Disambiguation Experiment for Spanish

Jorge Graña Gil

July 1998

Abstract

This document is, at the same time, a comparative study of two taggers, the BRILL system and the GALENA system, and an experiment to obtain conclusions about the features that a training corpus in Spanish should have, about the training strategies that must be used in order to obtain a given success rate in the tagging process, and about what is the highest ratio that we can expect. Finally, our work describes a complete and general methodology for the evaluation of all kinds of tagging systems.

Contents

Chapter 1

Introduction

Elimination of ambiguities is a crucial task during the process of tagging a text in natural language. If we take in isolation, for instance, the word `sobre`, we can see that it has three possible tags in Spanish: substantive (envelope), preposition (on), and subjunctive present, 1st and 3rd person of the verb `sobrar` (to be unnecessary). However, if we have a look at the context in which the word appears, only one of the three tags is possible. In addition, we are also interested in being able to give a tag to all the words that appear in a text, but are not present in our dictionary, and to guarantee that this tag is the correct one. A good performance in this stage will improve the viability of syntactic and semantic analysis.

One of the most common techniques for solving this kind of problem is the use of Hidden Markov Models [1]. This technique applies a stochastic procedure, which calculates Markov chains from a training corpus, and then uses these chains to tag new texts.

1.1 Brill system (transformation-based error driven tagging)

Another technique is transformation-based error driven tagging. This method automatically produces a lexicon and a set of lexical and contextual transformation rules. If a given transformation yields a correct pair (word, tag), then the rule is applied in order to try to obtain the best tagging. This is the technique used by Brill's tagger [2].

1.2 Galena system (tagging level configuration)

Galena system [3] uses a very simple stochastic procedure, also applied to a training corpus, but provides a new feature: the user can choose different configurations or specialization levels during the training process, by using the lexical category followed by all, some or none of the attributes present in the current tag set.

The aim of the present study is not only to compare the performance of both systems when they are trained and used in the same conditions, but also to fix training strategies and sizes of training corpora in order to guarantee a given success rate in the tagging process.

Chapter 2

Reference corpus

Linguistic resources for Spanish are not very common. One of the free available corpora is the *International Telecommunications Union CCITT Handbook*, also known as *The Blue Book*, in the environment of telecommunications, and as *ITU Corpus*, in the linguistic environment. It is the main collection of texts about telecommunications, and due to its large size is an excellent test framework for studying behaviours of tagging systems. The original corpus has 5 million words. Two versions of this text exist that have already been disambiguated: the whole corpus tagged with the Spanish version of the XEROX tagger, and a subcorpus of about half a million words hand-corrected by the University of Lancaster.

The second one is the text that has been used as the reference corpus in our study, although both can be downloaded from the URL

`http://www.lllf.uam.es/~fernando/projects/es_corpus.html`

which belongs to the set of pages that support all the information of the CRATER project [4].

2.1 ITU CORPUS features

The features of our reference corpus are the following:

- The corpus has 486073 words. The size of the file is 7564781 characters. The syntax of each line is:

  `form_1/tag_1/lemma_1 form_2/tag_2/lemma_2 ... form_n/tag_n/lemma_n ./Q./.`

  Lines could also end with `?/Q?/?` or with `.../Q.../...` or with any other punctuation mark, and thus each line is a sentence. The corpus has 14919 sentences, that is 33 words per sentence, on the average.

- The corpus has 45679 verbal forms, where 581 forms are in passive voice, 870 are compound verbal forms, and 3415 are verbal forms with one enclitic pronoun.

- The features of the lexicon formed by all the forms that appear in the corpus are the following:

  – 15745 forms with 1 tag,

  – 1097 forms with 2 tags,

- 223 forms with 3 tags,
- 61 forms with 4 tags,
- 8 forms with 5 tags,
- 3 forms with 6 tags and
- 1 form with 7 tags.

That is, 17138 different forms, with 18917 possible tags, and corresponding to 12462 different lemmas.

If we calculate the *percentage of ambiguous forms* and the *average number of tags per form*, we obtain:

$$\%\, ambiguous\, forms = \frac{\#\, ambiguous\, forms}{\#\, forms} \times 100 = \frac{17138 - 15745}{17138} \times 100 = 8.13\,\%$$

$$\#\, average\, of\, tags\, per\, form = \frac{\#\, tags}{\#\, forms} = \frac{18917}{17138} = 1.10\, tags\, per\, form$$

- Much more interesting is to calculate the same features directly with all the words in the corpus, and we obtain the following numbers:
  - 263592 words with 1 tag,
  - 107746 words with 2 tags,
  - 79751 words forms with 3 tags,
  - 7013 words with 4 tags,
  - 18949 words with 5 tags,
  - 8528 words with 6 tags and
  - 494 words with 7 tags.

  That is, 486073 different words, with 895760 possible tags.

$$\%\, ambiguous\, words = \frac{\#\, ambiguous\, words}{\#\, words} \times 100 = \frac{486073 - 263592}{486073} \times 100 = 45.77\,\%$$

$$\#\, average\, of\, tags\, per\, word = \frac{\#\, tags}{\#\, words} = \frac{895760}{486073} = 1.84\, tags\, per\, word$$

- And, finally, in order to show the general aspect of all these data, we include a small set of lines taken from the corpus:

```
En/P/en esta/Eyfs/este colaboraci'on/Scfs/colaboraci'on debe/V3spi0/deber
reconocerse/V000f0EP1/reconocer el/Dms/el car'acter/Scms/car'acter
consultivo/Ams0/consultivo de/P/de las/Dfp/el organizaciones/Scfp/organizaci'on
que/Cs/que participan/V3ppi0/participar en/P/en los/Dmp/el trabajos/Scmp/trabajo
de/P/de el/Dms/el CCITT/Spys/CCITT ,/Q,/, en_particular/Wy/en_particular a/P/a
la/Dfs/el ISO/Zgfs/ISO ,/Q,/, desde/P/desde el/Dms/el punto/Scms/punto de/P/de
vista/Scfs/vista de/P/de su/Mdys3s/suyo labor/Scfs/labor
con_respecto_a/P/con_respecto_a los/Dmp/el sistemas/Scmp/sistema de/P/de
datos/Scmp/dato y/Cc/y a/P/a las/Dfp/el comunicaciones/Scfp/comunicaci'on ./Q./.
```

and from the lexicon:

```
abajo Wn
abandon'o V3sei0
abandona V3spi0
abandonar V000f0
abandonar'a V3sfi0
abandono Scms
abarca V3spi0
abarcadas V0p0pf
abarcados V0p0pm
abarcan V3ppi0
abarcar V000f0
abarque Vysps0
abertura Scfs
abierta Afs0 V0s0pf
abiertas Afp0 V0p0pf
abierto Ams0 V0s0pm
abiertos Amp0
...
```

where the first column is the form and the other columns are the tags sorted by frequencies, from the highest to the lowest.

Chapter 3

Tag set

Brill's tagger does not need an explicit declaration of the tag set, because the tool is able to automatically deduce it by itself from the learning corpora. Therefore, the only important question with regard to the Brill system is to design a training corpus in which all the tags in the tag set appear at least once, as we will see later, during the discussion of the possible training strategies.

3.1 Galena system tag set

However, the Galena system does need a previous knowledge about the tag set. Furthermore, the system uses a lexical database in which each row references one or more elements of the tag set. For this reason, we have decided to use the tag set in the Galena system for both tools throughout the experiment. This tag set can be seen in figure 3.1.

This figure contains a set of 1048 tags, although not all of them are correct combinations from a linguistic point of view. On the other hand, for this study, we have decided to extend the tag set in the Galena system in order to mark compound verbal forms, verbal forms in passive voice and verbal forms with enclitic pronouns, as is described in the next section.

Therefore, the final tag set we will use throughout the experiment has 373 tags, and is shown in appendix A.

3.2 Extending the Galena tag set for the experiment

This section describes the notation that we have used in order to explicitly mark compound verbal forms, verbal forms in passive voice and verbal forms with enclitic pronouns. If `<tag>` is the tag that was previously used in the text, marks consist of adding a suffix to `<tag>`.

3.2.1 Marking compound verbal forms

Compound verbal forms, such as `he comido` (have eaten), have two components:

1. The auxiliary verb (`he`), which will be marked as `<tag>CT1`.
2. The participle (`comido`), which will be marked as `<tag>CT2`.

3.2.2 Marking verbal forms in passive voice

Verbal forms in passive voice, such as `fue comido` or `ha sido comido` (was eaten or has been eaten), have two components:

1. The auxiliary verb (`fue`), which will be marked as `<tag>PCT1`.
2. The participle (`comido`), which will be marked as `<tag>PCT2`.

or three:

1. The first auxiliary verb (`ha`), which will be marked as `<tag>PCT1`.
2. The second auxiliary verb (`sido`), which will be marked as `<tag>PCT2`.
3. The participle (`comido`), which will be marked as `<tag>PCT3`.

This produces the following implications:

- Each form of the verb `haber` (to have) can have three different taggings:
  - `<tag>` (alone)
  - `<tag>CT1` (Compound Tense 1st part)
  - `<tag>PCT1` (Passive Voice 1st part)
- Each form of the verb `ser` (to be) can have two different taggings:
  - `<tag>` (alone)
  - `<tag>PCT1` (Passive Voice 1st part)
- Each verbal participle can have five different taggings:
  - `V0s0pm` (participle)
  - `V0s0pmCT2` (Compound Tense 2nd part)
  - `V0s0pmPCT2` (Passive Voice 2nd part)
  - `V0s0pmPCT3` (Passive Voice 3rd part)
  - `Ams0` (adjective)

  The example above is only for the masculine singular participle. More exactly, for instance for the verb `ver` (to see), we have all these forms:

  - `visto V0s0pm V0s0pmCT2 V0s0pmPCT2 V0s0pmPCT3 Ams0`
  - `vista V0s0pf V0s0pfPCT2 V0s0pfPCT3 Afs0`
  - `vistos V0p0pm V0p0pmPCT2 V0p0pmPCT3 Amp0`
  - `vistas V0p0pf V0p0pfPCT2 V0p0pfPCT3 Afp0`

3.2.3 Marking verbal forms with enclitic pronouns

In Spanish, the verbal form with enclitic pronouns, such as `tenerlo` or `verse` (have it or see himself) can have up to 3 of these pronouns. However, the ITU CORPUS presents forms with only one enclitic pronoun, which will be marked as `<tag>EP1`.

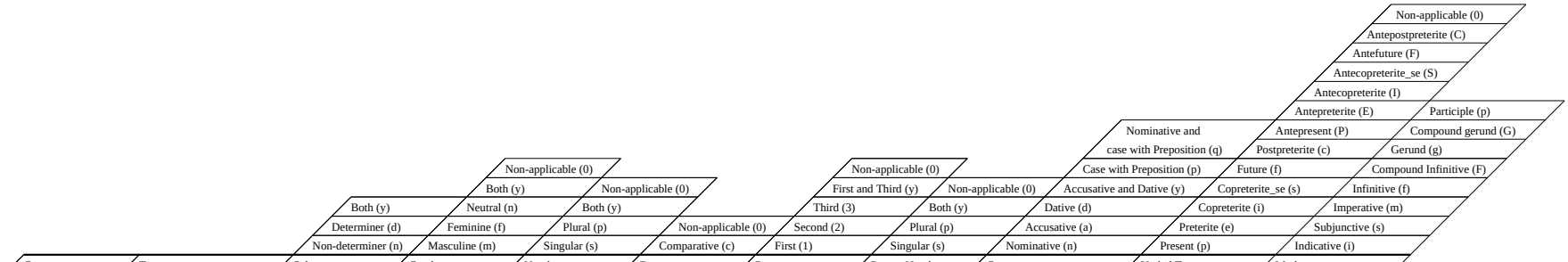
3.3 CRATER project tag set

However, the original version of the ITU CORPUS is not tagged with the GALENA EXTENDED TAG SET, but with another tag set, coming from the research performed in the frame of the CRATER project [4].

The CRATER TAG SET originates from the standard EAGLES, and is a little more precise than the GALENA TAG SET, because its cardinal is 475 tags, as is shown in appendix B.

3.4 Mapping the CRATER TAG SET into the GALENA EXTENDED TAG SET

Therefore, previous to the experiment, it is necessary to establish a correspondence from the tags in the CRATER TAG SET to the tags in the GALENA EXTENDED TAG SET. The application of this mapping to the ITU CORPUS provides us with a corpus completely tagged with the GALENA EXTENDED TAG SET for our experiment. This mapping is shown in appendix C.



Category	Type	Subtype	Gender	Number	Degree	Person	Person Number	Case	Verbal Tense	Mode
Substantive (S)	Common (c)		m f y	s p y						
	Proper (p)		m f y	s p y						
Adjective (A)			m f y	s p y	c 0					
Demonstrative (E)		n y	m f n y	s p y						
Relative (T)		n d y	m f n y	s p y						
Indefinite (I)		n d y	m f n y	s p y						
Interrogative (G)		n y	m f n y	s p y						
Possessive (M)		n d y	m f n y	s p y		1 2 3	s p			
Numeral (N)	Cardinal (c)	n d y	m f y	s p						
	Ordinal (o)	n d y	m f y	s p						
	Partitive (p)	n	m f y	s p						
	Multiple (m)	n	m f y	s p						
Article (D)			m f n	s p						
Personal Pronoun (R)	Tonic (t)		m f y (last)			1 2 3	s p y	n a d y p q		
	Proclitic atonic (p)		m f y (last)			1 2 3	s p y	n a d y p q		
	Enclitic atonic (e)		m f y (last)			1 2 3	s p y	n a d y p q		
Verb (V)			m f 0 (last)			1 2 3 y 0	s p 0		p i s e f c 0 P I S E F C	i s m f F g G p
Preposition (P)										
Conjunction (C)	Coordinate (c)									
	Subordinate (s)									
Adverb (W)	Nuclear (n)									
	Modifier (m)									
	Nuclear and Modifier (y)									
	Relative (r)									
	Exclamative (i)									
	Interrogative (v)									
Interjection (Y)										
Punctuation Mark (Q)	Dot (.)									
	Comma (,)									
	Semicolon (;)									
	Colon (:)									
	Open Question Mark ('?)									
	Close Question Mark (?)									
	Open Exclamation Mark ('!)									
	Close Exclamation Mark (!)									
	Open Parenthesis (()									
	Close Parenthesis ())									
	Dash (-)									
	Quotes (")									
	Dots (...)									
Peripheral (Z)	Foreign Word (e)		m f y 0	s p y 0						
	Formula (f)		m f y 0	s p y 0						
	Abbreviation (a)		m f y 0	s p y 0						
	Symbol (s)		m f y 0	s p y 0						
	Acronym (g)		m f y 0	s p y 0						
	Other (o)		m f y 0	s p y 0						
Unknown (U)										

Figure 3.1: GALENA TAG SET

Chapter 4

Lexica

The BRILL system neither needs an explicit declaration of the tag set, as we have seen above, nor a lexicon, because it is able to generate it from the training corpus. However, in the present study we have added a large extra Spanish lexicon to the BRILL system, in order to check whether this can help improve its performance or not.

This lexicon is the lexicon used by the GALENA system. Its features are described below.

4.1 GALENA LEXICON features

However, before describing the lexicon, let us see some general data about the GALENA system:

- The lexical database has the following numbers of stems:
  - 2944 adjectives
  - 125 adverbs
  - 2 articles
  - 102 conjunctions
  - 14 demonstratives
  - 36 indefinites
  - 40 interjections
  - 4 interrogatives
  - 62 numerals
  - 67 peripherals
  - 29 pronouns
  - 8 possessives
  - 43 prepositions
  - 5 relatives
  - 8027 substantives
  - 5600 verbs
  - 30 special

  That is, 17138 stems corresponding to 13667 different lemmas.

- The architecture of the tagger is a finite state transducer with 228445 states.
- The size of the tagger (executable file) is 2736128 bytes.
- The compilation time is about 9 minutes and the tagging speed is 1360 words per second, in the host

```
covas.dc.fi.udc.es
(SunOS covas 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-Enterprise)
```

We have generated the GALENA LEXICON from the lexical database described above, and the features are the following:

- 265418 forms with 1 tag,
- 9995 forms with 2 tags,
- 588 forms with 3 tags,
- 11343 forms with 4 tags,
- 4097 forms with 5 tags and
- 163 forms with 6 tags.

That is, 291604 different forms, with 354007 possible taggings, and corresponding, as we have seen, to 13667 different lemmas.

However, the lexical database does not store any information about frequencies, because this information is relevant for words, not for stems, and it depends on the corpora. Therefore, if we use only the lexical database, the generated lexicon will have tags sorted in no particular way and will be completely useless for BRILL system. In order to avoid this problem, we have study the ambiguity classes present in the ITU CORPUS, and we have generated the GALENA LEXICON with the tags of each form sorted as in the most frequent combination of tags in the ITU CORPUS for their ambiguity class.

The intersection between the GALENA LEXICON and the lexicon formed by all the words in the ITU CORPUS contains 6594 forms, where:

- 3670 forms (involving 166573 words in the ITU CORPUS, that is, 34.27% of the words) are in the same ambiguity class in both lexica.
- 2924 forms (involving 216106 words in the ITU CORPUS, that is, 44.46% of the words) are in different ambiguity classes in both lexica.

In order to be not too much cheat, all of these forms in common, even the ones in the same ambiguity class, has been generated in the same general way as we have explained above. We could generated them exactly as they appear in the lexicon formed by all the words in the ITU CORPUS, but to have a corpus in which to perform a previous study about lexica or even about ambiguity classes is not the usual case in practice.

Chapter 5

Strategy for the experiment

This chapter describes in detail the strategy that we have used when performing the experiments. Basically, the steps are: to build a training corpus formed by sentences randomly taken from the ITU CORPUS, to train the tools, to retag the remaining portion of the corpus, and to compare it with the original one. Obviously, this must be done several times and with different sizes of training corpus. The following sections describe the details.

5.1 Building Training Corpus Zero (every tag present at least once)

However, there is a component that is always present in the training corpora of each test, which is not a part of the ITU CORPUS and which we will call *Training Corpus Zero*. It should be a very small text and its unique relevant feature is that it contains each tag in the current tag set at least once. It is very convenient to include this component, because, as we have seen, the BRILL system has no *a priori* knowledge about the tag set.

In general, we can consider several aspects when thinking about this fixed component. With regard to size, the smaller the better, in order to not alter too much the features of the original reference corpus. And in regard to the building of it, and in order to obtain a small size, we prefer to create a new text instead of extracting sentences from the original corpus until all the tags in the tag set are covered. For this reason, we say that our *Training Corpus Zero* is not a part of the ITU CORPUS. Obviously, a lot of time may to be spent on this. It depends on the size of the tag set. In our case, we needed two days to cover all the 373 tags.

More exactly, the features of our *Training Corpus Zero* are the following:

- The corpus has 1228 words. The size of the file is 19921 characters. The syntax of each line is:

  ```
  form_1/tag_1/lemma_1 form_2/tag_2/lemma_2 ... form_n/tag_n/lemma_n ./Q./.
  ```

  Lines could also end with `?/Q?/?` or with `.../Q.../...` or with any other punctuation mark, and thus each line is a sentence. The corpus has 54 sentences, that is 22 words per sentence, in average.

- The corpus has 332 verbal forms, where 52 forms are in passive voice, 52 are compound verbal forms, and 11 are verbal forms with one enclitic pronoun.

- The features of the lexicon formed by all the forms that appear in the corpus are the following:
  - 544 forms with 1 tag,
  - 39 forms with 2 tags and
  - 3 forms with 3 tags.

  That is, 586 different forms, with 631 possible taggings, and corresponding to 386 different lemmas.

  If we calculate the *percentage of ambiguous forms* and the *average number of tags per form*, we obtain:

$$\%\, ambiguous\, forms = \frac{\#\, ambiguous\, forms}{\#\, forms} \times 100 = \frac{586 - 544}{586} \times 100 = 7.16\,\%$$

$$\#\, average\, of\, tags\, per\, form = \frac{\#\, tags}{\#\, forms} = \frac{631}{586} = 1.08\, tags\, per\, form$$

- And directly with all the words in the corpus:
  - 987 words with 1 tag,
  - 232 words with 2 tags and
  - 9 words with 3 tags.

  That is, 1228 different words, with 1478 possible tags.

$$\%\, ambiguous\, words = \frac{\#\, ambiguous\, words}{\#\, words} \times 100 = \frac{1228 - 987}{1228} \times 100 = 19.63\,\%$$

$$\#\, average\, of\, tags\, per\, word = \frac{\#\, tags}{\#\, words} = \frac{1478}{1228} = 1.20\, tags\, per\, word$$

5.2 Building Training Corpus One (6 different sizes of random sentences)

Training Corpus One is the variable component of the training corpus in each test, but it is the main component because it is the largest. We have chosen five different sizes: 1000, 2000, 3000, 4000 and 5000 sentences randomly taken from the ITU CORPUS. More exactly, our procedure extracts a 5000 sentences block once, and then builds 5 different training corpora: the first with the first 1000 sentences in the block, the second with the first 2000 sentences, the third with the first 3000, and so on until the 5000 sentences in the block are covered. In this way, we have 5 different tests, but the portion of text to be retagged and compared is the same in each one of them.

However, this extraction process has been performed 3 times, which produces three banks of experiments, each one with 5 different tests. That is, a total of 15 tests.

And finally, we have trained only once with a size of 10000 sentences, which makes a final total number of 16 tests. We will call this last bank *Special Bank*, and each test as follows:

TC1 Size / Bank	Bank 1	Bank 2	Bank 3	Special Bank
1000 sentences	test11	test21	test31	-
2000 sentences	test12	test22	test32	-
3000 sentences	test13	test23	test33	-
4000 sentences	test14	test24	test34	-
5000 sentences	test15	test25	test35	-
10000 sentences	-	-	-	testSP

Figures 5.1 and 5.2 describe the whole process. The evaluation rates S1 and S2 are described in the next chapter.

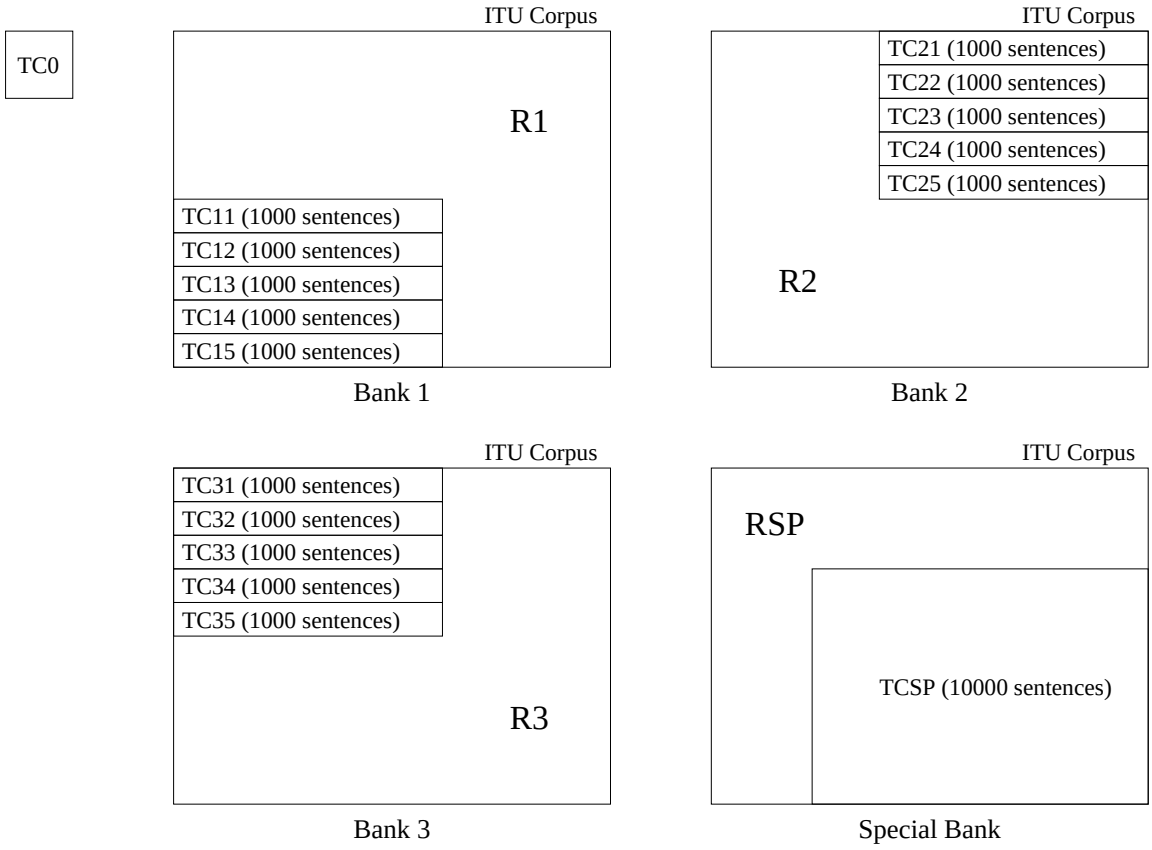


Figure 5.1: Training Corpus Zero and extraction of random sentences for building Training Corpus One

5.3 Retagging with BRILL (4 different lexica)

As we have seen, in each experiment and from the training corpus, the BRILL system is able to generate its own lexicon, that we will call BRILL LEXICON, and a set of lexical and contextual rules. However, it also is possible to use the system with the same set of rules, but with a different lexicon to the generated one.

