: essai sur la diversité linguistique et la difficulté des langues*. Asiathèque, Maison des Langues du Monde. 21

Wang, Avery (2003). An industrial strength audio search algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, volume 2003, pages 7–13. Washington, DC. 147

Wang, Ilaine (2016). A syntax-based query system adapted to language learning and teaching. In *American Association for Corpus Linguistics (AACL) and Technology for Second Language Learning (TSLL) Joint Conference*, Ames, USA. Poster presentation. 88, 91

Wang, Ilaine, Kahane, Sylvain, and Tellier, Isabelle (2014). Macrosyntactic segmenters of a french spoken corpus. In *Ninth Language Resources and Evaluation Conference (LREC'14)*, pages 3891–3896. 3

Webster, Jonathan J and Kit, Chunyu (1992). Tokenization as the initial phase in nlp. In *Proceedings of the 14th conference on Computational linguistics-Volume 4*, pages 1106–1110. Association for Computational Linguistics. 58

Wha, Lee Tae, Jin, Kang Soo, Hyun, Kim Hye, Ra, Woo So, and Sinhye, Kim (2011). Suitability and readability assessment of printed educational materials on hypertension. *Journal of Korean Academy of Nursing*, 41(3). 217

Wilks, Yorick (2005). Reveal: the notion of anomalous texts in a very large corpus. In *Tuscan Word Centre International Workshop. Certosa di Pontignano, Tuscany, Italy*, volume 31. 83

Yi, Wooyong, Park, Eunil, and Cho, Kwangsu (2011). E-book readability, comprehensibility and satisfaction. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, page 38. ACM. 217

Yip, Virginia and Matthews, Stephen (2007). *The bilingual child*. Ernst Klett Sprachen. 19

Index