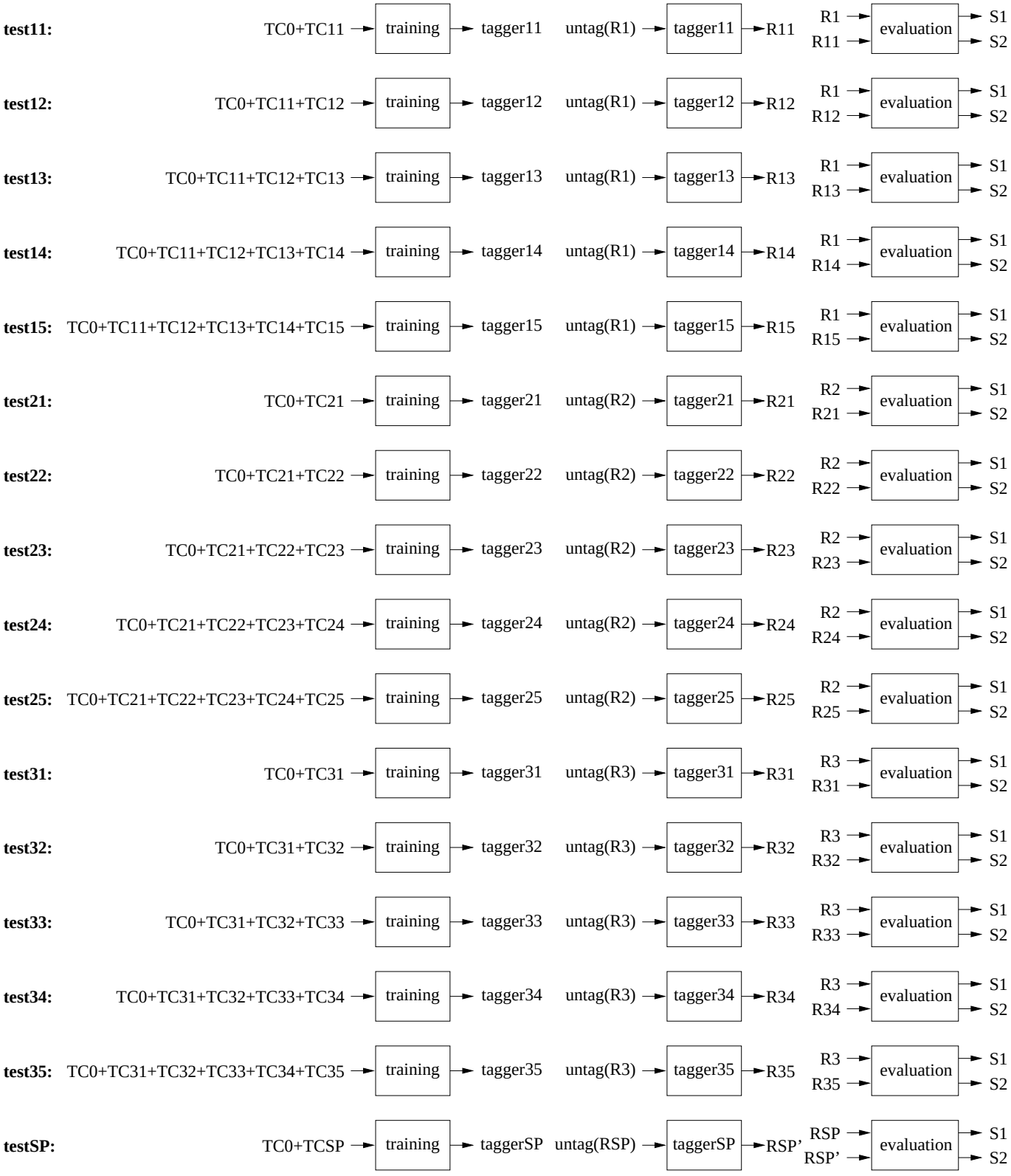


Figure 5.2: Steps for each test

In this study we wanted to add a large extra Spanish lexicon to the BRILL system, the GALENA LEXICON (see chapter 4), in order to check whether this can help improve its performance or not.

On the other hand, we also wanted to check the behaviour of BRILL system when it uses a lexicon containing all the words that appear in the text that we are going to tag. Therefore, we have also generated all the lexicon present in the ITU CORPUS. We will call this lexicon ITU LEXICON. The objective here is to calculate the highest level of pure disambiguation, that is, the highest success rate in the tagging process that we can expect with a given training corpus size.

And finally, by putting together both the ITU LEXICON and the GALENA LEXICON, we check the behaviour of BRILL system in an intermediate situation.

A lot of different ways exist to mix up these three lexica together. But once again, in order to be not too much cheat, we always have involved the BRILL LEXICON as primary lexicon in each combination. This defines the four situations in which we have evaluated the BRILL system:

- With the lexicon: BRILL LEXICON.
- With the lexicon: BRILL LEXICON + GALENA LEXICON.
- With the lexicon: BRILL LEXICON + ITU LEXICON.
- With the lexicon: BRILL LEXICON + ITU LEXICON + GALENA LEXICON.

All of them have been generated with the PERL script `combine-lexicons.prl`, that can be found in the `Utilities` directory of the BRILL implementation package.

Experiments with the BRILL system have been performed on the following hosts:

- The first bank in

```
liasun13.epfl.ch
(SunOS liasun13 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-2)
```

- The second bank in

```
covas.dc.fi.udc.es
(SunOS covas 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-Enterprise)
```

- The third bank in

```
ds.cesga.es
(SunOS ds 5.6 Generic sun4u sparc SUNW,Ultra-Enterprise)
```

- And the special bank again in

```
liasun13.epfl.ch
(SunOS liasun13 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-2)
```

5.4 Retagging with GALENA (3 different configurations)

As with BRILL, we have evaluated the GALENA system in different situations. The GALENA system does not allow us to use a lexicon different from the one integrated in it, but it is possible to define different configurations or specialization levels during the learning phase. For this, the user specifies the lexical category followed by all, some or none of the attributes present in the current tag set.

In this study we have chosen the 3 configurations that appear below. For each one, we show the lexical category followed by the active attributes:

- Configuration 1:
  - Substantive: Type, Gender.
  - Adjective: Number.
  - Demonstrative: Subtype, Gender.
  - Relative: Subtype, Gender.
  - Indefinite: Subtype, Gender.
  - Interrogative: Subtype, Gender.
  - Possessive: Subtype, Gender.
  - Numeral: Type, Subtype, Gender.
  - Article: Gender.
  - Personal pronoun: Type, Person.
  - Verb: Gender, Person.
  - Preposition.
  - Conjunction: Type.
  - Adverb: Type.
  - Interjection.
  - Punctuation mark: Type.
  - Peripheral: Type, Gender.
- Configuration 2:
  - Substantive: Gender, Number.
  - Adjective: Gender, Number.
  - Demonstrative: Gender, Number.
  - Relative: Gender, Number.
  - Indefinite: Gender, Number.
  - Interrogative: Gender, Number.
  - Possessive: Gender, Number.
  - Numeral: Gender, Number.
  - Article: Gender, Number.

  - Personal pronoun: Gender, Person.
  - Verb: Person.
  - Preposition.
  - Conjunction: Type.
  - Adverb: Type.
  - Interjection.
  - Punctuation mark: Type.
  - Peripheral: Gender, Number.

- Configuration 3:
  - Substantive: Gender, Number.
  - Adjective: Gender, Number.
  - Demonstrative: Gender, Number.
  - Relative: Subtype, Number.
  - Indefinite: Subtype, Number.
  - Interrogative: Subtype, Number.
  - Possessive: Gender, Number, Person.
  - Numeral: Type, Gender, Number.
  - Article: Gender, Number.
  - Personal pronoun: Type, Gender, Person.
  - Verb: Person, Person number.
  - Preposition.
  - Conjunction: Type.
  - Adverb: Type.
  - Interjection.
  - Punctuation mark: Type.
  - Peripheral: Type, Gender, Number.

All the experiments with the GALENA system have been performed on host

```
covas.dc.fi.udc.es
(SunOS covas 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-Enterprise)
```

Training times are very fast, as we can see in the tables of the next chapter. Compilation time is 00:09:20, but recompilation is mandatory only if you change the disambiguation configuration, or if you add new stems to the lexical database.

Chapter 6

Evaluation strategy

This chapter describes exactly what we are going to measure. First, in each test, when we compare the retagged text and the reference corpus, we calculate the following counters:

- OOVF+ (Out Of Vocabulary Form Successful)
- OOVF- (Out Of Vocabulary Form Failure)
- NAF+ (Non-Ambiguous Form Successful)
- NAF- (Non-Ambiguous Form Failure)
- AF+ (Ambiguous Form Successful)
- AF- (Ambiguous Form Failure)

From them, we also calculate:

- #OOVF (Number of Out Of Vocabulary Forms) = (OOVF+) + (OOVF-)
- #NAF (Number of Non-Ambiguous Forms) = (NAF+) + (NAF-)
- #AF (Number of Ambiguous Forms) = (AF+) + (AF-)
- #F (Number of Forms) = (#OOVF) + (#NAF) + (#AF)

And finally, we will use two rates, S1 (precision) and S2 (recall), in order to obtain a definitive idea about the performance of both tagging systems. The next section explains in detail what each rate is and how to calculate it.

6.1 Rates S1 (precision) and S2 (recall)

S1 (precision) measures the global level of success. Therefore, in general, it is calculated as:

$$S1\ (precision) = \frac{OK}{OK + ERROR} \times 100$$

and, in our case, as:

$$S1\ (precision) = \frac{(OOVF+) + (NAF+) + (AF+)}{\#F} \times 100$$

S2 (recall) is a rate that belongs to the terminology used in *information extraction*, and measures the level of success when the system has a real chance of success. In general, it is calculated as:

$$S2\,(recall) = \frac{OK + ERROR}{OK + ERROR + SILENCE} \times 100$$

and, in our case, as:

$$S2\,(recall) = \frac{(OOVF+) + (AF+)}{\#F - \#NAF} \times 100$$

A good tagging system should present a high value (close to 100, the highest value), not only in the rate of global success, but in both rates.

If the tagging system that we are evaluating does not present high values in S1 and S2, but is able to generate not only the most probable tag, but several tags sorted by probability from the highest to the lowest, then we can calculate three counters more:

- AFAT+ (Ambiguous Form Ambiguously Tagged Successful)
- AFAT- (Ambiguous Form Ambiguously Tagged Failure)
- #AFAT (Number of Ambiguous Forms Ambiguously Tagged) = (AFAT+) + (AFAT-)

and recalculate the rates by simply replacing AF counters with AFAT counters:

$$S1'\,(precision) = \frac{(OOVF+) + (NAF+) + (AFAT+)}{\#F} \times 100$$

$$S2'\,(recall) = \frac{(OOVF+) + (AFAT+)}{\#F - \#NAF} \times 100$$

If our original intention was not to use the system any more, but the new values of S1 and S2 are acceptable, then we can think about performing tasks such as reprogramming the system in order to improve the original performance, or filtering the output through another system.

The tools that have been evaluated in this work both present the property of generating several sorted taggings for each form. However, we have observed that the BRILL system spends a lot of extra training time if we active the n-best step and does not improve the values of the rates. Therefore, the calculus of rates S1' and S2' has been performed only for the GALENA system, taking into account the 3 best tags for each word.

6.2 Tables and graphics

The following pages show tables of counters, rates, training and tagging CPU times, sizes of training and reference corpora, etc., for each one of the 16 tests, and for each one of the "three" systems: BRILL, GALENA y GALENA 3 BEST TAGS. Each table is followed by a pair of figures which graphically represent the evolution of values S1 and S2. For Special Banks, the graphics show the average values of the three precedent banks and the values of the Special Bank *per se*. At the end of the section, two graphics (figure 6.57 and figure 6.58) conclude the study by showing the curves of the BRILL system in standard conditions (that is, with the lexicon *learned*

by itself), of the GALENA system with configuration 2 (the best one), and of the GALENA 3 BEST TAGS system with configuration 1 (also the best one in this case).

However, before introducing them, we present some useful discussions for the correct understanding of tables and graphics. The first question is: What do the percentages beside each counter value of BRILL tables represent? These percentages represent the distribution of successful and failure inside of each counter category: OOVF, NAF or AF. For instance, in test11 with BRILL LEXICON, OOVF+ is 24235 and OOVF- is 14315. That is, OOVF+ is the 62.87% of all the out of vocabulary forms in this test, and OOVF- is the remaining 37.13%. Therefore, by looking at these distribution percentages throughout the table, we can obtain and idea about the *local* performance of each test in regard to out of vocabulary forms, non-ambiguous forms or ambiguous forms, instead of the *global* performance represented by S1 and S2. And we can see that a test can be *globally* worse than another one, but *locally* better for example in regard to ambiguous forms.

Another important question is: Why NAF- is 0 when there are no out of vocabulary forms? Which is the same question as: Why NAF- is different from 0 in other conditions? Once again, let us consider test11. For each non-ambiguous form f in the BRILL LEXICON, f/t_i appears in TC0, or in TC11, or in both, but always with the tag t_i. We retag R1 and obtain R11. All the forms f in R11 will be tagged with t_i, but f/t_j, where $t_i \neq t_j$, could appear in R1. That means that f is a form potentially ambiguous, but is non-ambiguous in regard to the BRILL LEXICON, is a failure, and therefore is a NAF-. However, when we work with the ITU LEXICON, f is present in the lexicon at least with both tags t_i and t_j. We retag R1 and obtain R11, and we will have either a successful or a failure, but it will be either an AF+ or an AF-, because now f is an ambiguous form, an this always yields NAF- equal to 0. Therefore, we deduce that BRILL system use contextual rules only to change the first tag of ambiguous forms, that is, assumes all non-ambiguous forms as non-potentially ambiguous. This could seem a very strong assumption, but if we look at the local distribution percentages for non-ambiguous forms, we can see that they actually are good. Therefore, to change this hypothesis could dramatically decrease the performance. It could be very interesting to know how many NAF- words are changed in AF+ and how many in AF-, or even what kinds of changes happen in the other counters. For this reason, in each bank of experiments of BRILL system, a set of word transitions graphs can be found immediately after the table. In these graphs, each node contains the name of one of the counters, the value of this counter in the test with the BRILL LEXICON, and its new value with the extended lexicon, in this order. And each label in the arrows represents how many words change from the counter in the source node to the counter in the target node. Thick arrows go always from "failure" nodes to "successful" nodes, representing the gain in the tagging process with the extended lexicon.

Distribution percentages and word transitions graphs have not been calculated for GALENA system, because variations of values of counters among different configurations are very low.

System: BRILL - Bank of Experiments: 1					
Experiment	test11	test12	test13	test14	test15
Host	liasun13	liasun13	liasun13	liasun13	liasun13
Training Time	03:40:08	08:53:11	14:23:03	21:12:40	27:33:28
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	32774	68322	96775	126302	162521
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	324780	324780	324780	324780	324780
Lexicon: BRILL					
Tagging Time	00:01:15	00:01:37	00:01:52	00:02:11	00:02:26
OOVF+	24235 62.87%	14862 65.32%	11375 70.90%	9330 73.85%	7997 78.01%
OOVF-	14315 37.13%	7892 34.68%	4669 29.10%	3303 26.15%	2254 21.99%
NAF+	172907 98.58%	181111 98.84%	185763 99.13%	187425 99.32%	173623 99.47%
NAF-	2498 1.42%	2120 1.16%	1627 0.87%	1280 0.68%	929 0.53%
AF+	104215 94.04%	113103 95.21%	115850 95.47%	118234 95.78%	135045 96.48%
AF-	6610 5.96%	5692 4.79%	5496 4.53%	5208 4.22%	4932 3.52%
S1	92.788	95.164	96.369	96.985	97.501
S2	85.991	90.403	92.601	93.745	95.216
Lexicon: BRILL+GALENA					
Tagging Time	00:04:59	00:05:27	00:05:43	00:06:03	00:06:17
OOVF+	13037 61.02%	9006 64.18%	7391 72.40%	6194 76.82%	5615 80.88%
OOVF-	8329 38.98%	5026 35.82%	2817 27.60%	1869 23.18%	1327 19.12%
NAF+	159427 98.20%	161919 98.78%	165034 99.21%	166431 99.40%	152679 99.54%
NAF-	2926 1.80%	2003 1.22%	1314 0.79%	1000 0.60%	709 0.46%
AF+	131088 92.93%	138774 94.52%	140989 95.12%	142888 95.71%	158611 96.45%
AF-	9973 7.07%	8052 5.48%	7235 4.88%	6398 4.29%	5839 3.55%
S1	93.463	95.356	96.500	97.146	97.575
S2	88.732	91.869	93.655	94.746	95.818
Lexicon: BRILL+ITU					
Tagging Time	00:01:16	00:01:32	00:01:49	00:02:06	00:02:20
OOVF+	0 -	0 -	0 -	0 -	0 -
OOVF-	0 -	0 -	0 -	0 -	0 -
NAF+	171441 100%	171441 100%	171441 100%	171441 100%	171441 100%
NAF-	0 0%	0 0%	0 0%	0 0%	0 0%
AF+	143027 93.28%	145154 94.66%	146045 95.24%	146826 95.75%	147514 96.20%
AF-	10312 6.72%	8185 5.34%	7294 4.76%	6513 4.25%	5825 3.80%
S1	96.824	97.479	97.754	97.994	98.206
S2	93.275	94.662	95.243	95.752	96.201
Lexicon: BRILL+ITU+GALENA					
Tagging Time	00:05:14	00:05:20	00:05:39	00:06:03	00:06:12
OOVF+	0 -	0 -	0 -	0 -	0 -
OOVF-	0 -	0 -	0 -	0 -	0 -
NAF+	151816 100%	151816 100%	151816 100%	151816 100%	151816 100%
NAF-	0 0%	0 0%	0 0%	0 0%	0 0%
AF+	162547 93.98%	164455 95.08%	165453 95.66%	166336 96.17%	167009 96.56%
AF-	10417 6.02%	8509 4.92%	7511 4.34%	6628 3.83%	5955 3.44%
S1	96.792	97.380	97.687	97.959	98.166
S2	93.977	95.080	95.657	96.167	96.557

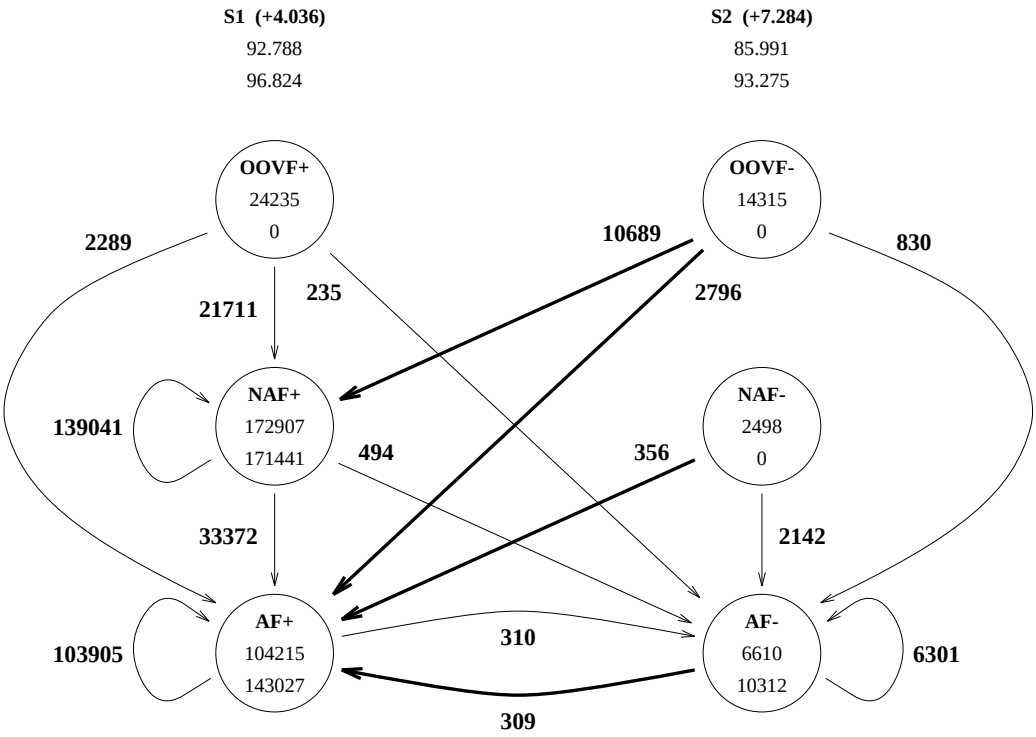


Figure 6.1: System: BRILL - Test: test11 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

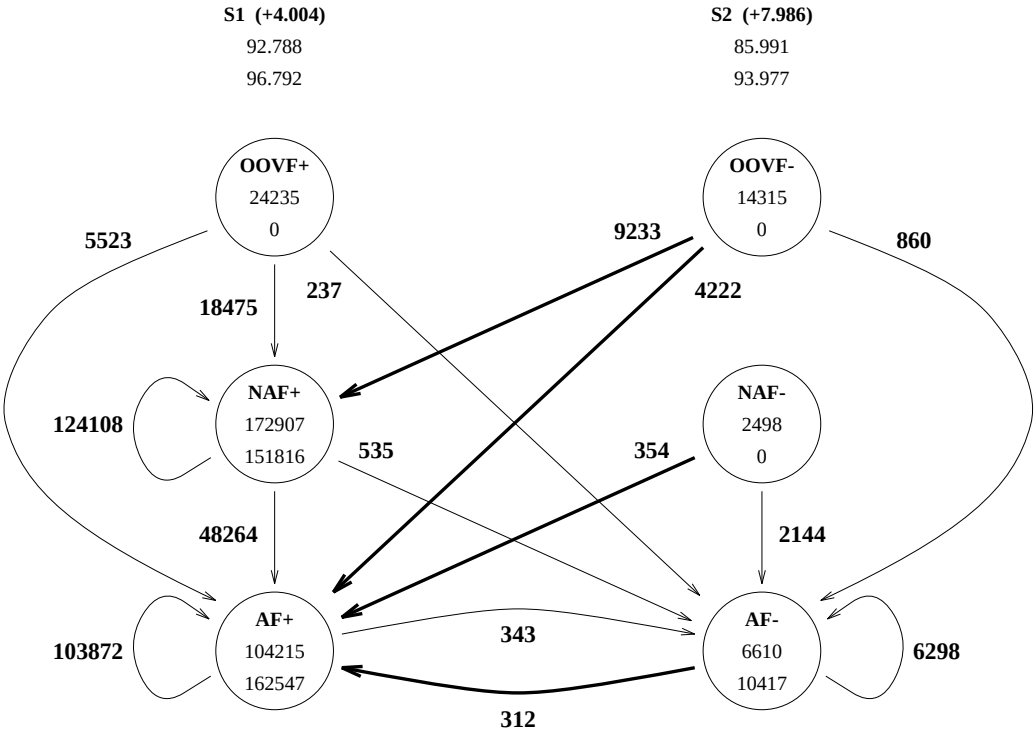


Figure 6.2: System: BRILL - Test: test11 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

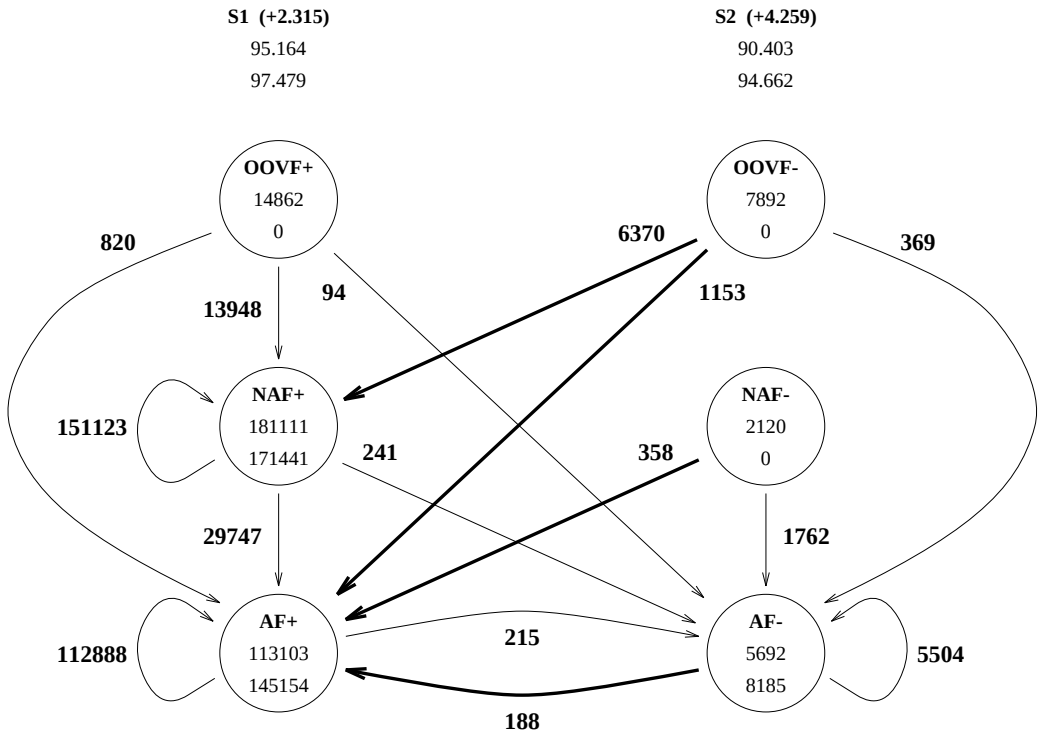


Figure 6.3: System: BRILL - Test: test12 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

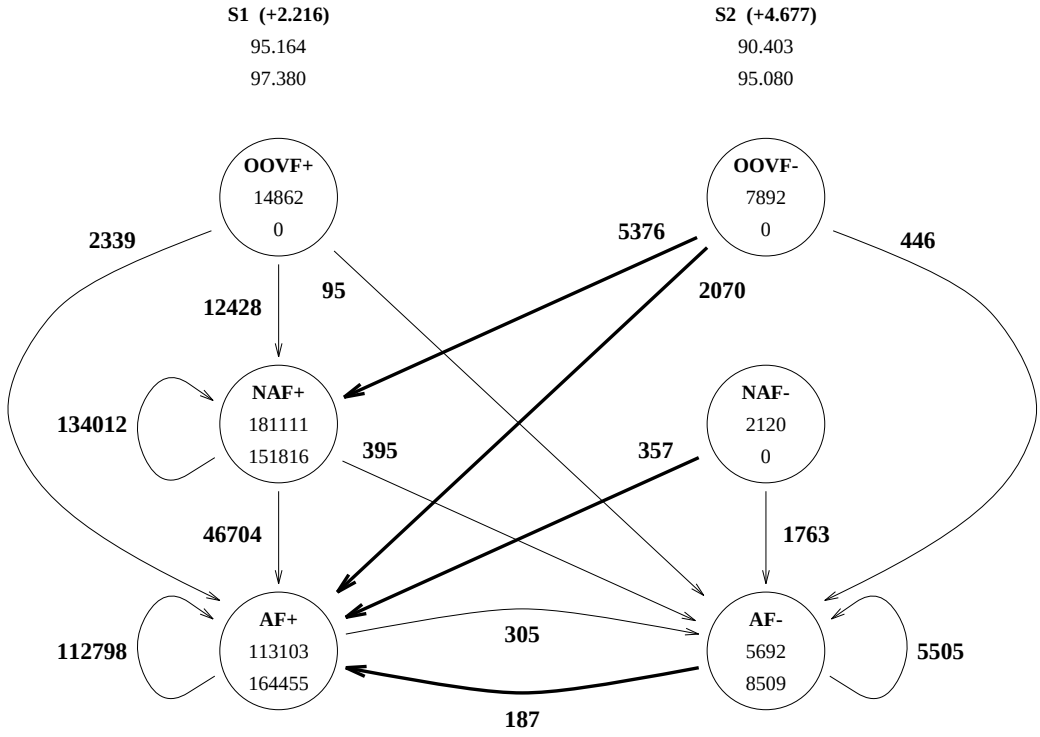


Figure 6.4: System: BRILL - Test: test12 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

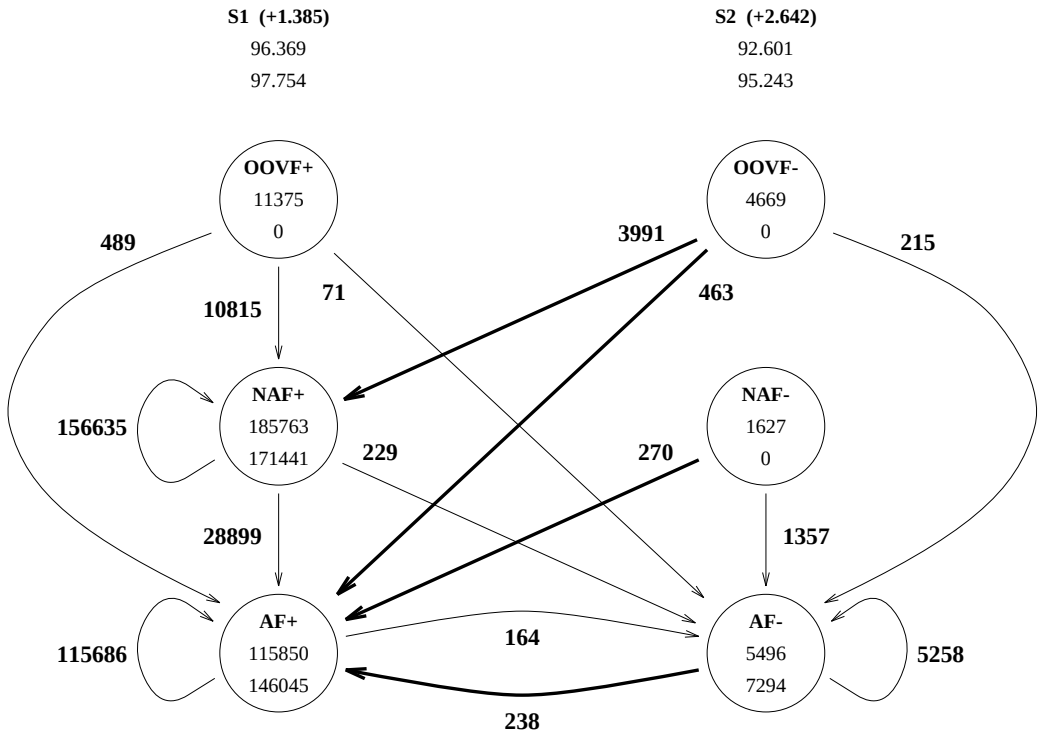


Figure 6.5: System: BRILL - Test: test13 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

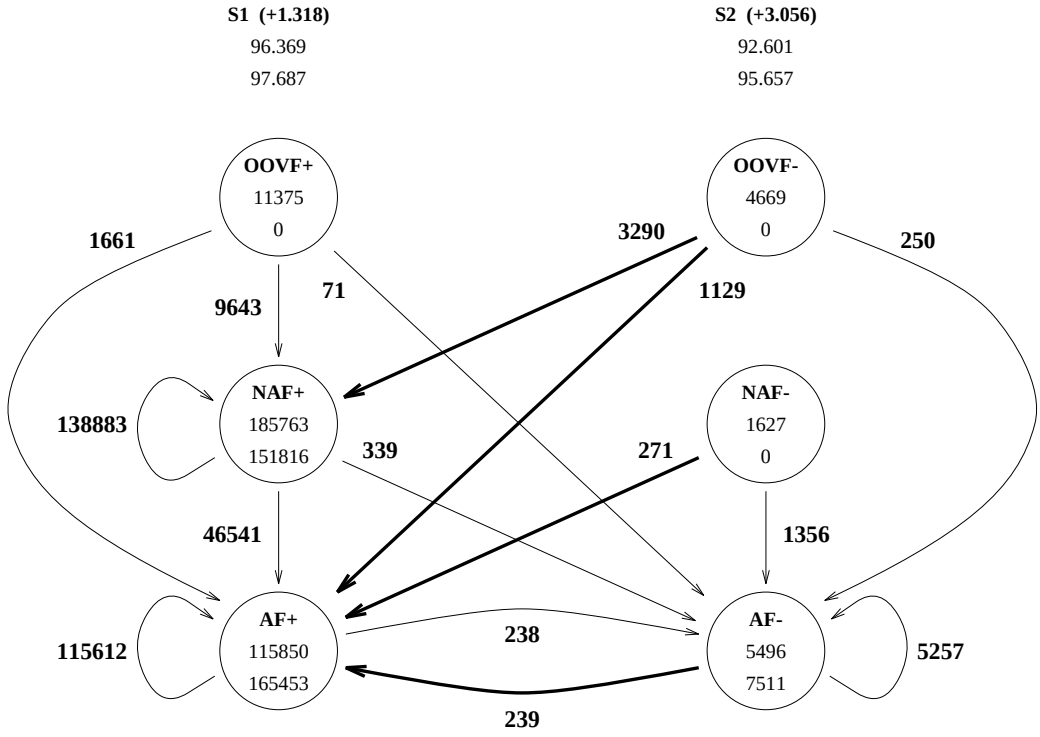


Figure 6.6: System: BRILL - Test: test13 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

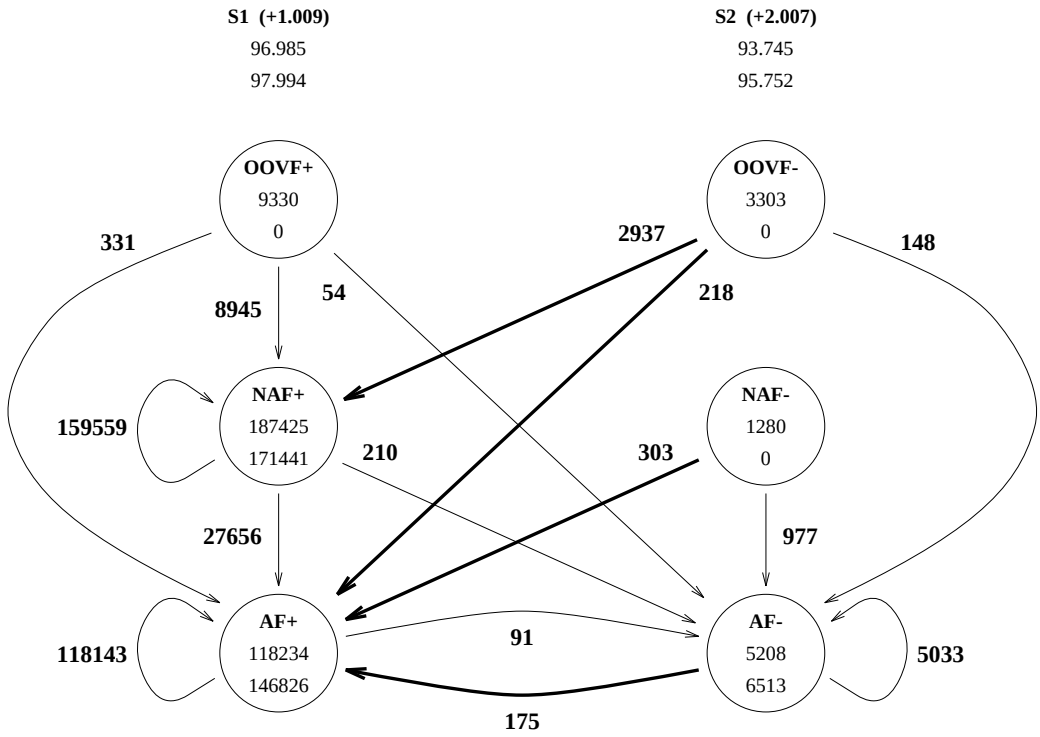


Figure 6.7: System: BRILL - Test: test14 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

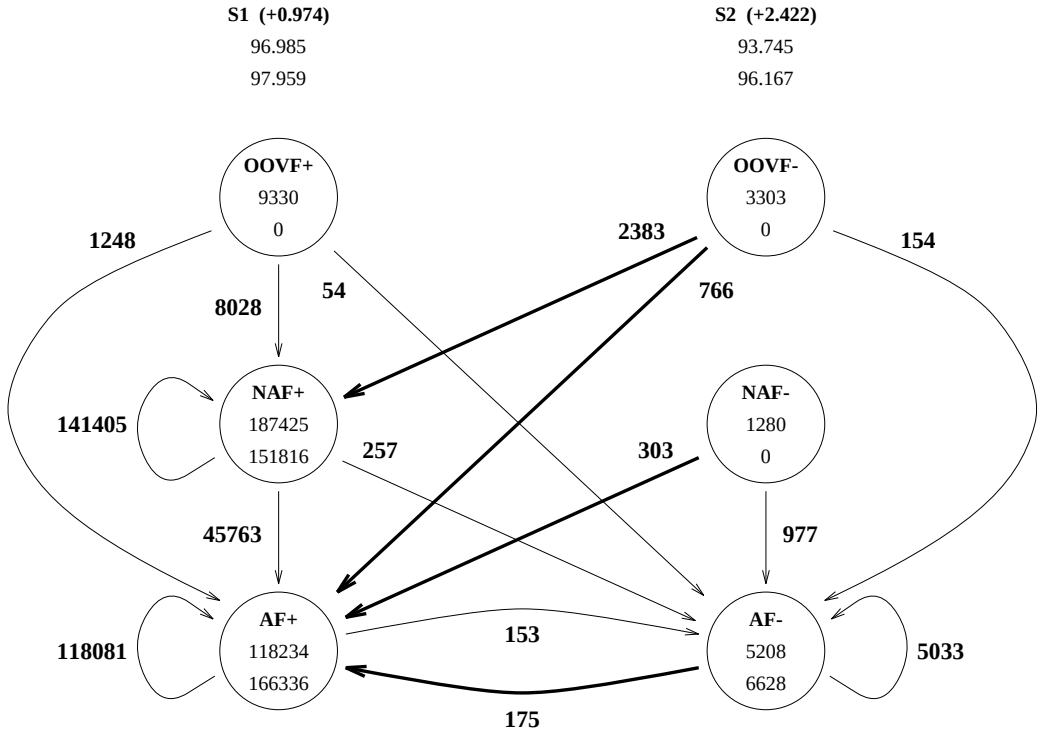


Figure 6.8: System: BRILL - Test: test14 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

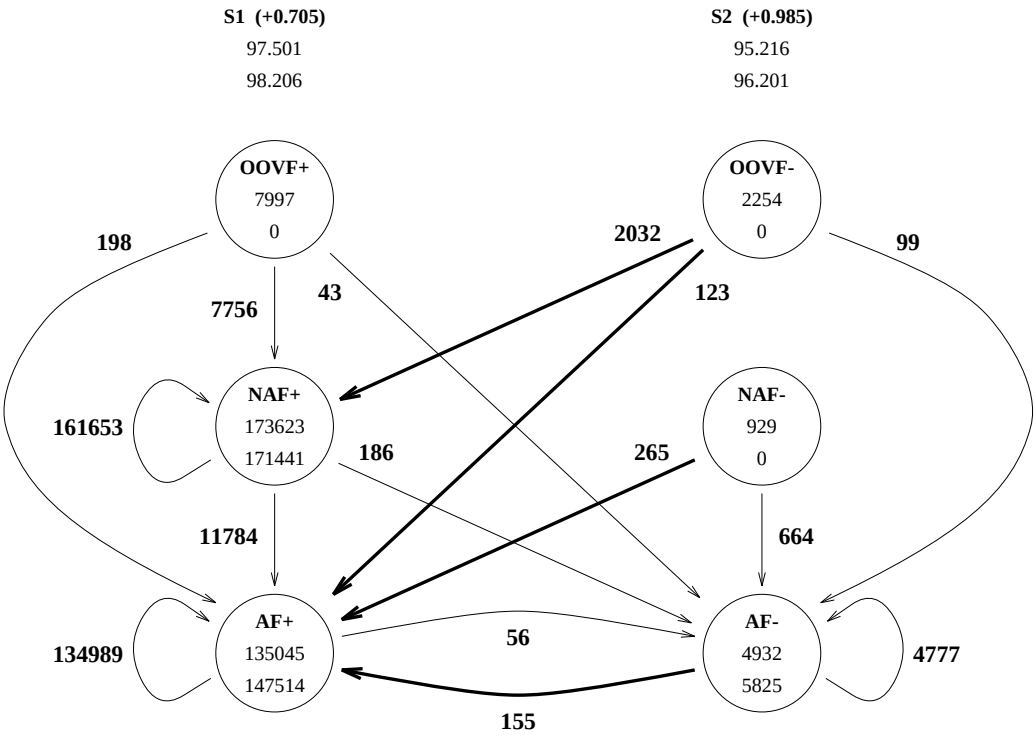


Figure 6.9: System: BRILL - Test: test15 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

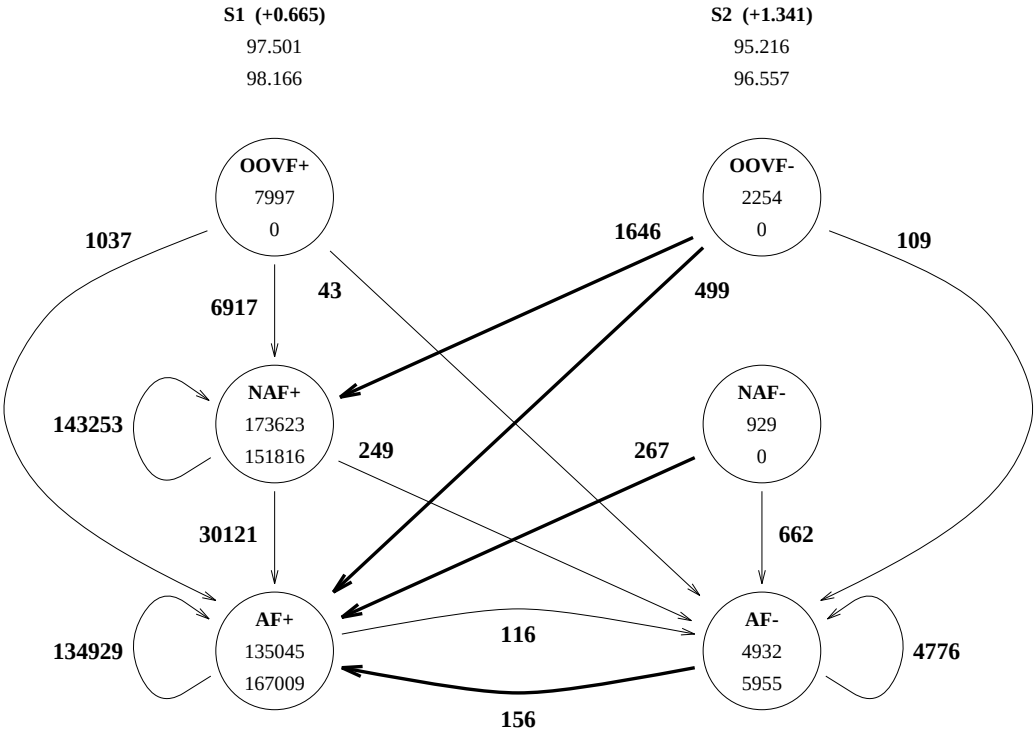


Figure 6.10: System: BRILL - Test: test15 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

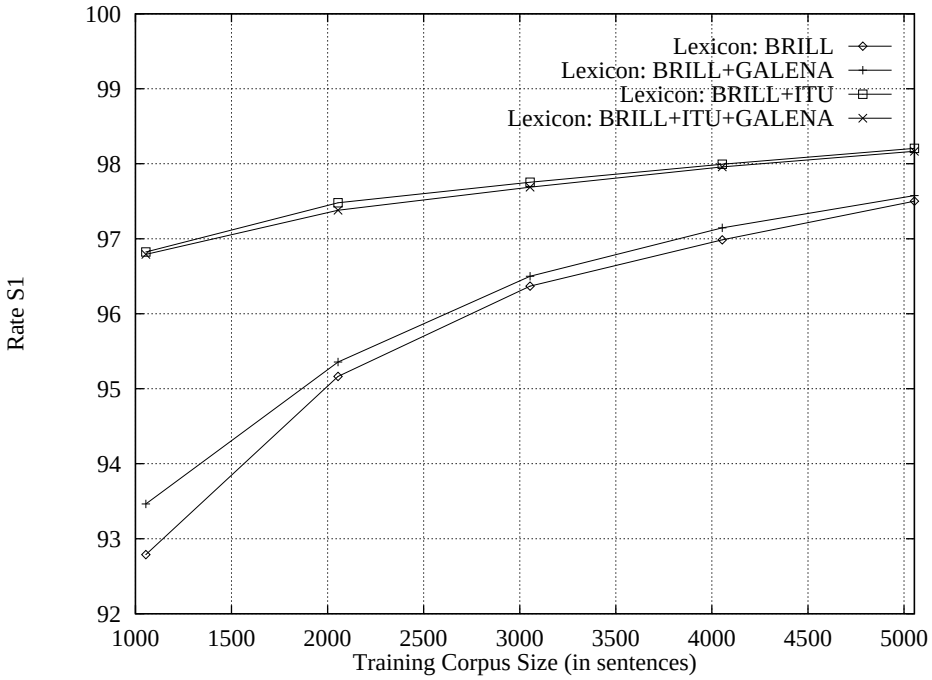


Figure 6.11: System: BRILL - Bank of Experiments: 1 - Rate: S1

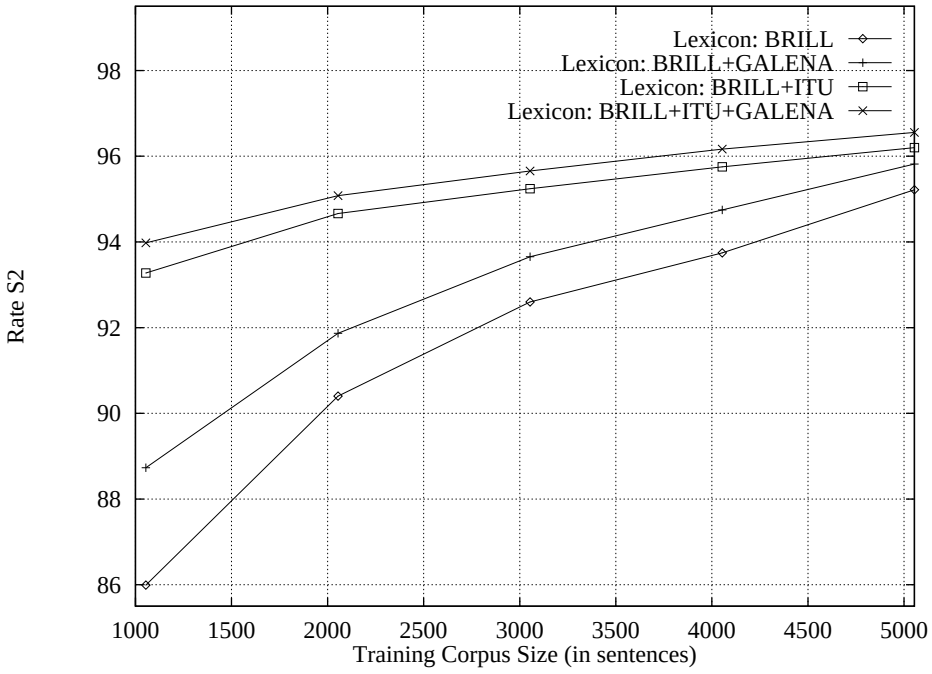


Figure 6.12: System: BRILL - Bank of Experiments: 1 - Rate: S2

System: BRILL - Bank of Experiments: 2										
Experiment	test21		test22		test23		test24		test25	
Host	covas		covas		covas		covas		covas	
Training Time	03:55:12		08:49:43		17:04:23		24:10:13		31:22:33	
Training Corpus Size										
Sentences	1054		2054		3054		4054		5054	
Words	33192		71367		100218		129764		166064	
Reference Corpus Size										
Sentences	9919		9919		9919		9919		9919	
Words	321237		321237		321237		321237		321237	
Lexicon: BRILL										
Tagging Time	00:01:26		00:01:49		00:02:13		00:02:41		00:02:51	
OOVF+	23216	61.06%	14198	64.44%	10966	70.18%	9073	72.79%	7870	77.14%
OOVF-	14804	38.94%	7834	35.56%	4660	29.82%	3392	27.21%	2332	22.86%
NAF+	169564	98.62%	178135	98.74%	182740	99.10%	184074	99.31%	183836	99.48%
NAF-	2380	1.38%	2269	1.26%	1666	0.90%	1282	0.69%	961	0.52%
AF+	104272	93.71%	113529	95.56%	115823	95.56%	118401	95.94%	121166	95.98%
AF-	7001	6.29%	5272	4.44%	5382	4.44%	5015	4.06%	5072	4.02%
S1	92.471		95.213		96.355		96.983		97.396	
S2	85.394		90.693		92.661		93.812		94.573	
Lexicon: BRILL+GALENA										
Tagging Time	00:06:17		00:06:30		00:06:58		00:07:51		00:07:32	
OOVF+	13051	61.30%	8835	63.24%	7413	71.94%	6213	75.14%	5443	79.08%
OOVF-	8241	38.70%	5136	36.76%	2892	28.06%	2056	24.86%	1440	20.92%
NAF+	156261	97.92%	158642	98.70%	161829	99.17%	163115	99.40%	162799	99.58%
NAF-	3323	2.08%	2088	1.30%	1350	0.83%	980	0.60%	681	0.42%
AF+	130087	92.68%	139250	95.03%	140749	95.26%	142579	95.77%	144753	95.94%
AF-	10274	7.32%	7286	4.97%	7004	4.74%	6294	4.23%	6121	4.06%
S1	93.201		95.483		96.499		97.095		97.434	
S2	88.546		92.260		93.739		94.686		95.207	
Lexicon: BRILL+ITU										
Tagging Time	00:01:24		00:01:50		00:02:11		00:02:32		00:02:45	
OOVF+	0	-	0	-	0	-	0	-	0	-
OOVF-	0	-	0	-	0	-	0	-	0	-
NAF+	169795	100%	169795	100%	169795	100%	169795	100%	169795	100%
NAF-	0	0%	0	0%	0	0%	0	0%	0	0%
AF+	141289	93.30%	143878	95.00%	143281	94.61%	144977	95.73%	145603	96.14%
AF-	10153	6.70%	7564	5.00%	8161	5.39%	6465	4.27%	5839	3.86%
S1	96.839		97.645		97.459		97.987		98.182	
S2	93.295		95.005		94.611		95.731		96.144	
Lexicon: BRILL+ITU+GALENA										
Tagging Time	00:06:03		00:06:34		00:06:57		00:07:53		00:07:37	
OOVF+	0	-	0	-	0	-	0	-	0	-
OOVF-	0	-	0	-	0	-	0	-	0	-
NAF+	150485	100%	150485	100%	150485	100%	150485	100%	150485	100%
NAF-	0	0%	0	0%	0	0%	0	0%	0	0%
AF+	160451	93.97%	163075	95.50%	162456	95.14%	164079	96.09%	164693	96.45%
AF-	10301	6.03%	7677	4.50%	8296	4.86%	6673	3.91%	6059	3.55%
S1	96.793		97.610		97.417		97.922		98.113	
S2	93.967		95.504		95.141		96.091		96.451	

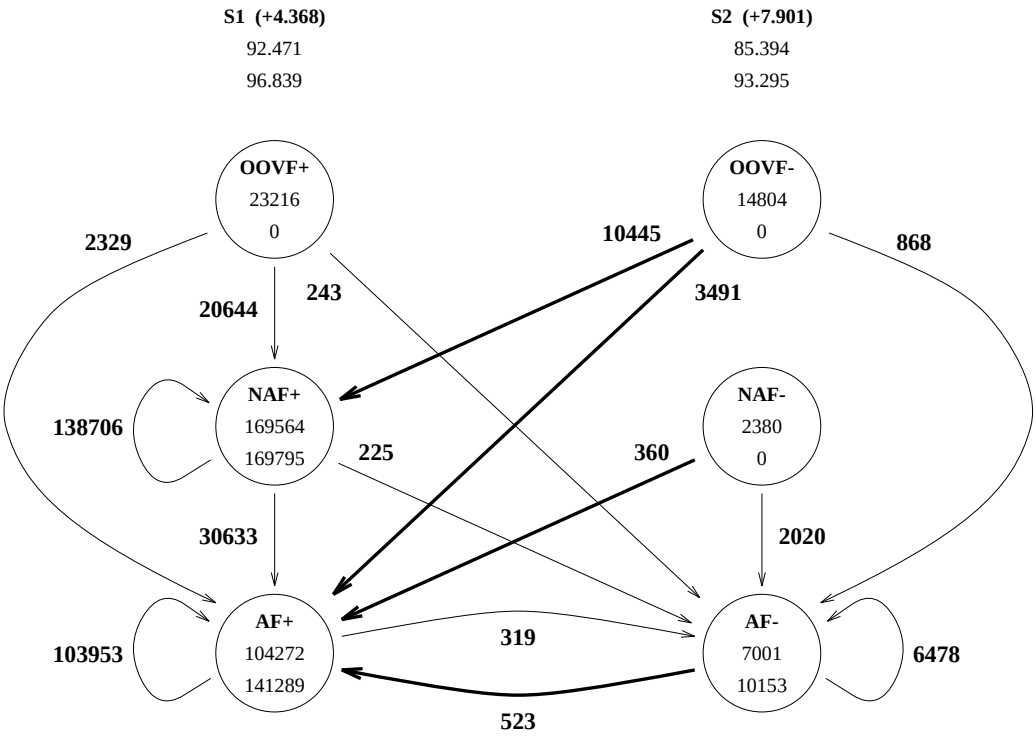


Figure 6.13: System: BRILL - Test: test21 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

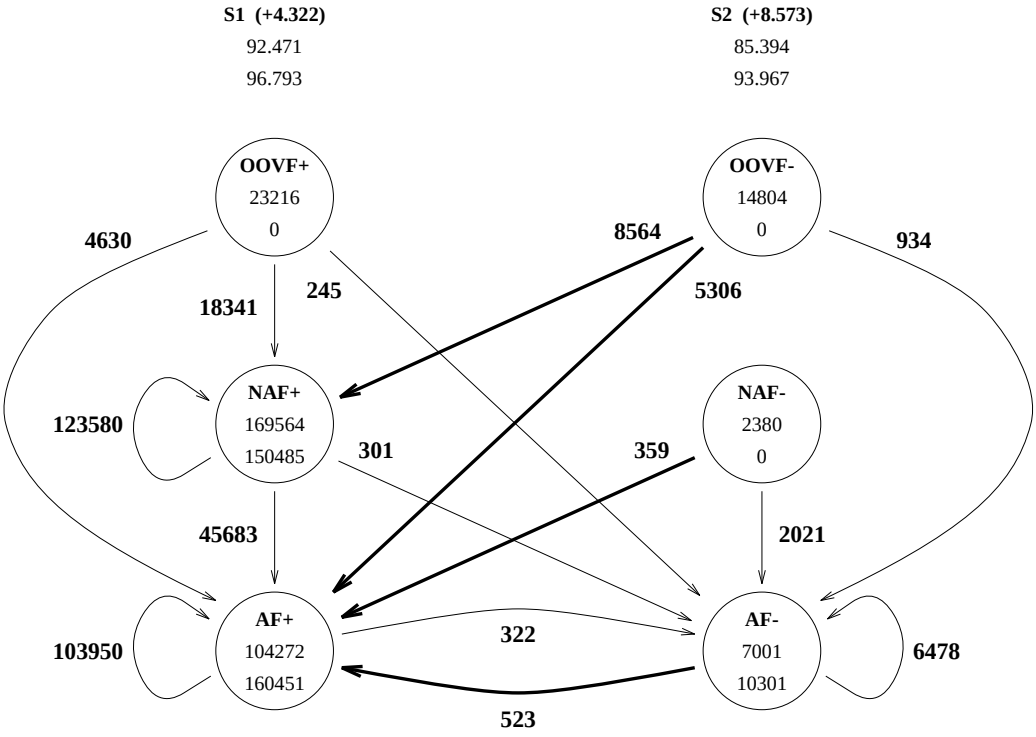


Figure 6.14: System: BRILL - Test: test21 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

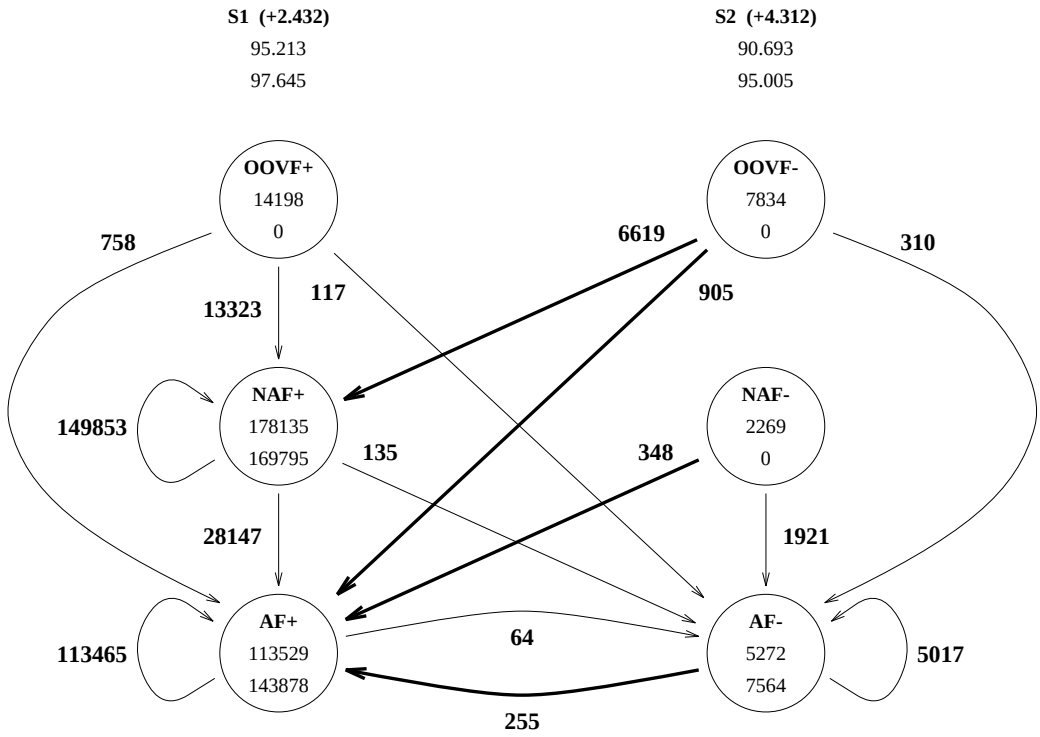


Figure 6.15: System: BRILL - Test: test22 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

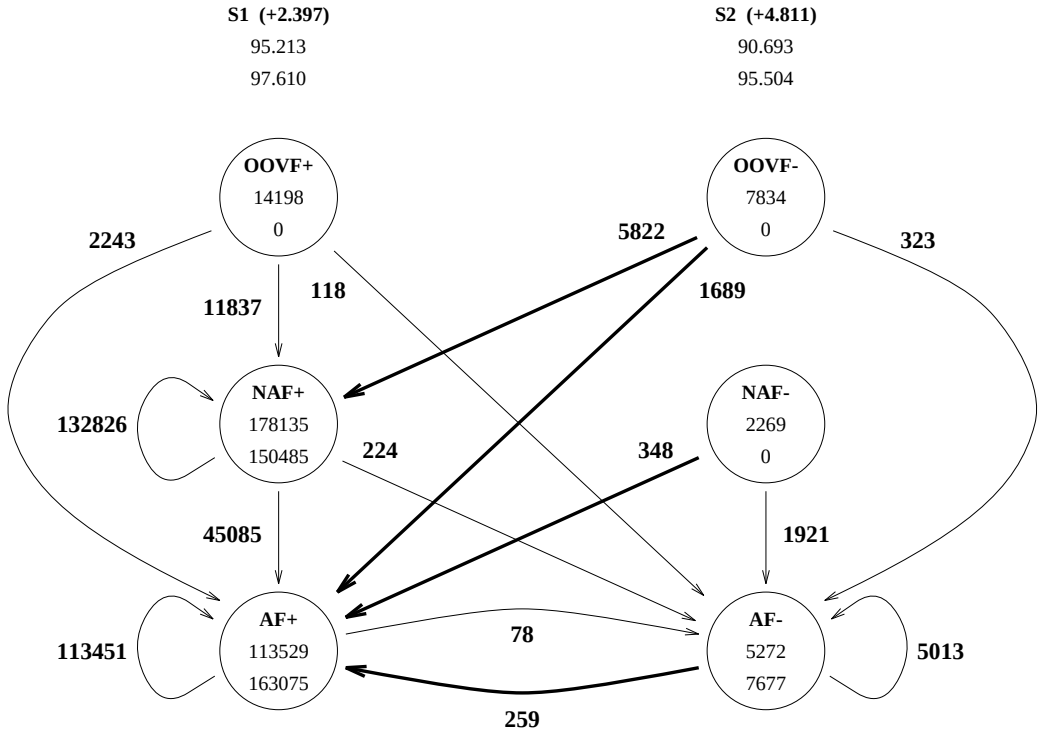


Figure 6.16: System: BRILL - Test: test22 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

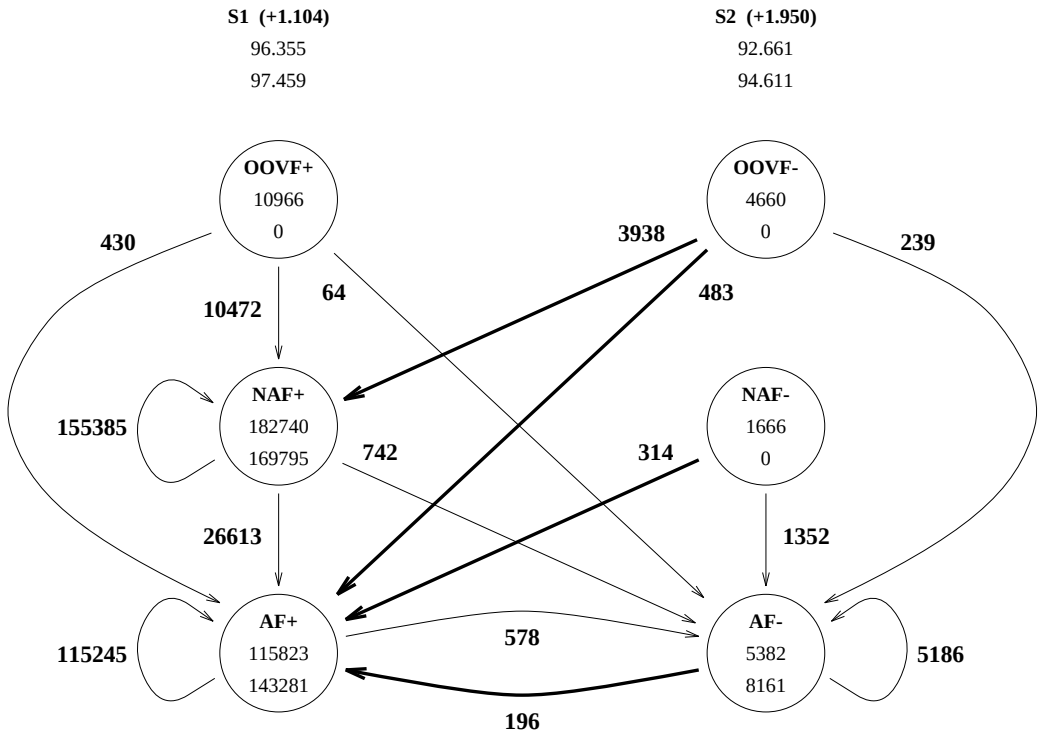


Figure 6.17: System: BRILL - Test: test23 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

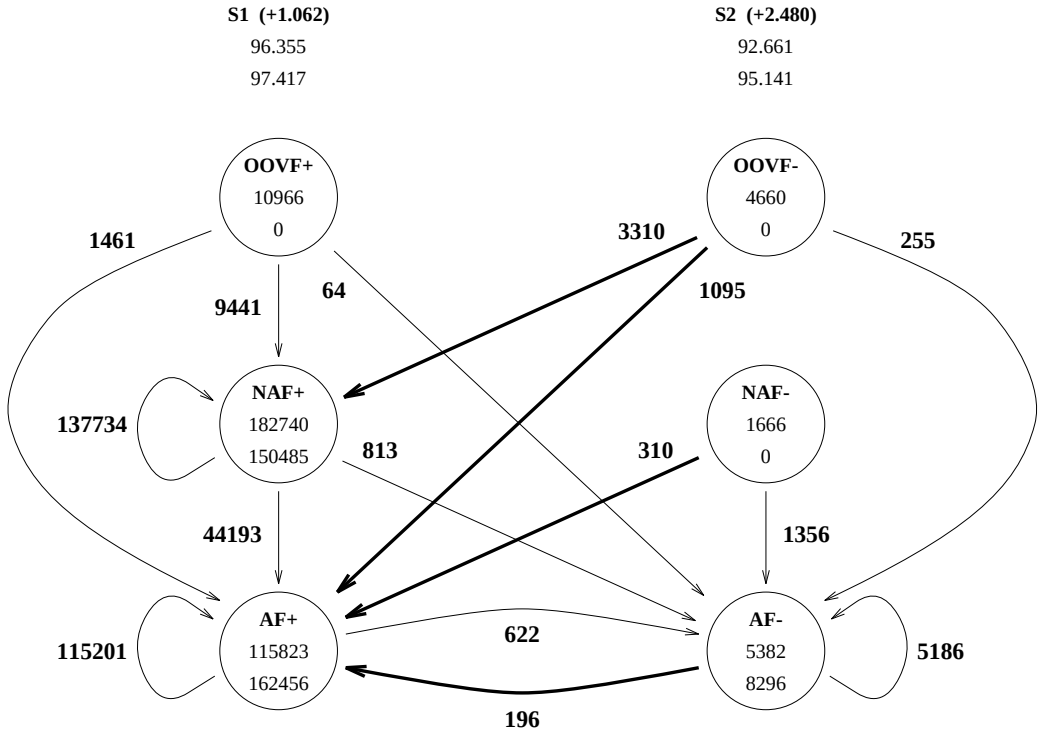


Figure 6.18: System: BRILL - Test: test23 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

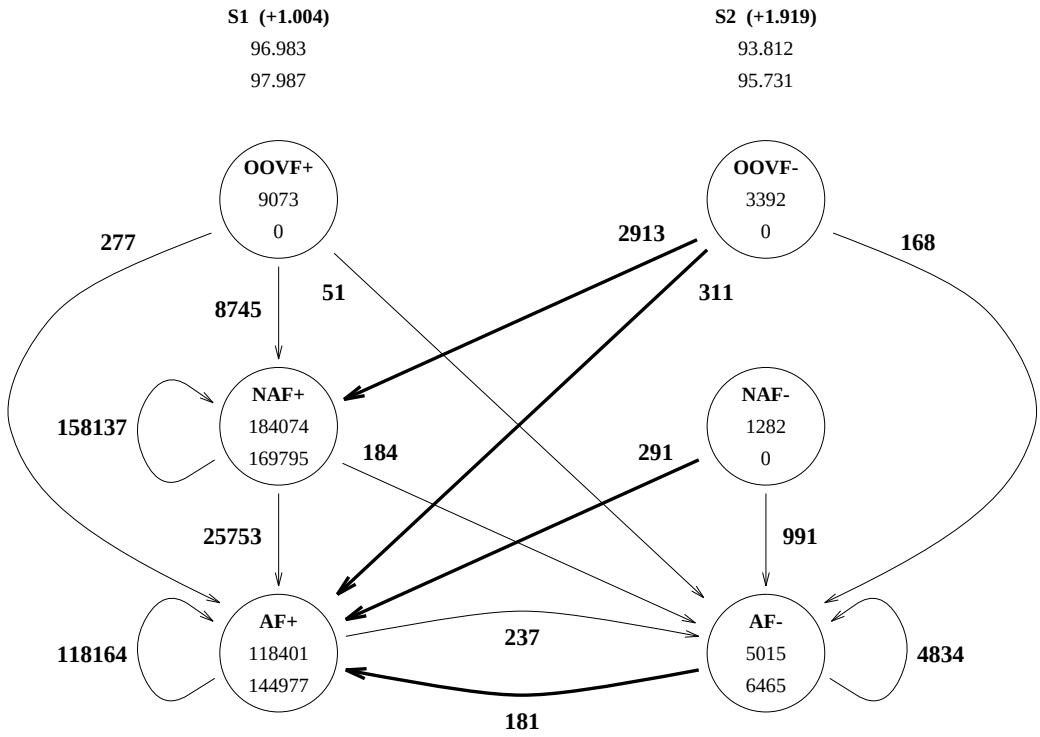


Figure 6.19: System: BRILL - Test: test24 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

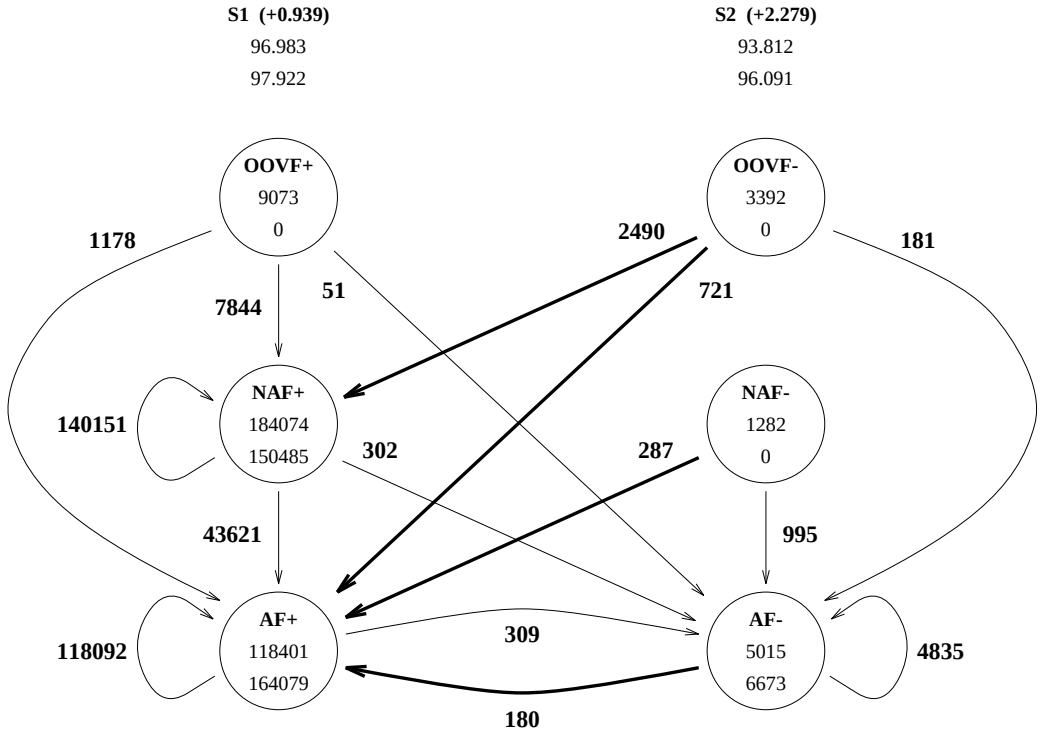


Figure 6.20: System: BRILL - Test: test24 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

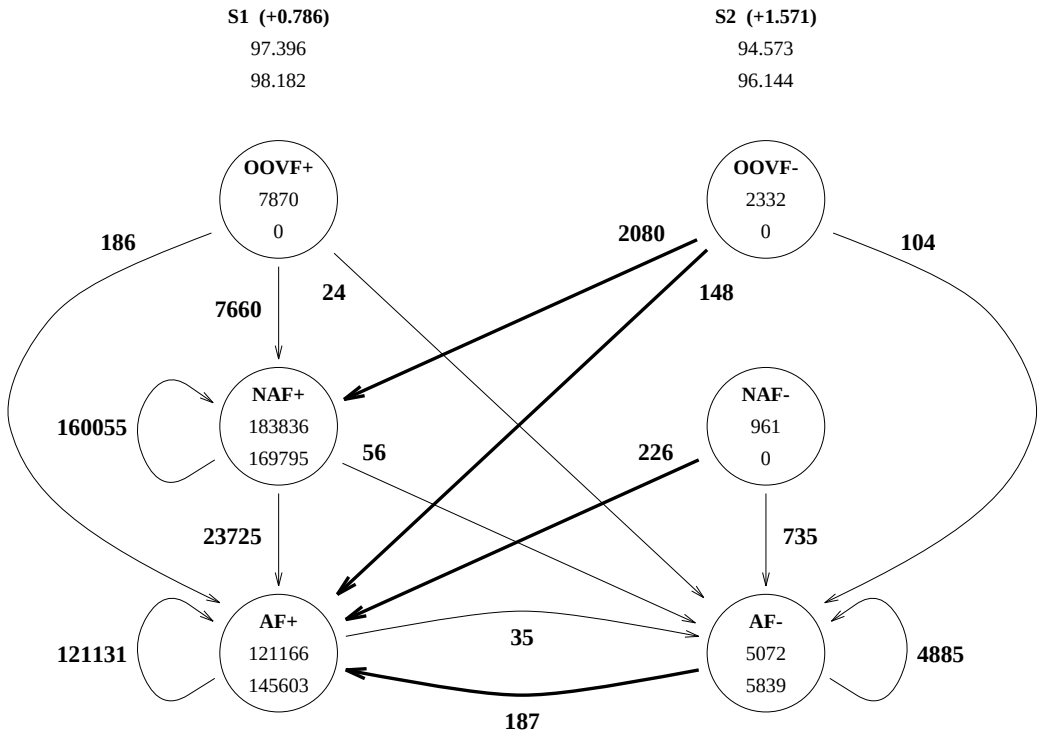


Figure 6.21: System: BRILL - Test: test25 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

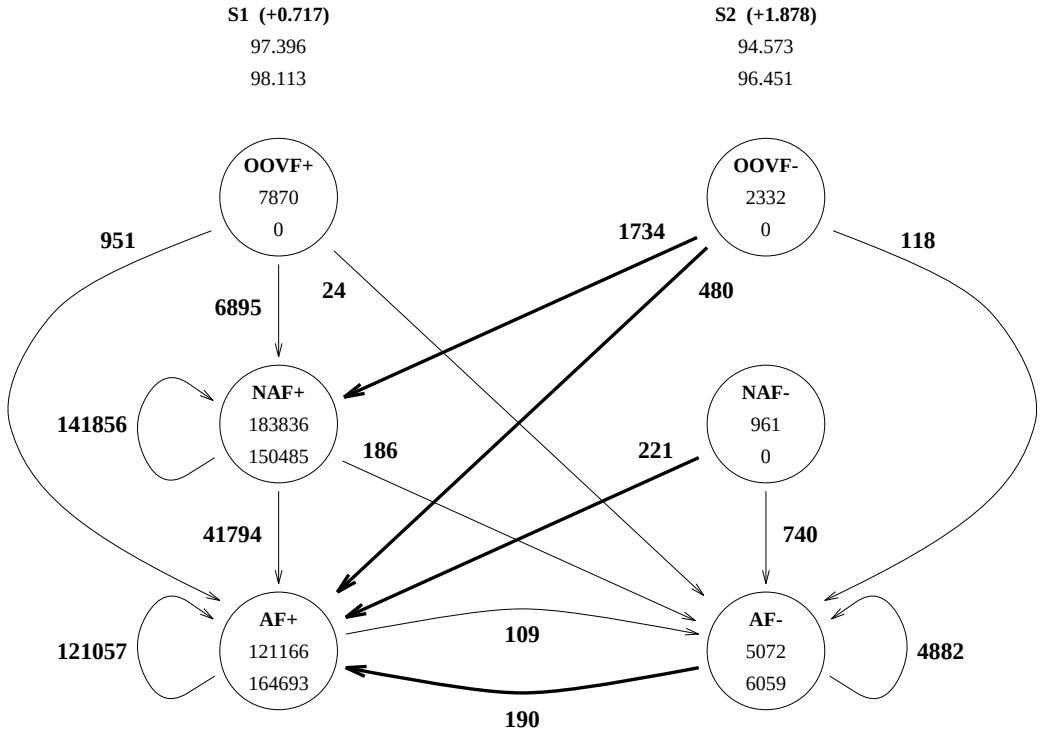


Figure 6.22: System: BRILL - Test: test25 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

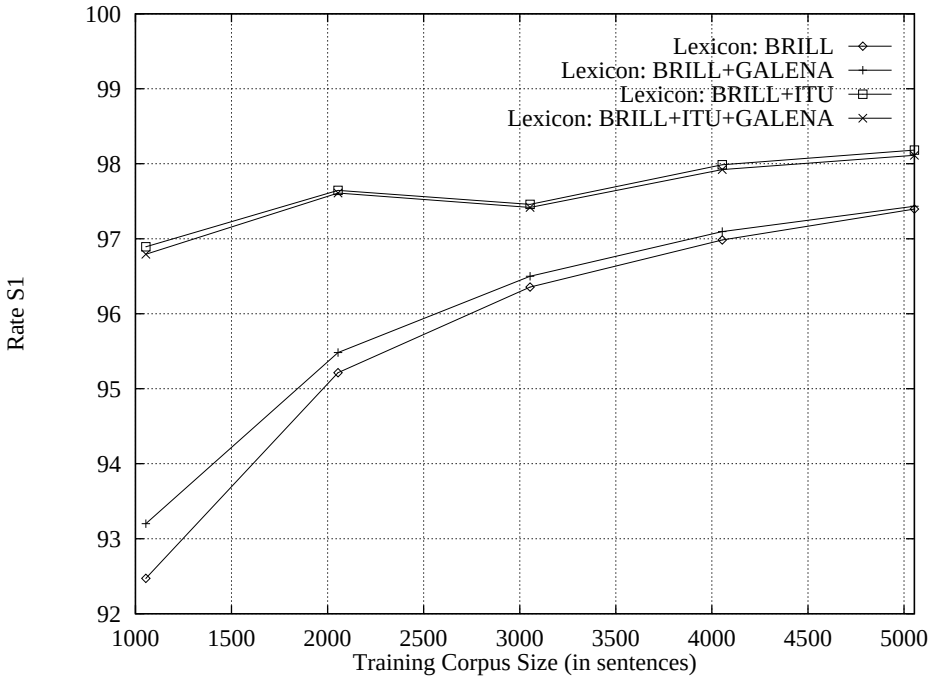


Figure 6.23: System: BRILL - Bank of Experiments: 2 - Rate: S1

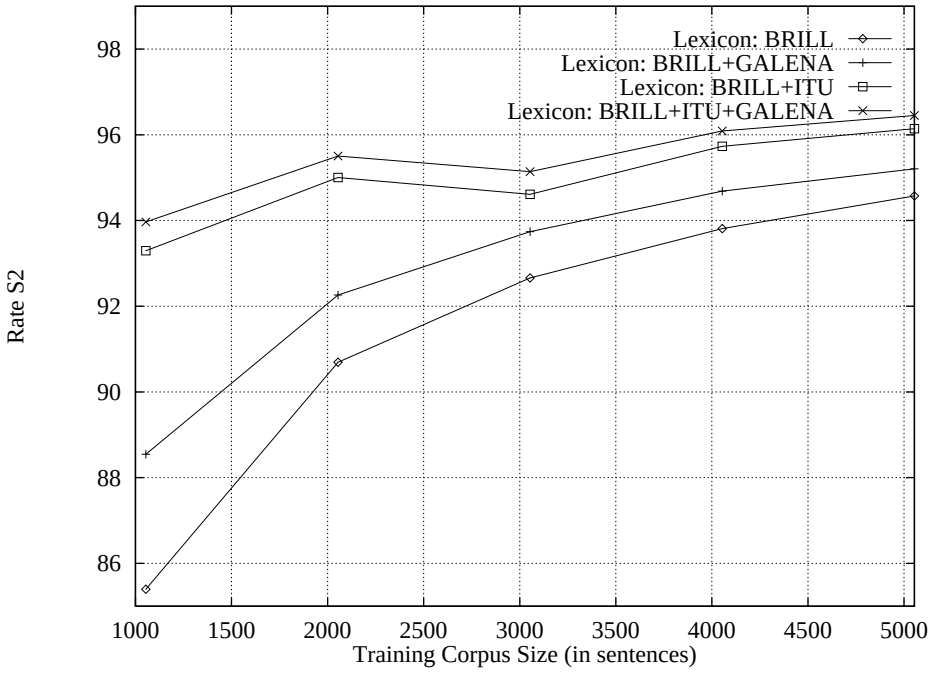


Figure 6.24: System: BRILL - Bank of Experiments: 2 - Rate: S2

System: BRILL - Bank of Experiments: 3					
Experiment	test31	test32	test33	test34	test35
Host	ds.cesga	ds.cesga	ds.cesga	ds.cesga	ds.cesga
Training Time	03:18:40	08:49:10	15:19:08	23:05:48	28:56:27
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	33284	69764	97557	127230	162819
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	324482	324482	324482	324482	324482
Lexicon: BRILL					
Tagging Time	00:01:24	00:01:54	00:02:08	00:02:29	00:02:46
OOVF+	25187 64.68%	14451 63.51%	11769 72.00%	9601 75.55%	8119 77.45%
OOVF-	13753 35.32%	8303 36.49%	4577 28.00%	3107 24.45%	2364 22.55%
NAF+	173063 98.75%	178555 98.79%	182177 99.10%	184065 99.24%	182840 99.42%
NAF-	2194 1.25%	2193 1.21%	1652 0.90%	1412 0.76%	1061 0.58%
AF+	103575 93.92%	115541 95.50%	118960 95.70%	121369 96.10%	125189 96.22%
AF-	6710 6.08%	5439 4.50%	5347 4.30%	4928 3.90%	4909 3.78%
S1	93.017	95.089	96.432	97.088	97.431
S2	86.287	90.439	92.944	94.219	94.826
Lexicon: BRILL+GALENA					
Tagging Time	00:06:00	00:06:08	00:06:31	00:06:46	00:07:06
OOVF+	14140 63.60%	9138 64.23%	7942 74.15%	6566 77.17%	5683 78.57%
OOVF-	8093 36.40%	5090 35.77%	2769 25.85%	1943 22.83%	1550 21.43%
NAF+	159925 98.22%	159740 98.67%	161768 99.09%	163181 99.35%	162715 99.53%
NAF-	2897 1.78%	2161 1.33%	1490 0.91%	1064 0.65%	763 0.47%
AF+	129652 92.99%	140882 94.96%	143782 95.53%	145602 95.96%	147834 96.14%
AF-	9775 7.01%	7471 5.04%	6731 4.47%	6126 4.03%	5937 3.86%
S1	93.600	95.462	96.613	97.185	97.457
S2	88.947	92.274	94.107	94.964	95.349
Lexicon: BRILL+ITU					
Tagging Time	00:01:26	00:01:54	00:02:10	00:02:26	00:02:40
OOVF+	0 -	0 -	0 -	0 -	0 -
OOVF-	0 -	0 -	0 -	0 -	0 -
NAF+	171476 100%	171476 100%	171476 100%	171476 100%	171476 100%
NAF-	0 0%	0 0%	0 0%	0 0%	0 0%
AF+	143546 93.82%	145167 94.88%	146135 95.51%	146572 95.79%	147135 96.17%
AF-	9460 6.18%	7839 5.12%	6871 4.49%	6434 4.21%	5871 3.83%
S1	97.084	97.584	97.882	98.017	98.190
S2	93.817	94.876	95.509	95.794	96.162
Lexicon: BRILL+ITU+GALENA					
Tagging Time	00:06:11	00:06:13	00:06:30	00:06:52	00:07:08
OOVF+	0 -	0 -	0 -	0 -	0 -
OOVF-	0 -	0 -	0 -	0 -	0 -
NAF+	152150 100%	152150 100%	152150 100%	152150 100%	152150 100%
NAF-	0 0%	0 0%	0 0%	0 0%	0 0%
AF+	162655 94.38%	164310 95.35%	165315 95.93%	165785 96.20%	166301 96.50%
AF-	9677 5.62%	8022 4.65%	7017 4.07%	6547 3.80%	6031 3.50%
S1	97.017	97.527	97.837	97.982	98.141
S2	94.384	95.345	95.928	96.200	96.500

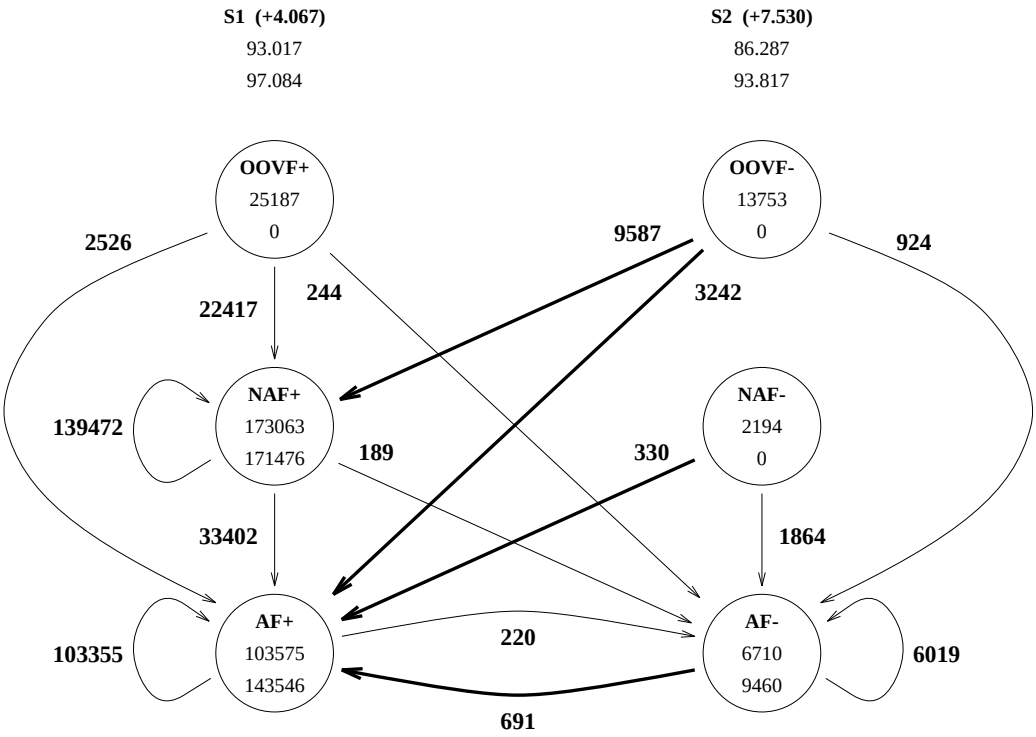


Figure 6.25: System: BRILL - Test: test31 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

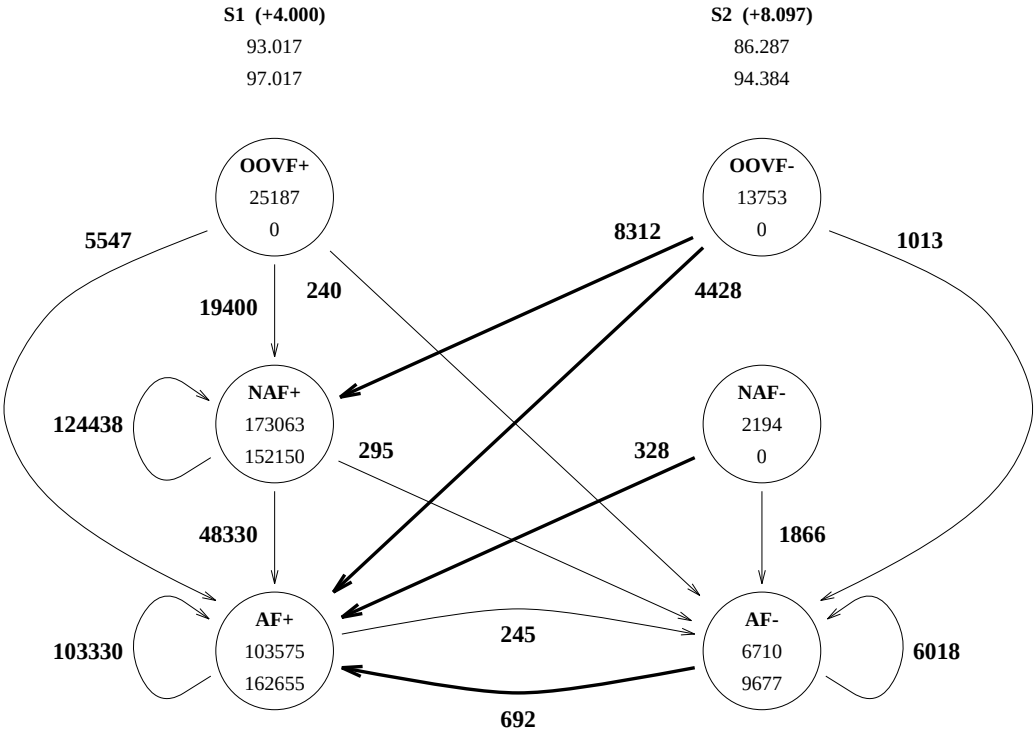


Figure 6.26: System: BRILL - Test: test31 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

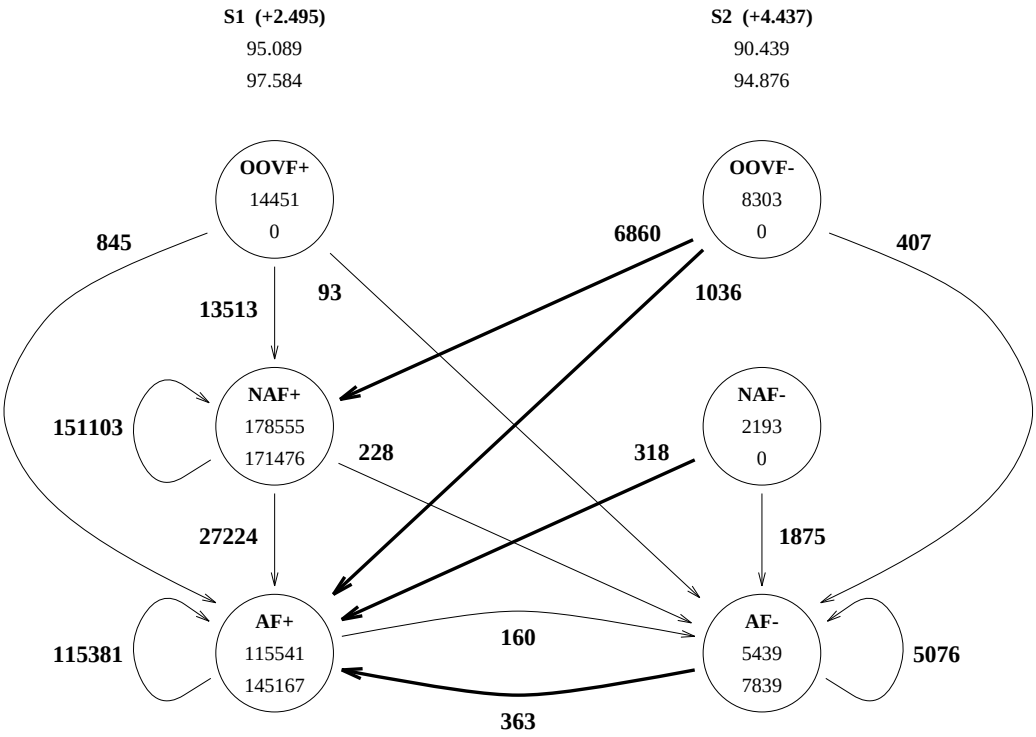


Figure 6.27: System: BRILL - Test: test32 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

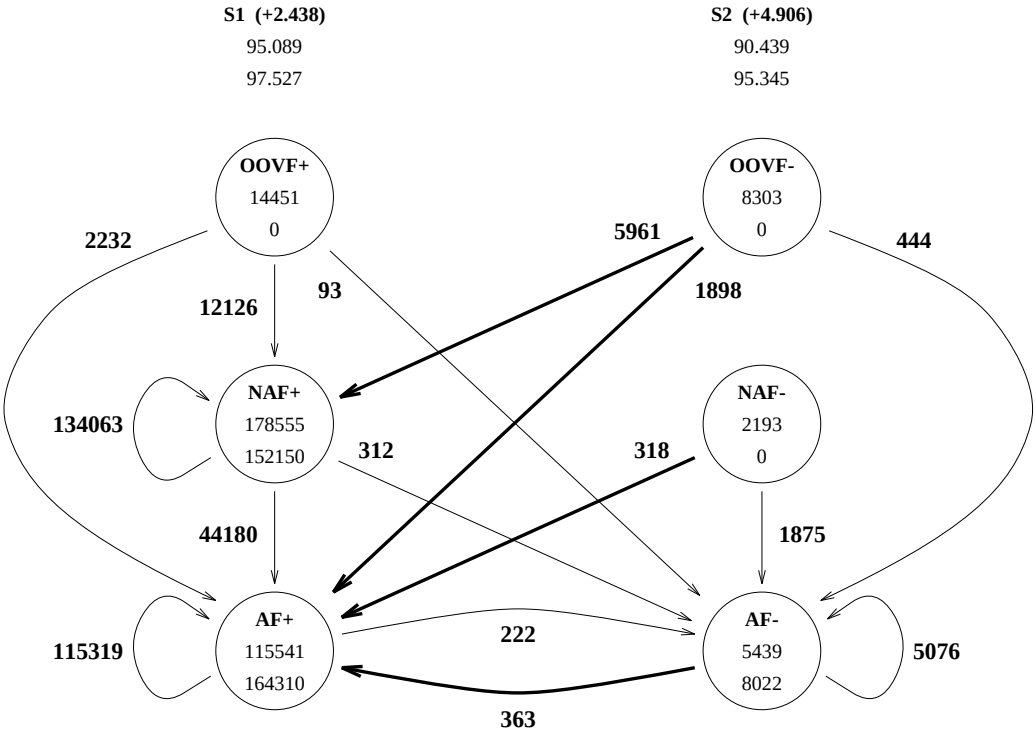


Figure 6.28: System: BRILL - Test: test32 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

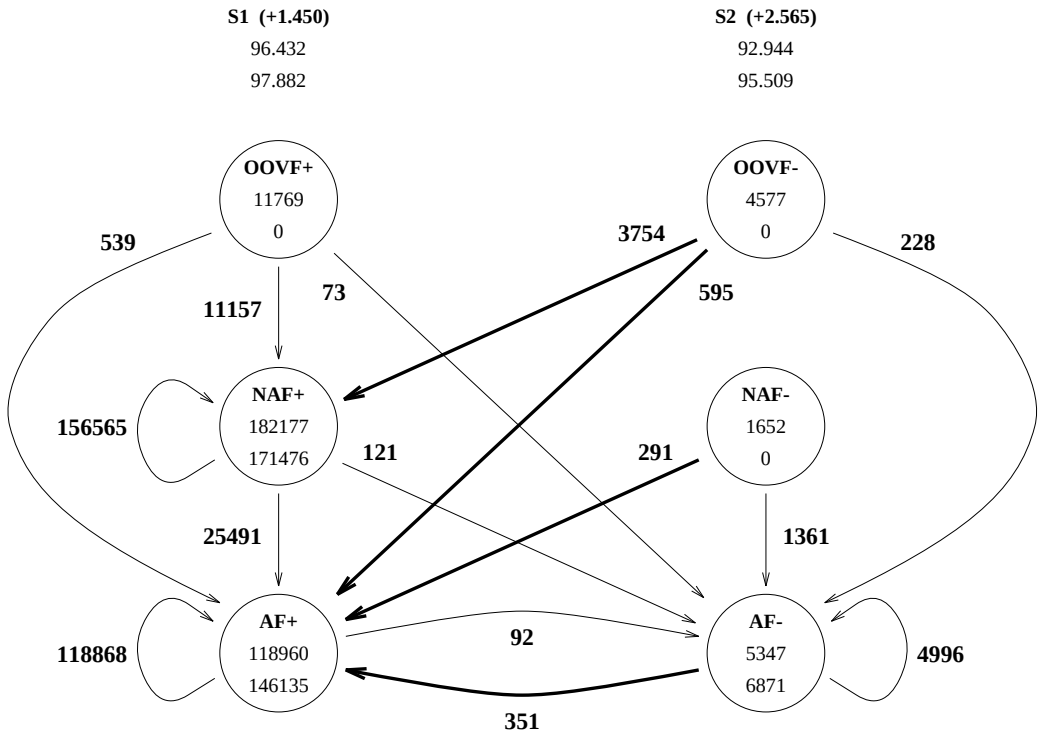


Figure 6.29: System: BRILL - Test: test33 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

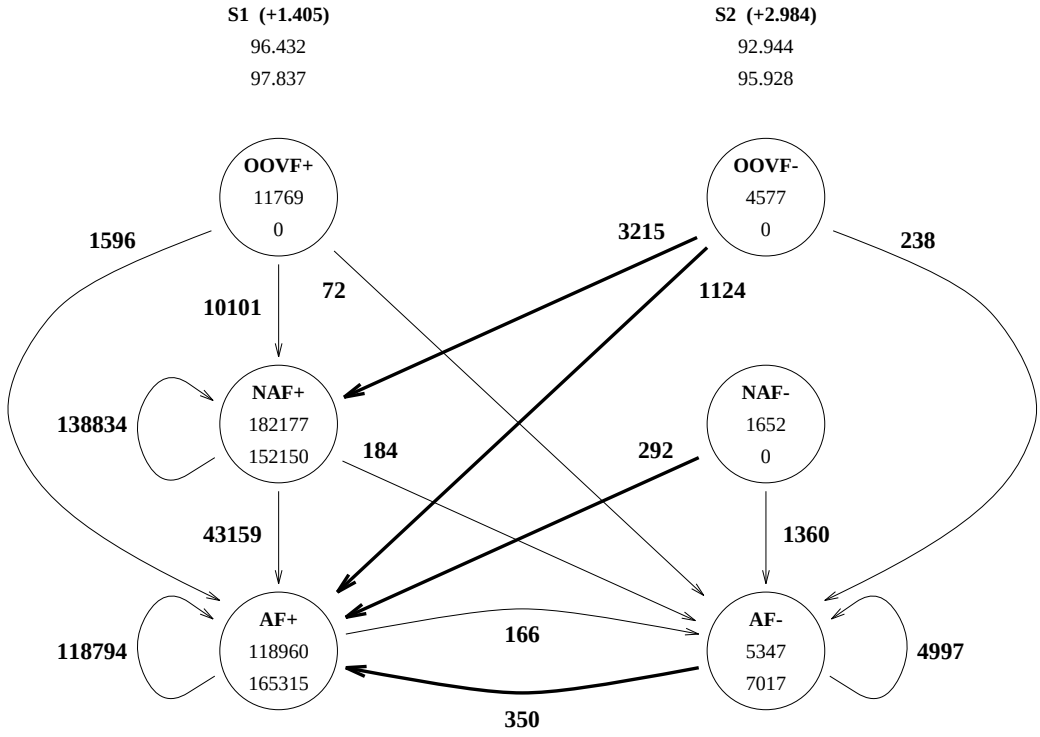


Figure 6.30: System: BRILL - Test: test33 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

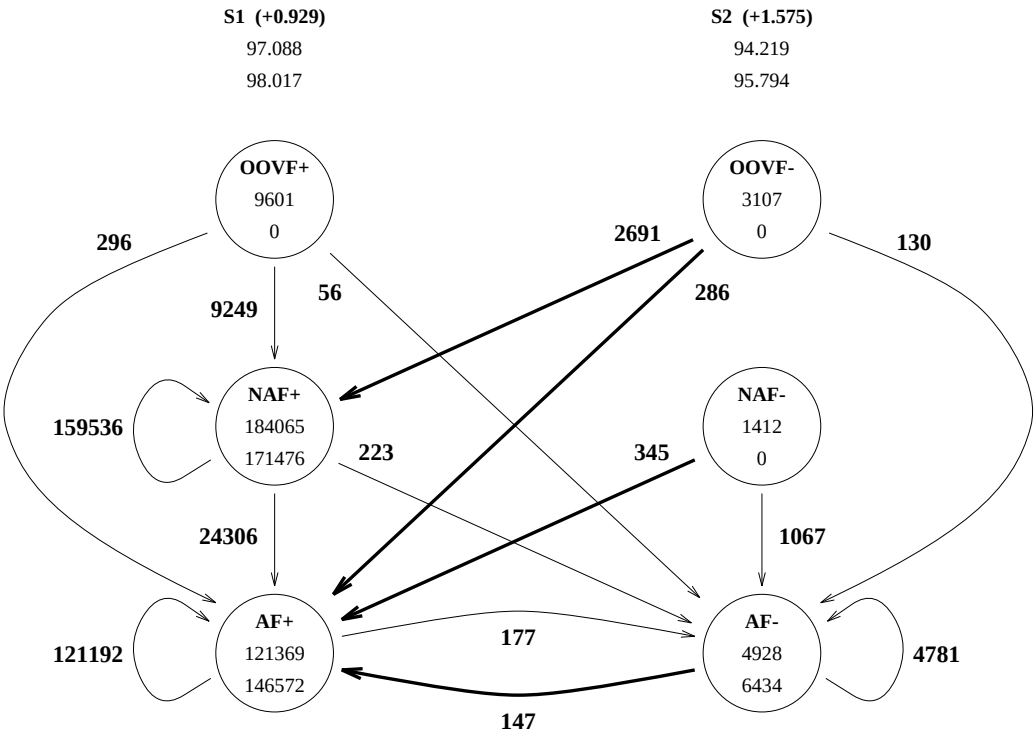


Figure 6.31: System: BRILL - Test: test34 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

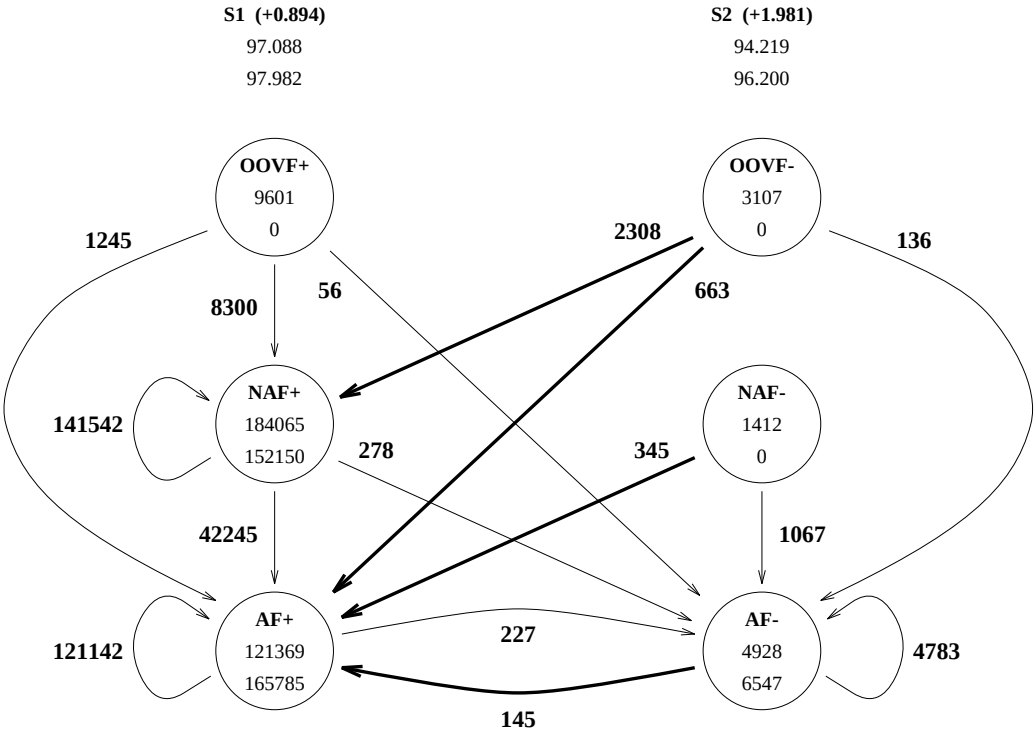


Figure 6.32: System: BRILL - Test: test34 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

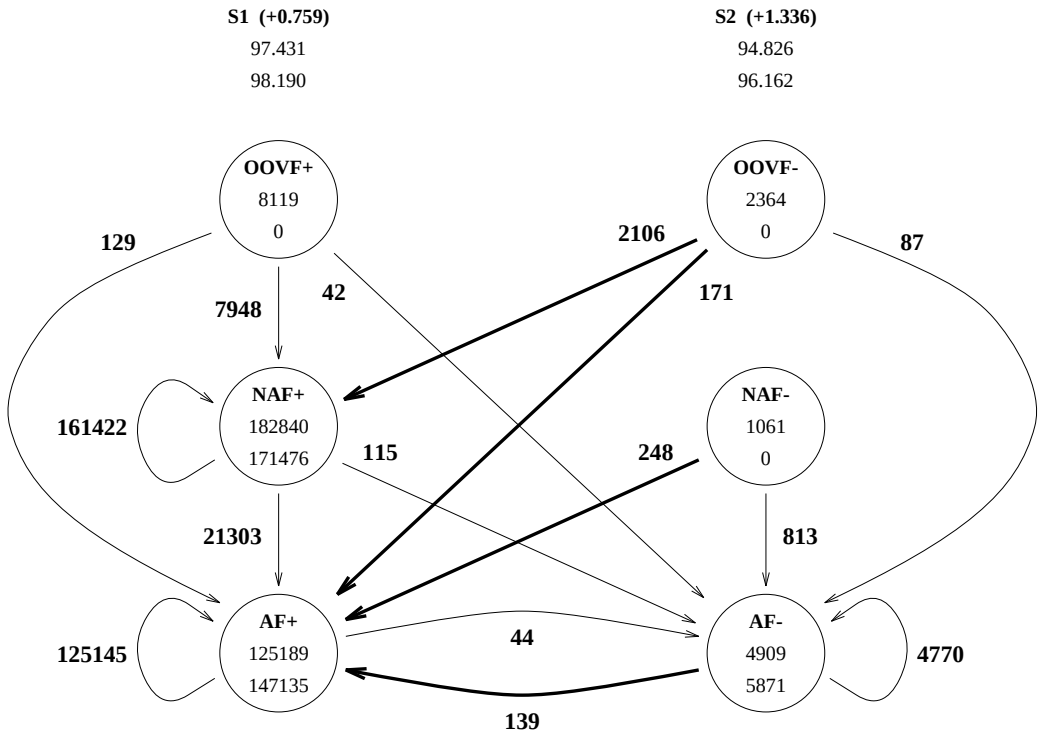


Figure 6.33: System: BRILL - Test: test35 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

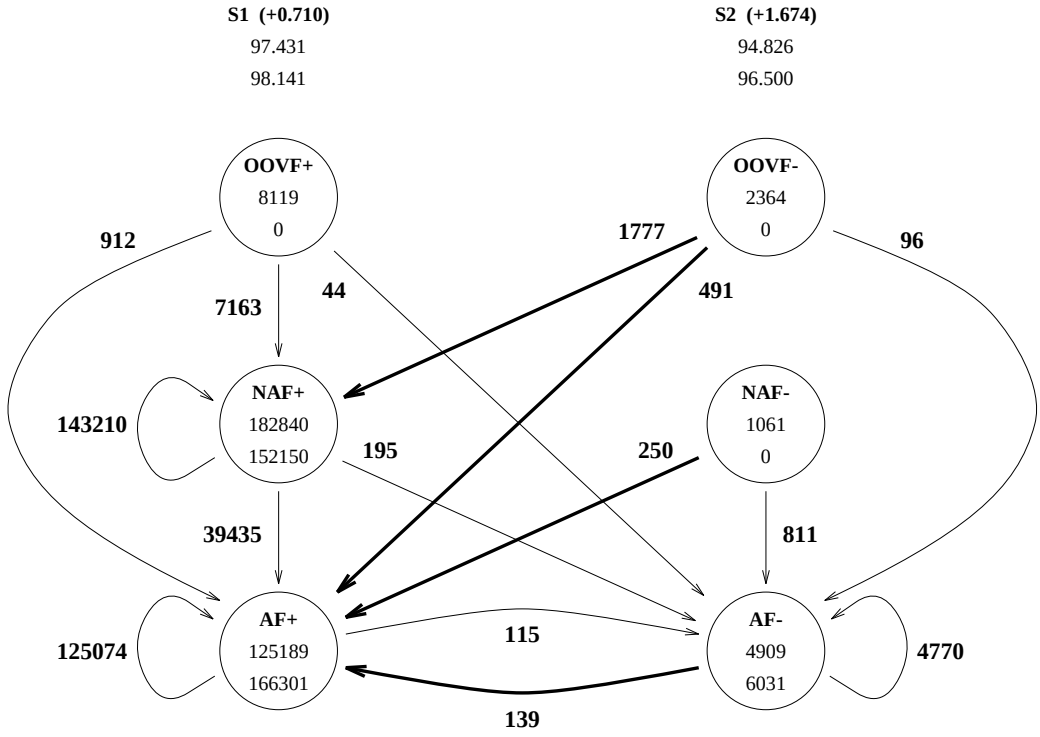


Figure 6.34: System: BRILL - Test: test35 - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

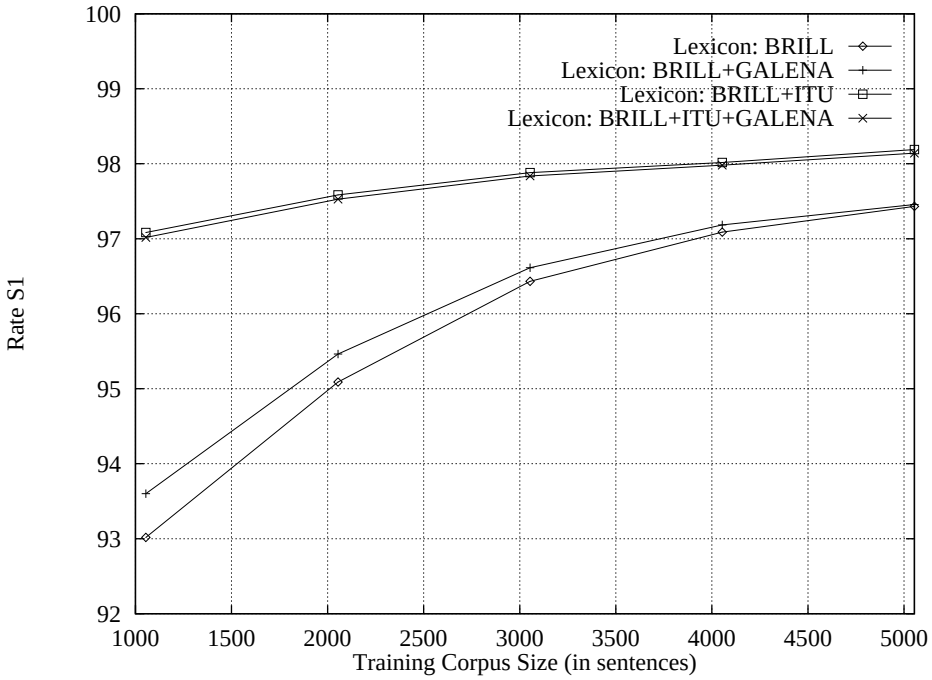


Figure 6.35: System: BRILL - Bank of Experiments: 3 - Rate: S1

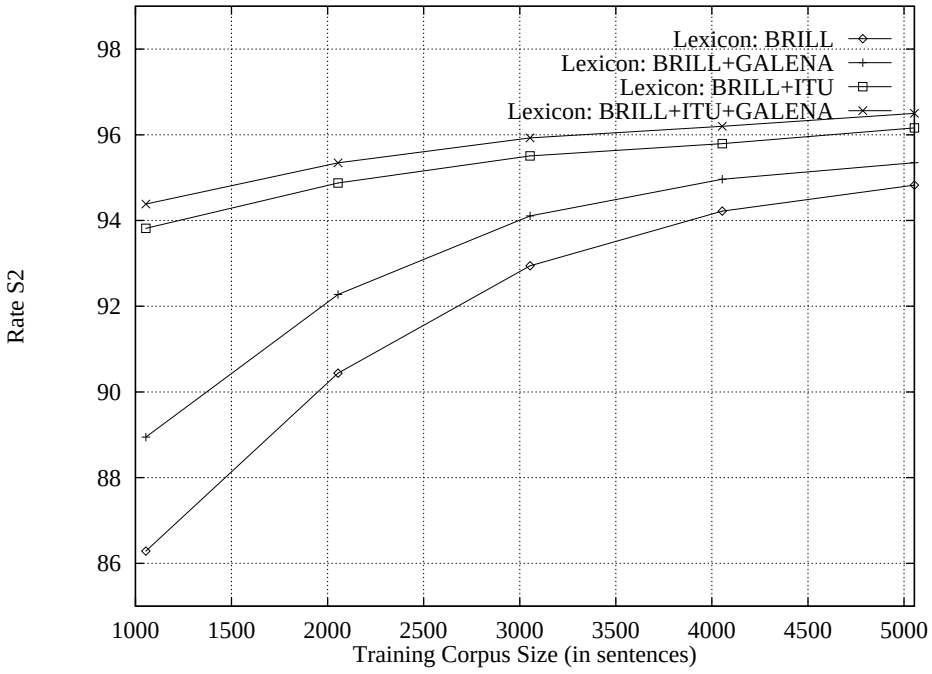


Figure 6.36: System: BRILL - Bank of Experiments: 3 - Rate: S2

System: BRILL		
Bank of Experiments: SPECIAL		
Experiment	testSP	
Host	liasun13	
Training Time	66:54:12	
Training Corpus Size		
Sentences	10054	
Words	324329	
Reference Corpus Size		
Sentences	4919	
Words	162972	
Lexicon: BRILL		
Tagging Time	00:01:46	
OOVF+	2719	80.02%
OOVF-	679	19.98%
NAF+	84705	99.62%
NAF-	327	0.38%
AF+	72268	96.95%
AF-	2274	3.05%
S1	97.987	
S2	96.211	
Lexicon: BRILL+GALENA		
Tagging Time	00:03:47	
OOVF+	1985	81.39%
OOVF-	454	18.61%
NAF+	75123	99.66%
NAF-	253	0.34%
AF+	82649	97.05%
AF-	2508	2.95%
S1	98.027	
S2	96.618	
Lexicon: BRILL+ITU		
Tagging Time	00:01:37	
OOVF+	0	-
OOVF-	0	-
NAF+	86291	100%
NAF-	0	0%
AF+	74180	96.74%
AF-	2501	3.26%
S1	98.465	
S2	96.738	
Lexicon: BRILL+ITU+GALENA		
Tagging Time	00:03:37	
OOVF+	0	-
OOVF-	0	-
NAF+	76523	100%
NAF-	0	0%
AF+	83896	97.05%
AF-	2553	2.95%
S1	98.433	
S2	97.046	

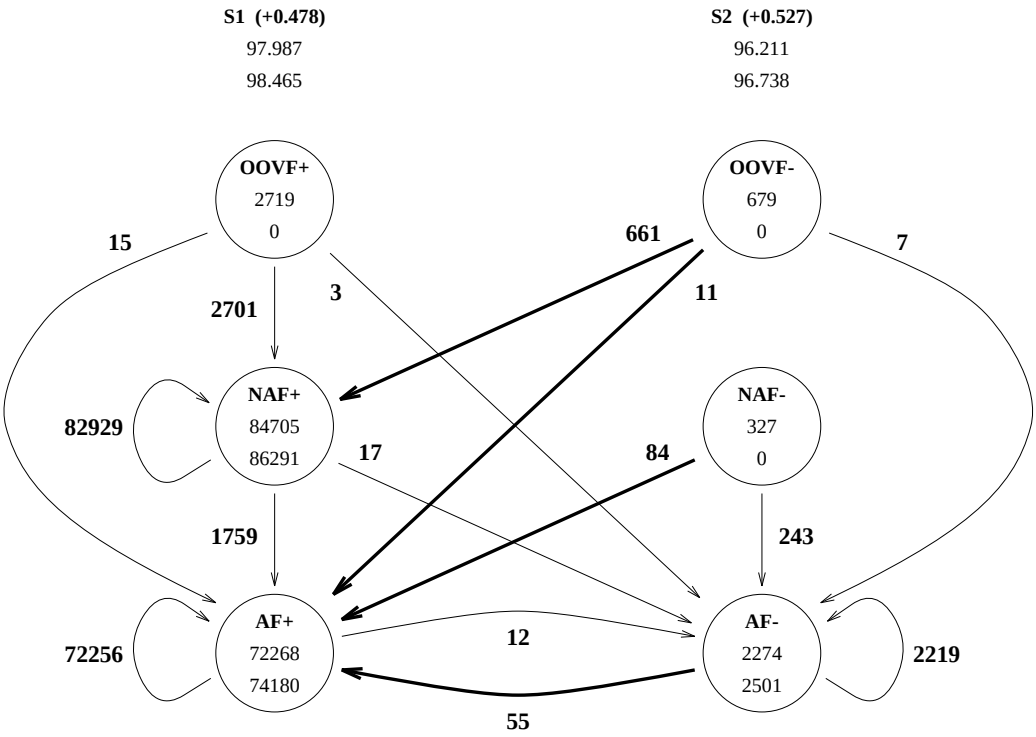


Figure 6.37: System: BRILL - Test: testSP - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU

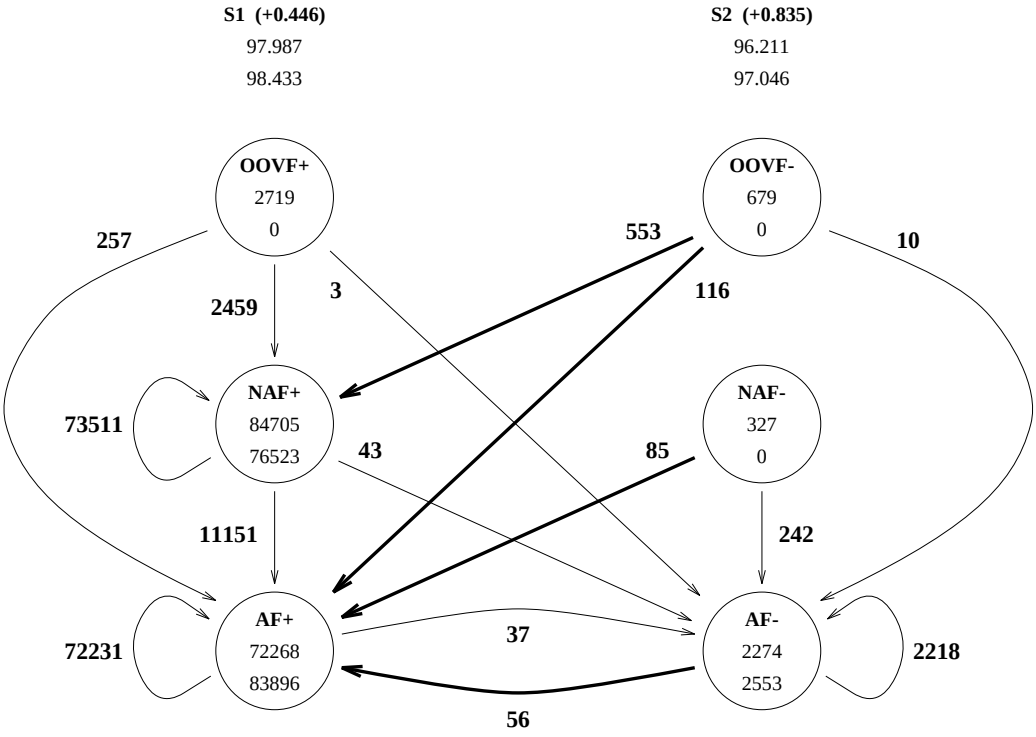


Figure 6.38: System: BRILL - Test: testSP - Word Transitions from Lexicon BRILL to Lexicon BRILL+ITU+GALENA

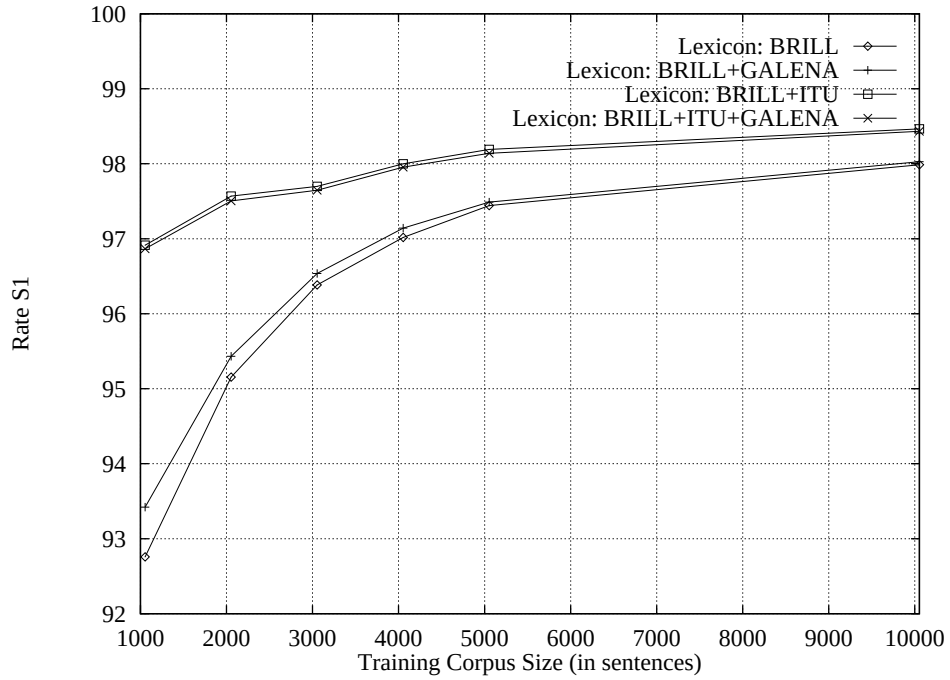


Figure 6.39: System: BRILL - Bank of Experiments: AVERAGE (Bank 1, Bank 2, Bank 3) and SPECIAL - Rate: S1

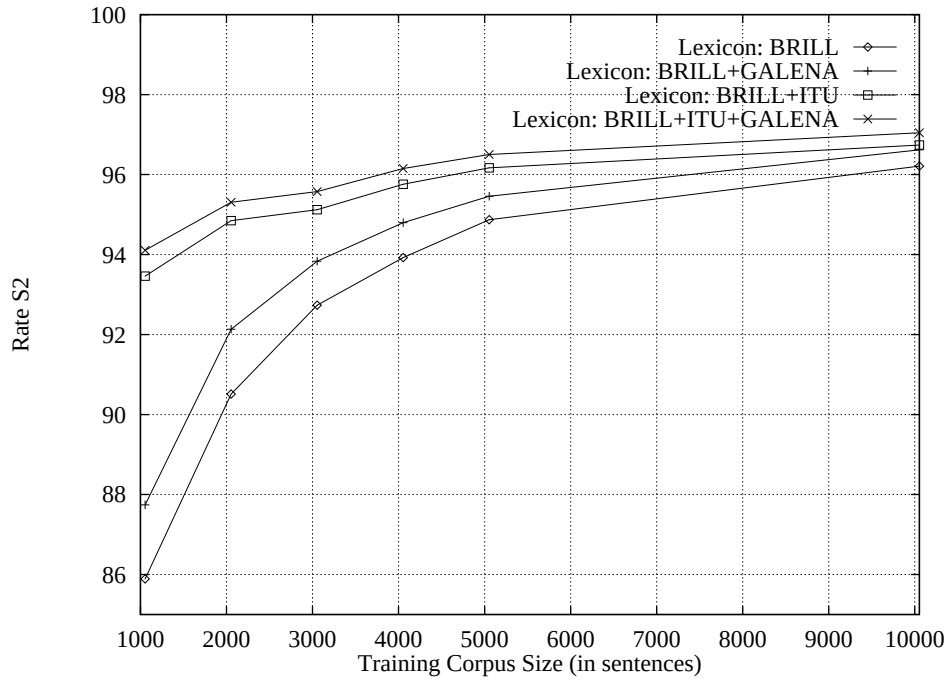


Figure 6.40: System: BRILL - Bank of Experiments: AVERAGE (Bank 1, Bank 2, Bank 3) and SPECIAL - Rate: S2

System: GALENA - Bank of Experiments: 1					
Experiment	test11	test12	test13	test14	test15
Host	covas	covas	covas	covas	covas
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	32839	68597	97194	126902	163358
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	326502	326502	326502	326502	326502
Configuration: 1					
Training Time	00:00:04	00:00:03	00:00:05	00:00:07	00:00:09
Tagging Time	00:02:03	00:02:08	00:02:08	00:02:10	00:02:40
OOVF+	37991	41387	43200	43857	43541
OOVF-	12856	9460	7647	6990	7306
NAF+	187710	187710	187710	187710	187710
NAF-	9665	9665	9665	9665	9665
AF+	55267	56174	56475	56460	56595
AF-	23013	22106	21805	21820	21685
S1	86.054	87.372	88.018	88.215	88.159
S2	72.222	75.553	77.189	77.688	77.547
Configuration: 2					
Training Time	00:00:03	00:00:04	00:00:05	00:00:08	00:00:08
Tagging Time	00:02:14	00:02:04	00:02:07	00:02:05	00:02:11
OOVF+	37991	41387	43200	43857	43541
OOVF-	12856	9460	7647	6990	7306
NAF+	187710	187710	187710	187710	187710
NAF-	9665	9665	9665	9665	9665
AF+	55525	56952	57442	57547	57867
AF-	22755	21328	20838	20733	20413
S1	86.133	87.610	88.314	88.548	88.549
S2	72.421	76.156	77.939	78.529	78.533
Configuration: 3					
Training Time	00:00:04	00:00:04	00:00:06	00:00:07	00:00:09
Tagging Time	00:02:17	00:02:07	00:02:05	00:02:19	00:02:07
OOVF+	37991	41387	43200	43857	43541
OOVF-	12856	9460	7647	6990	7306
NAF+	187710	187710	187710	187710	187710
NAF-	9665	9665	9665	9665	9665
AF+	55433	56800	57314	57475	57638
AF-	22847	21480	20966	20805	20642
S1	86.105	87.563	88.275	88.526	88.479
S2	72.350	76.038	77.839	78.473	78.354

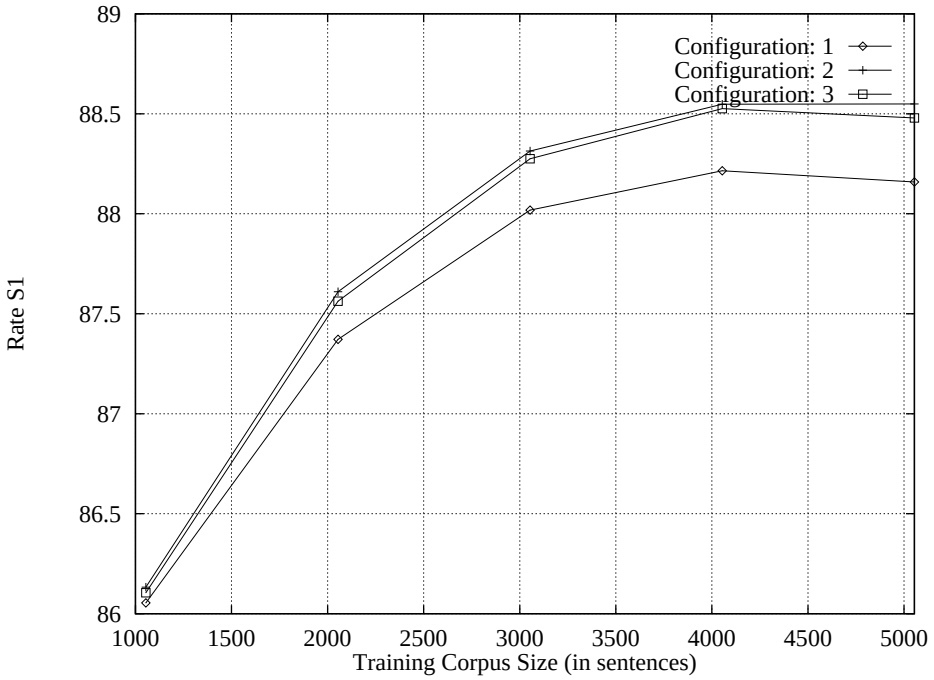


Figure 6.41: System: GALENA - Bank of Experiments: 1 - Rate: S1

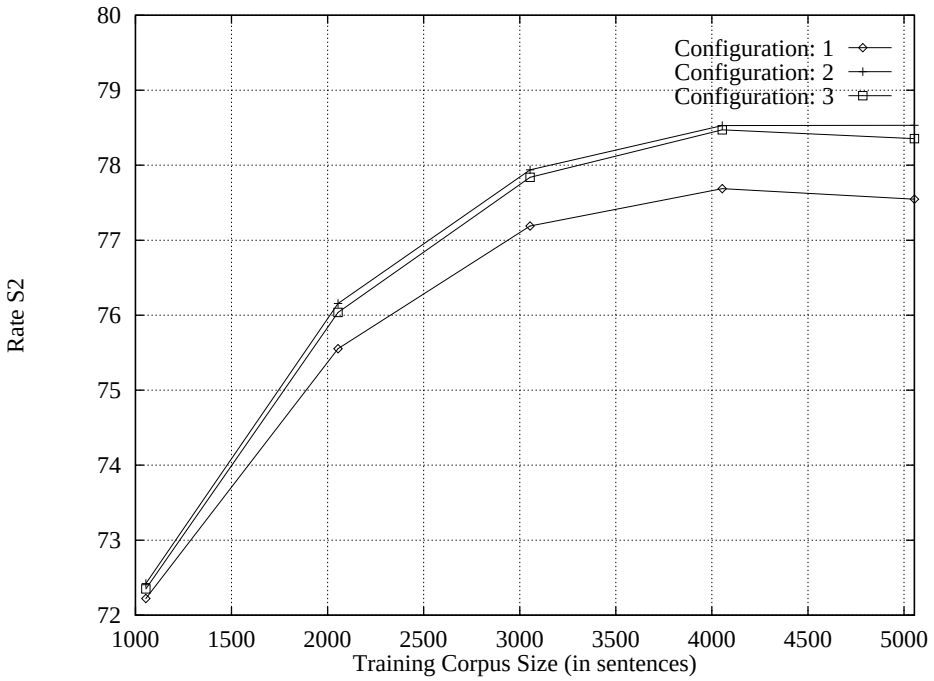


Figure 6.42: System: GALENA - Bank of Experiments: 1 - Rate: S2

System: GALENA - Bank of Experiments: 2					
Experiment	test21	test22	test23	test24	test25
Host	covas	covas	covas	covas	covas
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	33244	71640	1 00637	130377	166920
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	322940	322940	322940	322940	322940
Configuration: 1					
Training Time	00:00:02	00:00:04	00:00:06	00:00:07	00:00:08
Tagging Time	00:02:06	00:02:04	00:02:06	00:02:04	00:02:06
OOVF+	37467	40860	42174	43198	44276
OOVF-	12750	9357	8043	7019	5941
NAF+	185946	185946	185946	185946	185946
NAF-	9634	9634	9634	9634	9634
AF+	54599	55582	55780	55997	56144
AF-	22544	21561	21363	21146	20999
S1	86.987	87.443	87.911	88.295	88.673
S2	72.288	75.723	76.911	77.885	78.845
Configuration: 2					
Training Time	00:00:02	00:00:04	00:00:05	00:00:07	00:00:08
Tagging Time	00:02:12	00:02:18	00:02:20	00:02:06	00:02:06
OOVF+	37467	40860	42174	43198	44276
OOVF-	12750	9357	8043	7019	5941
NAF+	185946	185946	185946	185946	185946
NAF-	9634	9634	9634	9634	9634
AF+	54856	56286	56682	57080	57196
AF-	22287	20857	20461	20063	19947
S1	86.167	87.661	88.190	88.630	88.999
S2	72.489	76.277	77.619	78.736	79.671
Configuration: 3					
Training Time	00:00:02	00:00:04	00:00:06	00:00:07	00:00:09
Tagging Time	00:02:20	00:02:07	00:02:05	00:02:04	00:02:14
OOVF+	37467	40860	42174	43198	44276
OOVF-	12750	9357	8043	7019	5941
NAF+	185946	185946	185946	185946	185946
NAF-	9634	9634	9634	9634	9634
AF+	54791	56082	56416	56873	56987
AF-	22352	21061	20727	20270	20156
S1	86.146	87.597	88.107	88.566	88.934
S2	72.438	76.116	77.410	78.574	79.507

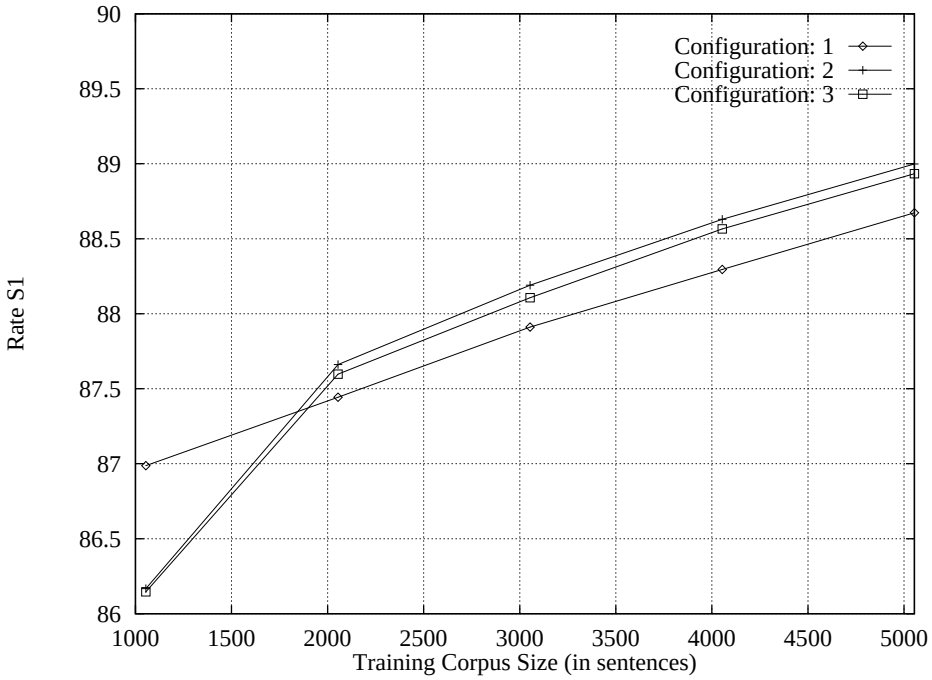


Figure 6.43: System: GALENA - Bank of Experiments: 2 - Rate: S1

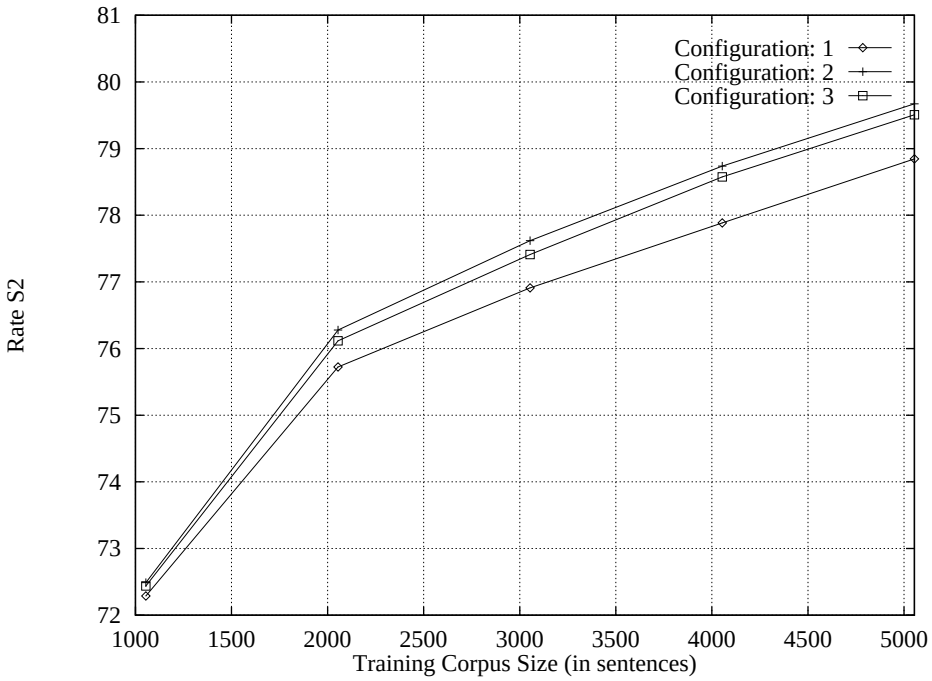


Figure 6.44: System: GALENA - Bank of Experiments: 2 - Rate: S2

System: GALENA - Bank of Experiments: 3					
Experiment	test31	test32	test33	test34	test35
Host	covas	covas	covas	covas	covas
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	33371	70040	97979	127835	163646
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	326214	326214	326214	326214	326214
Configuration: 1					
Training Time	00:00:02	00:00:04	00:00:05	00:00:06	00:00:08
Tagging Time	00:02:10	00:02:08	00:02:18	00:02:11	00:02:05
OOVF+	38194	41431	43198	43590	43996
OOVF-	12661	9424	7657	7265	6859
NAF+	188042	188042	188042	188042	188042
NAF-	9736	9736	9736	9736	9736
AF+	55322	56352	56563	56550	56875
AF-	22259	21229	21018	21031	20706
S1	86.311	87.618	88.224	88.341	88.565
S2	72.811	76.133	77.673	77.969	78.537
Configuration: 2					
Training Time	00:00:02	00:00:04	00:00:05	00:00:06	00:00:08
Tagging Time	00:02:19	00:02:04	00:02:05	00:02:12	00:02:07
OOVF+	38194	41431	43198	43590	43996
OOVF-	12661	9424	7657	7265	6859
NAF+	188042	188042	188042	188042	188042
NAF-	9736	9736	9736	9736	9736
AF+	55415	56894	57238	56550	57704
AF-	22166	20687	20343	21031	19877
S1	86.339	87.784	88.431	88.341	88.819
S2	72.884	76.555	78.199	77.969	79.183
Configuration: 3					
Training Time	00:00:02	00:00:04	00:00:05	00:00:08	00:00:09
Tagging Time	00:02:04	00:02:04	00:02:04	00:02:07	00:02:04
OOVF+	38194	41431	43198	43590	43996
OOVF-	12661	9424	7657	7265	6859
NAF+	188042	188042	188042	188042	188042
NAF-	9736	9736	9736	9736	9736
AF+	55209	56720	57134	57435	57541
AF-	22372	20861	20447	20146	20040
S1	86.276	87.731	88.399	88.612	88.769
S2	72.723	76.419	78.117	78.657	79.055

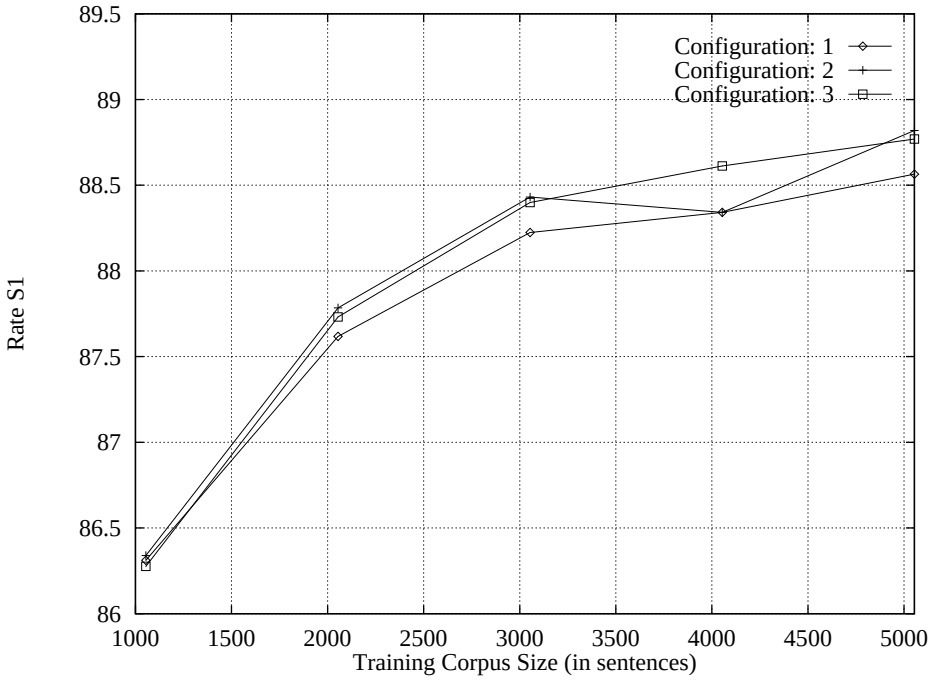


Figure 6.45: System: GALENA - Bank of Experiments: 3 - Rate: S1

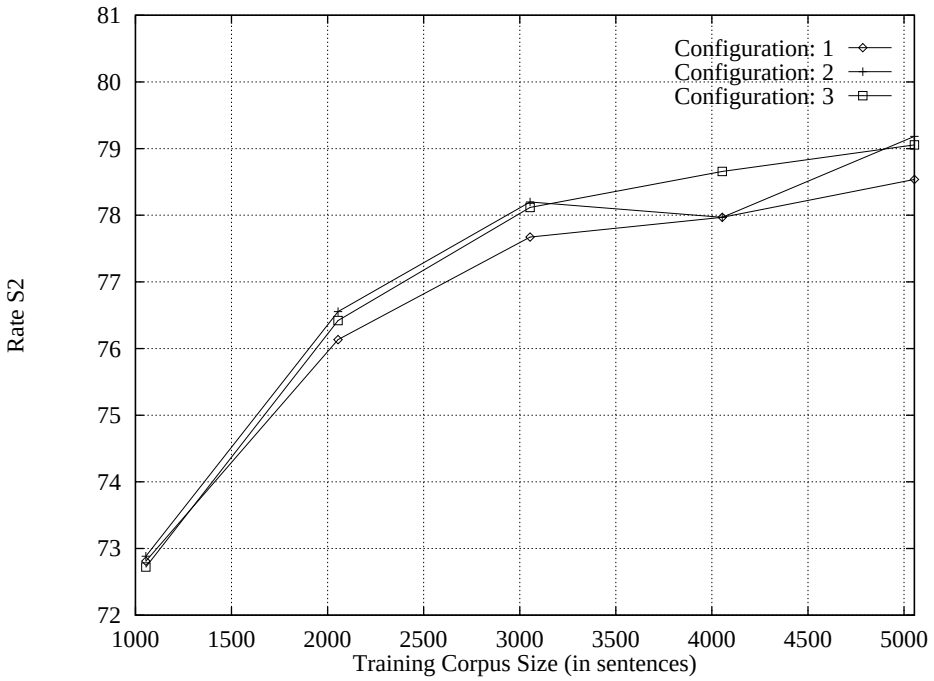


Figure 6.46: System: GALENA - Bank of Experiments: 3 - Rate: S2

System: GALENA Bank of Experiments: SPECIAL	
Experiment	testSP
Host	covas
Training Corpus Size	
Sentences	10054
Words	325998
Reference Corpus Size	
Sentences	4919
Words	163862
Configuration: 1	
Training Time	00:00:16
Tagging Time	00:01:03
OOVF+	22901
OOVF-	2766
NAF+	94342
NAF-	4881
AF+	28585
AF-	10387
S1	88.994
S2	79.651
Configuration: 2	
Training Time	00:00:16
Tagging Time	00:01:04
OOVF+	22901
OOVF-	2766
NAF+	94342
NAF-	4881
AF+	28585
AF-	9761
S1	89.376
S2	80.620
Configuration: 3	
Training Time	00:00:17
Tagging Time	00:01:03
OOVF+	22901
OOVF-	2766
NAF+	94342
NAF-	4881
AF+	29140
AF-	9832
S1	89.333
S2	80.510

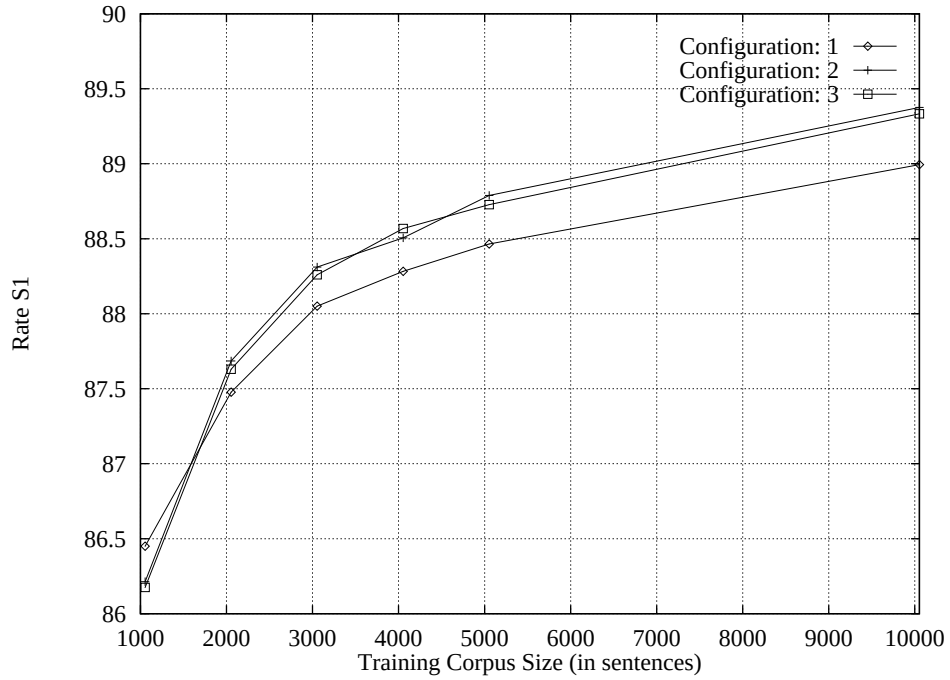


Figure 6.47: System: GALENA - Bank of Experiments: AVERAGE (Bank 1, Bank 2, Bank 3) and SPECIAL - Rate: S1

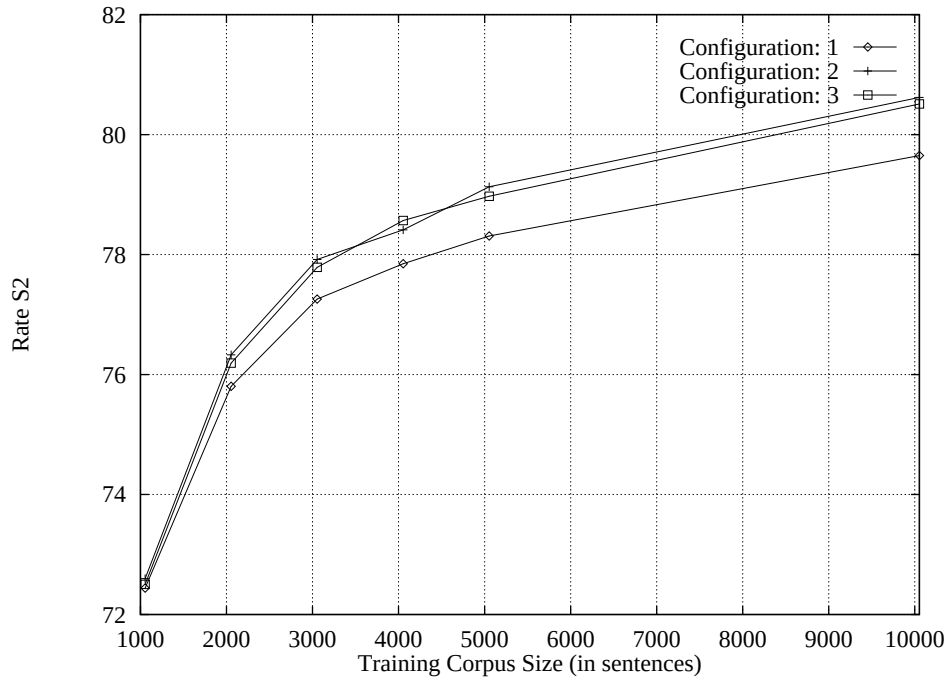


Figure 6.48: System: GALENA - Bank of Experiments: AVERAGE (Bank 1, Bank 2, Bank 3) and SPECIAL - Rate: S2

System: GALENA (3 BEST TAGS) - Bank of Experiments: 1					
Experiment	test11	test12	test13	test14	test15
Host	covas	covas	covas	covas	covas
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	32839	68597	97194	126902	163358
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	326502	326502	326502	326502	326502
Configuration: 1					
Training Time	00:00:04	00:00:03	00:00:05	00:00:07	00:00:09
Tagging Time	00:02:03	00:02:08	00:02:08	00:02:10	00:02:40
OOVF+	37991	41387	43200	43857	43541
OOVF-	12856	9460	7647	6990	7306
NAF+	187710	187710	187710	187710	187710
NAF-	9665	9665	9665	9665	9665
AFAT+	71442	71592	71623	71634	71652
AFAT-	6838	6688	6657	6646	6628
S1'	91.008	92.094	92.658	92.863	92.771
S2'	84.748	87.493	88.921	89.493	89.207
Configuration: 2					
Training Time	00:00:03	00:00:04	00:00:05	00:00:08	00:00:08
Tagging Time	00:02:14	00:02:04	00:02:07	00:02:05	00:02:11
OOVF+	37991	41387	43200	43857	43541
OOVF-	12856	9460	7647	6990	7306
NAF+	187710	187710	187710	187710	187710
NAF-	9665	9665	9665	9665	9665
AFAT+	71385	71548	71590	71608	71633
AFAT-	6895	6732	6690	6672	6647
S1'	90.990	92.080	92.647	92.854	92.765
S2'	84.704	87.460	88.896	89.418	89.193
Configuration: 3					
Training Time	00:00:04	00:00:04	00:00:06	00:00:07	00:00:09
Tagging Time	00:02:17	00:02:07	00:02:05	00:02:19	00:02:07
OOVF+	37991	41387	43200	43857	43541
OOVF-	12856	9460	7647	6990	7306
NAF+	187710	187710	187710	187710	187710
NAF-	9665	9665	9665	9665	9665
AFAT+	71340	71509	71548	71574	71604
AFAT-	6940	6771	6732	6706	6676
S1'	90.977	92.068	92.635	92.844	92.756
S2'	84.668	87.429	88.862	89.391	89.169

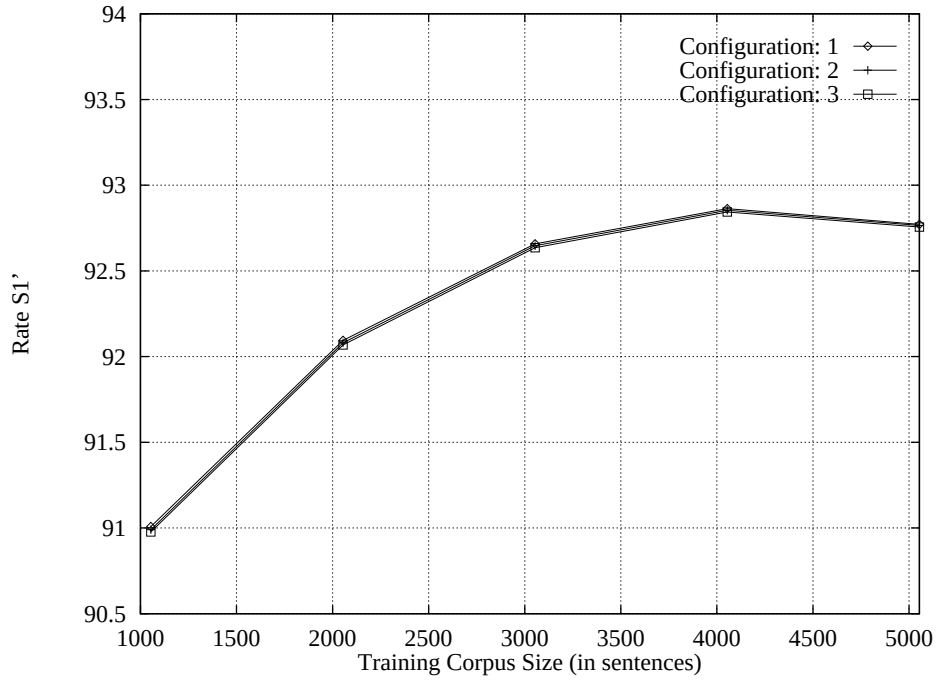


Figure 6.49: System: GALENA (3 BEST TAGS) - Bank of Experiments: 1 - Rate: S1'

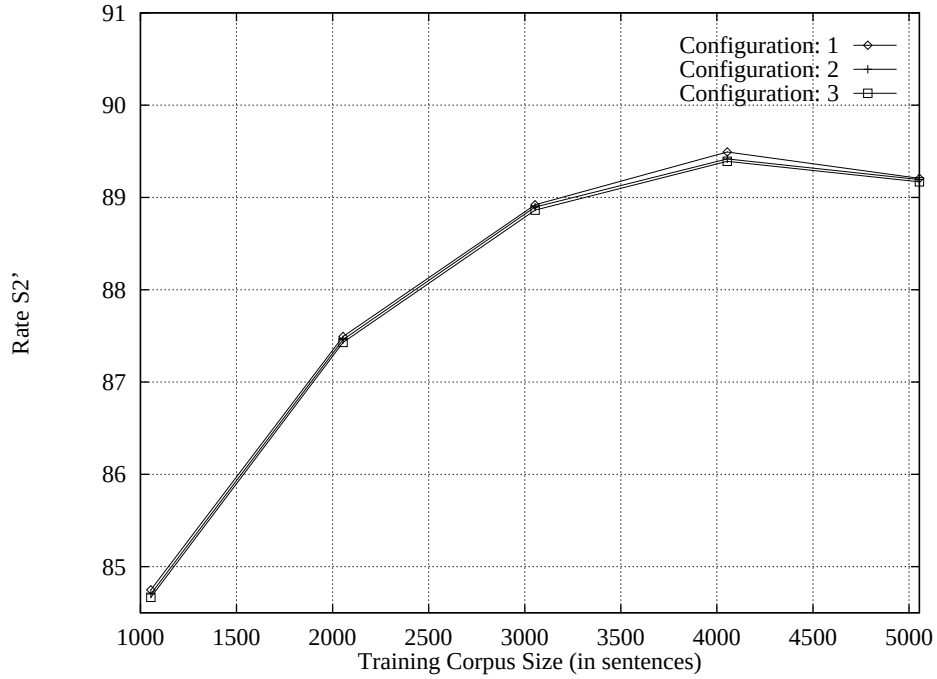


Figure 6.50: System: GALENA (3 BEST TAGS) - Bank of Experiments: 1 - Rate: S2'

System: GALENA (3 BEST TAGS) - Bank of Experiments: 2					
Experiment	test21	test22	test23	test24	test25
Host	covas	covas	covas	covas	covas
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	33244	71640	1 00637	130377	166920
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	322940	322940	322940	322940	322940
Configuration: 1					
Training Time	00:00:02	00:00:04	00:00:06	00:00:07	00:00:08
Tagging Time	00:02:06	00:02:04	00:02:06	00:02:04	00:02:06
OOVF+	37467	40860	42174	43198	44276
OOVF-	12750	9357	8043	7019	5941
NAF+	185946	185946	185946	185946	185946
NAF-	9634	9634	9634	9634	9634
AFAT+	70415	70556	70595	70609	70629
AFAT-	6728	6587	6548	6534	6514
S1'	90.985	92.079	92.498	92.819	93.159
S2'	84.706	87.481	88.544	89.358	90.218
Configuration: 2					
Training Time	00:00:02	00:00:04	00:00:05	00:00:07	00:00:08
Tagging Time	00:02:12	00:02:18	00:02:20	00:02:06	00:02:06
OOVF+	37467	40860	42174	43198	44276
OOVF-	12750	9357	8043	7019	5941
NAF+	185946	185946	185946	185946	185946
NAF-	9634	9634	9634	9634	9634
AFAT+	70370	70544	70582	70598	70614
AFAT-	6773	6599	6561	6545	6529
S1'	90.971	92.076	92.494	92.816	93.154
S2'	84.671	87.472	88.533	89.350	90.206
Configuration: 3					
Training Time	00:00:02	00:00:04	00:00:06	00:00:07	00:00:09
Tagging Time	00:02:20	00:02:07	00:02:05	00:02:04	00:02:14
OOVF+	37467	40860	42174	43198	44276
OOVF-	12750	9357	8043	7019	5941
NAF+	185946	185946	185946	185946	185946
NAF-	9634	9634	9634	9634	9634
AFAT+	70317	70508	70547	70569	70594
AFAT-	6826	6635	6596	6574	6549
S1'	90.954	92.064	92.483	92.807	93.147
S2'	84.629	87.443	88.506	89.328	90.191

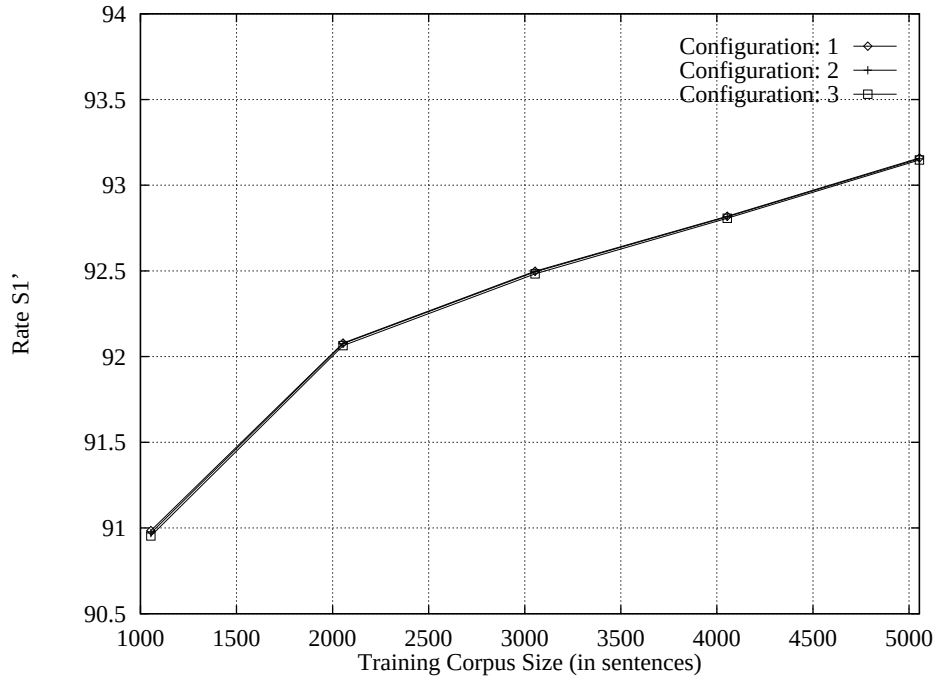


Figure 6.51: System: GALENA (3 BEST TAGS) - Bank of Experiments: 2 - Rate: S1'

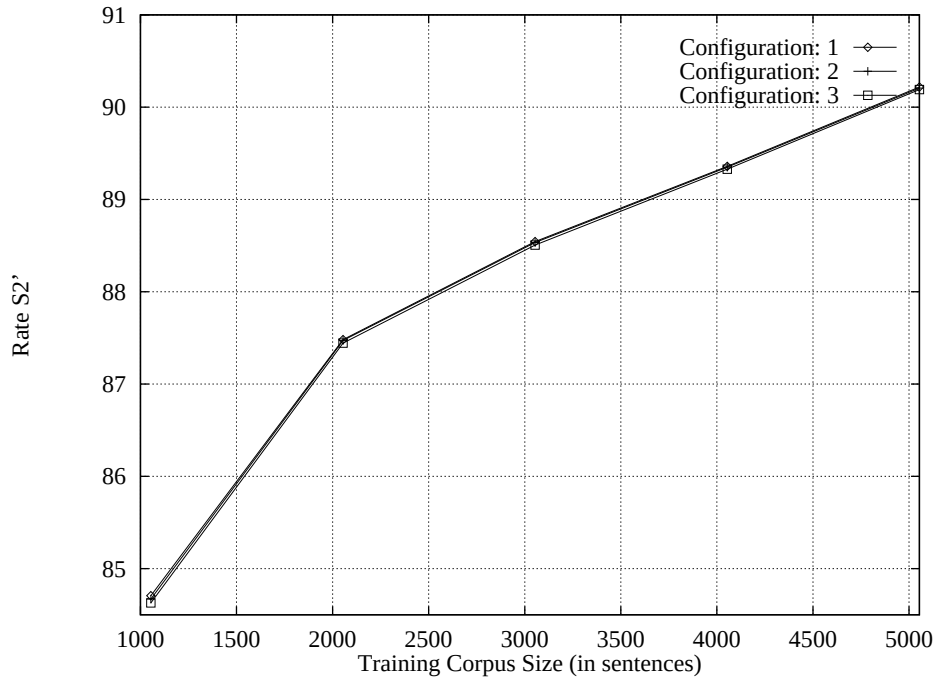


Figure 6.52: System: GALENA (3 BEST TAGS) - Bank of Experiments: 2 - Rate: S2'

System: GALENA (3 BEST TAGS) - Bank of Experiments: 3					
Experiment	test31	test32	test33	test34	test35
Host	covas	covas	covas	covas	covas
Training Corpus Size					
Sentences	1054	2054	3054	4054	5054
Words	33371	70040	97979	127835	163646
Reference Corpus Size					
Sentences	9919	9919	9919	9919	9919
Words	326214	326214	326214	326214	326214
Configuration: 1					
Training Time	00:00:02	00:00:04	00:00:05	00:00:06	00:00:08
Tagging Time	00:02:10	00:02:08	00:02:18	00:02:11	00:02:05
OOVF+	38194	41431	43198	43590	43996
OOVF-	12661	9424	7657	7265	6859
NAF+	188042	188042	188042	188042	188042
NAF-	9736	9736	9736	9736	9736
AFAT+	70940	71033	71056	71052	71073
AFAT-	6641	6548	6525	6529	6508
S1'	91.098	92.118	92.667	92.786	92.917
S2'	84.971	87.564	88.958	89.260	89.592
Configuration: 2					
Training Time	00:00:02	00:00:04	00:00:05	00:00:06	00:00:08
Tagging Time	00:02:19	00:02:04	00:02:05	00:02:12	00:02:07
OOVF+	38194	41431	43198	43590	43996
OOVF-	12661	9424	7657	7265	6859
NAF+	188042	188042	188042	188042	188042
NAF-	9736	9736	9736	9736	9736
AFAT+	70866	70982	71013	71033	71062
AFAT-	6715	6599	6568	6548	6519
S1'	91.076	92.103	92.654	92.780	92.914
S2'	84.914	87.524	88.925	89.245	89.583
Configuration: 3					
Training Time	00:00:02	00:00:04	00:00:05	00:00:08	00:00:09
Tagging Time	00:02:04	00:02:04	00:02:04	00:02:07	00:02:04
OOVF+	38194	41431	43198	43590	43996
OOVF-	12661	9424	7657	7265	6859
NAF+	188042	188042	188042	188042	188042
NAF-	9736	9736	9736	9736	9736
AFAT+	70808	70932	70969	70999	71034
AFAT-	6773	6649	6612	6582	6547
S1'	91.058	92.087	92.640	92.770	92.905
S2'	84.868	87.485	88.889	89.218	89.561

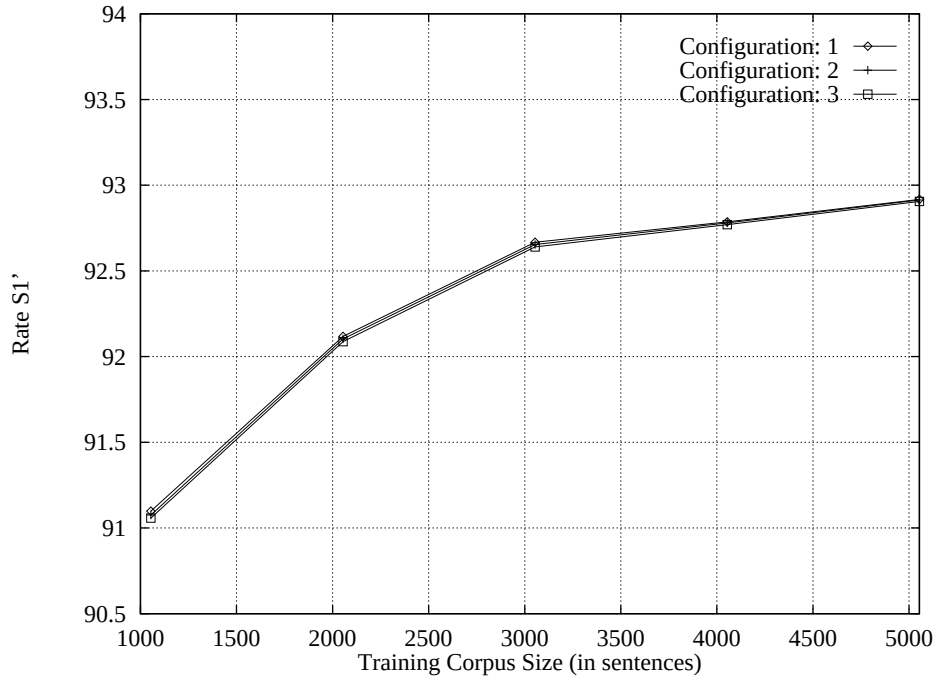


Figure 6.53: System: GALENA (3 BEST TAGS) - Bank of Experiments: 3 - Rate: S1'

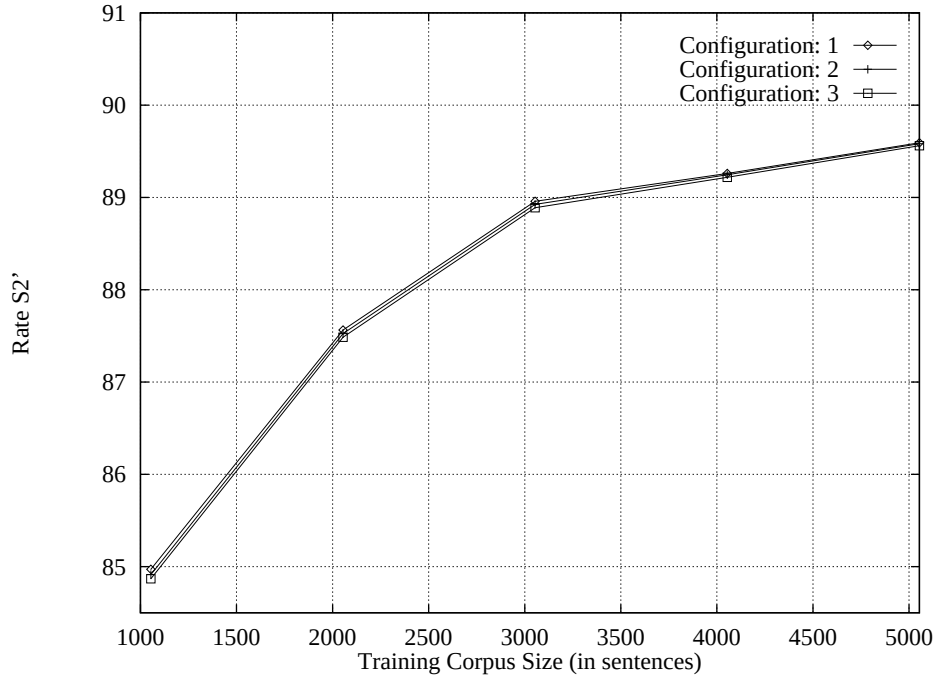


Figure 6.54: System: GALENA (3 BEST TAGS) - Bank of Experiments: 3 - Rate: S2'

System: GALENA (3 BEST TAGS) Bank of Experiments: SPECIAL	
Experiment	testSP
Host	covas
Training Corpus Size	
Sentences	10054
Words	325998
Reference Corpus Size	
Sentences	4919
Words	163862
Configuration: 1	
Training Time	00:00:16
Tagging Time	00:01:03
OOVF+	22901
OOVF-	2766
NAF+	94342
NAF-	4881
AFAT+	35712
AFAT-	3260
S1'	93.343
S2'	90.677
Configuration: 2	
Training Time	00:00:16
Tagging Time	00:01:04
OOVF+	22901
OOVF-	2766
NAF+	94342
NAF-	4881
AFAT+	35719
AFAT-	3253
S1'	93.348
S2'	90.688
Configuration: 3	
Training Time	00:00:17
Tagging Time	00:01:03
OOVF+	22901
OOVF-	2766
NAF+	94342
NAF-	4881
AFAT+	35697
AFAT-	3275
S1'	93.334
S2'	90.654

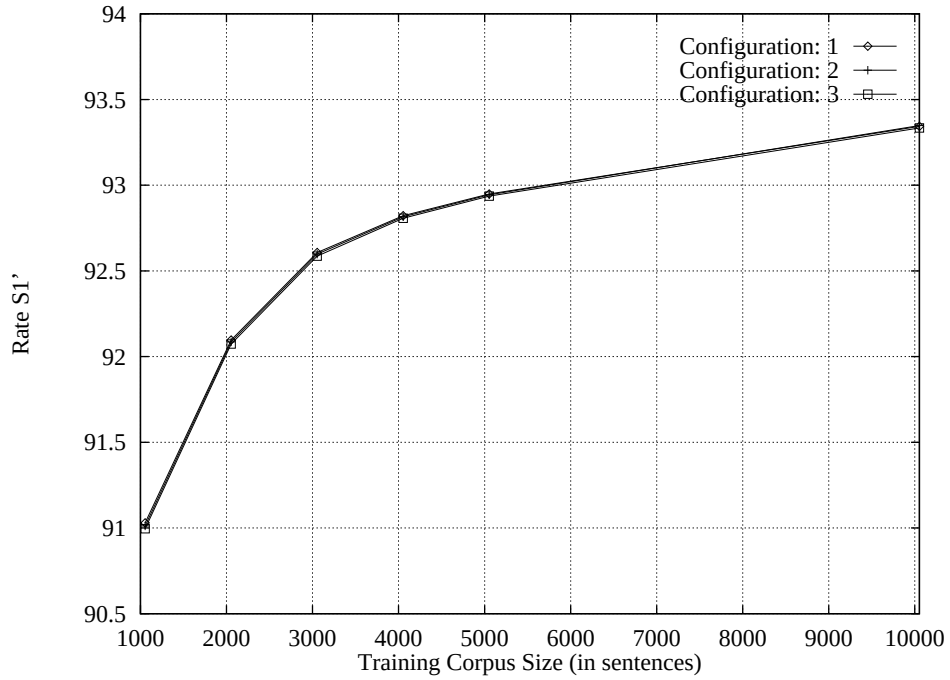


Figure 6.55: System: GALENA (3 BEST TAGS) - Bank of Experiments: AVERAGE (Bank 1, Bank 2, Bank 3) and SPECIAL - Rate: S1'

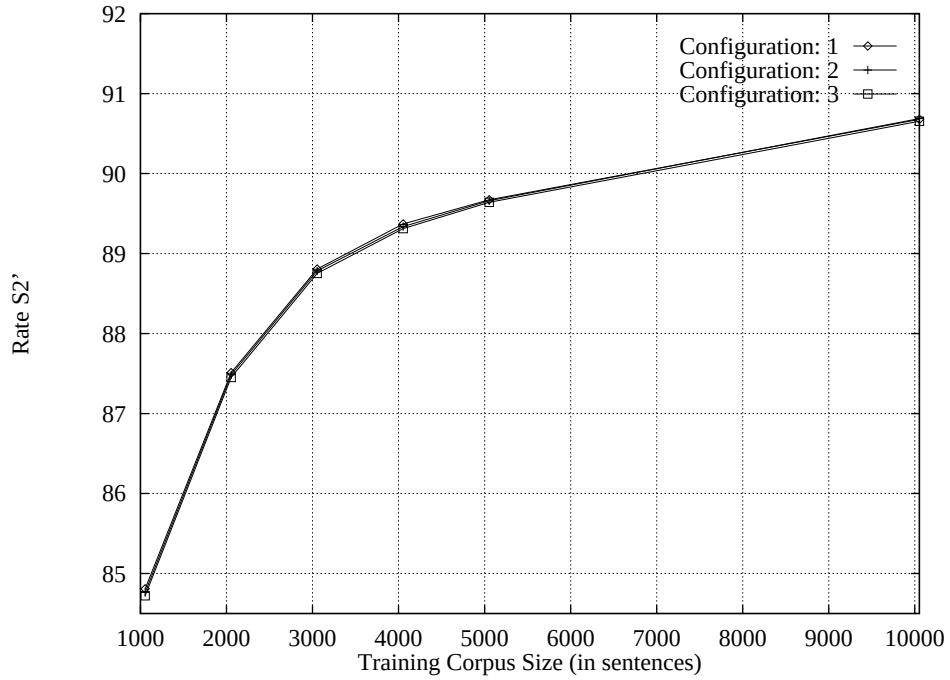


Figure 6.56: System: GALENA (3 BEST TAGS) - Bank of Experiments: AVERAGE (Bank 1, Bank 2, Bank 3) and SPECIAL - Rate: S2'

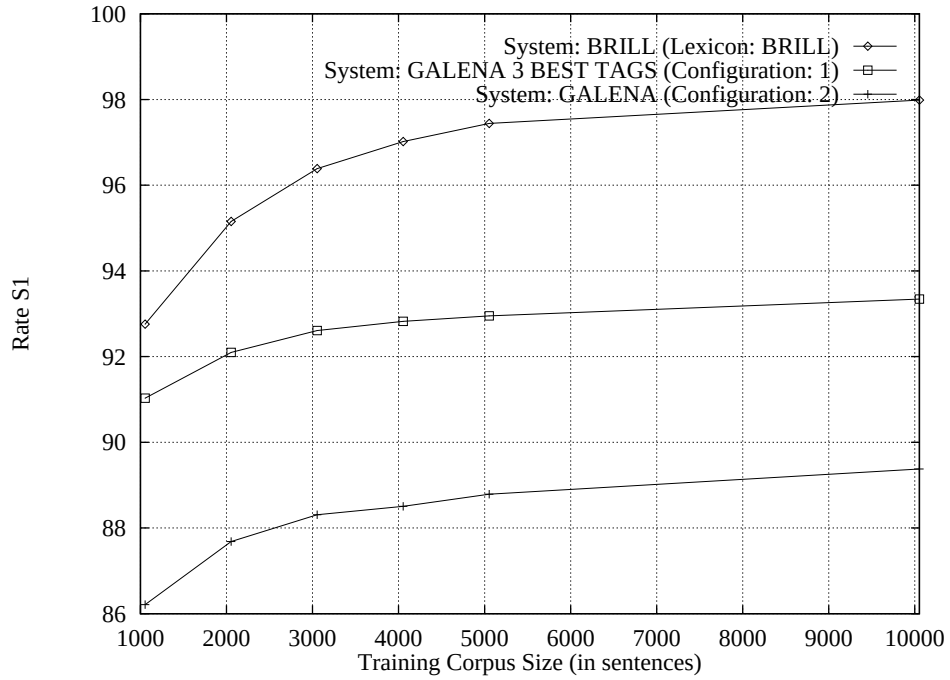


Figure 6.57: System: BRILL (Lexicon: BRILL) vs. GALENA 3 BEST TAGS (Configuration: 1) vs. GALENA (Configuration: 2) - Rate: S1

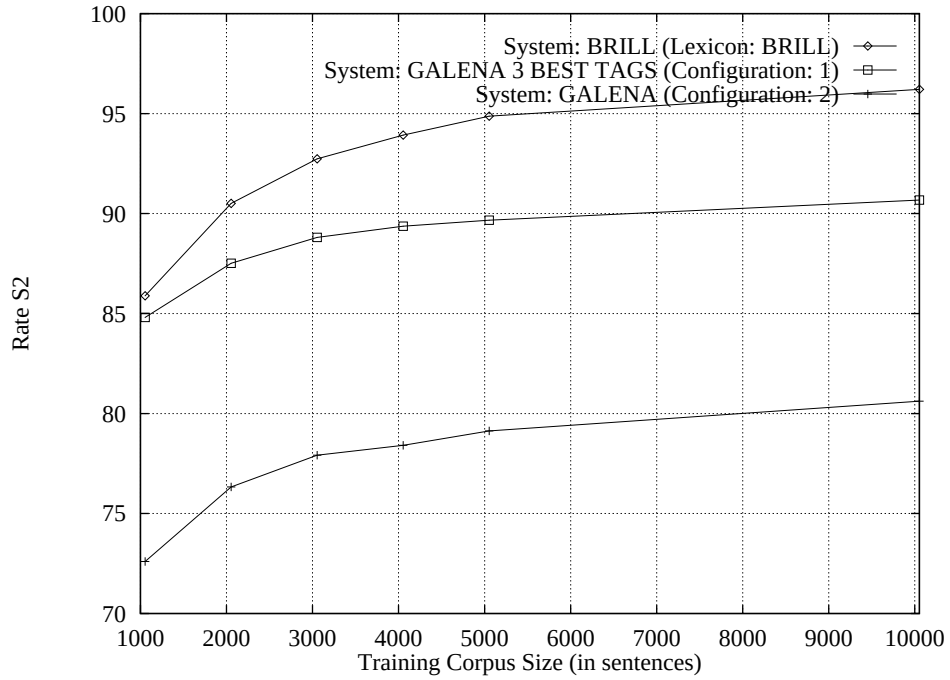


Figure 6.58: System: BRILL (Lexicon: BRILL) vs. GALENA 3 BEST TAGS (Configuration: 1) vs. GALENA (Configuration: 2) - Rate: S2

Chapter 7

Conclusion

As a conclusion of the present work, the first idea to show is that GALENA system has very fast training times. This was exactly the main objective for the experiment, together with the property of allowing the user interact with the system during the learning phase in order to reduce training times. Althoug the other parts of the system, such as the scanner and the parser, have demonstrated to be very robust, our disambiguation module is still a prototype, and at the end we have obtained a low performance in the disambiguation phase. For this reason, BRILL system and its results could be a reference for future implementations in the frame of the GALENA project.

The large amount of data that we have available now about the BRILL system shows that training phase in the BRILL system is very slow, but its general performance is very good. With a training corpus size of 4000 sentences, rates are already very close to the ideally highest values also obtained in this experiment. This general performance also is very similar to other rates that can be found for other tools and languages. Our intention is to repeat the same experiment for other languages (English, French, German) in the same conditions. Strategies and scripts for training, retagging and evaluation are perfectly defined now, so the only thing that we need is new data. Anyway, we can conclude that Spanish language does not present any special feature which make disambiguation process difficult, and a training corpus size of something between 3000 and 4000 sentences should be enough to obtain a good success rate in the tagging process.

Finally, in order to try to remove the small percentage of wrongly tagged words (2-3%), a feature always present in pure stochastic systems, we support the idea of using them in combination with syntactic information, that is, with parsing techniques, and this is our main subject of work for the future.

Bibliography

[1] L. Rabiner, B.H. Juang. Fundamentals of speech recognition. Chapter 6, Theory and implementation of Hidden Markov Models. Prentice-Hall, 1993.

[2] E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Wa., 1994.

[3] M. Vilares Ferro, A. Valderruten Vidal, J. Graña Gil, M.A. Alonso Pardo. Une Approche Formelle pour la Gènération d'Analyseurs de Langages Naturels. In *Proceedings of TALN'95, Philippe Blache (Ed)*, Marseille, France, 1995, pp. 246-255.

[4] Corpus Resources And Terminology ExtRaction (MLAP-93/20), creation of a set of tools and resources for multilingual corpus linguistics work. Lancaster University, UK; Computers, Communications and Visions, France; Universidad Autónoma de Madrid, Spain.

Appendix A

Galena extended tag set

The following list is the tag set used in the present experiment. It is simply the enumeration of the tags in the Galena system tag set, extended with the marks for compound verbal forms, verbal forms in passive voice and verbal forms with enclitic pronouns. A description of each tag can be obtained from the discussion in chapter 3. The cardinal of the tag set is 373 tags.

Afp0	Afpc	Afs0	Afsc	Afy0	Amp0
Ampc	Ams0	Amsc	Amy0	Ayp0	Aypc
Ays0	Aysc	Ayy0	Cc	Cs	Dfp
Dfs	Dmp	Dms	Dns	Enfp	Enfs
Enmp	Enms	Enns	Eyfp	Eyfs	Eymp
Eyms	Eyns	Eyyp	Eyys	Gnyp	Gnys
Gyfp	Gyfs	Gymp	Gyms	Gyyp	Gyys
Gyyy	Idfp	Idfs	Idmp	Idms	Idyp
Idys	Infp	Infs	Inmp	Inms	Inns
Inyp	Inys	Inyy	Iyfp	Iyfs	Iymp
Iyms	Iyyp	Iyys	Iyyy	Mdyp1s	Mdyp2s
Mdyp3p	Mdyp3s	Mdys1s	Mdys2s	Mdys3s	Mnfp1s
Mnfp2s	Mnfp3p	Mnfs1s	Mnfs2s	Mnfs3s	Mnmp1s
Mnmp2s	Mnmp3p	Mnms1s	Mnms2s	Mnms3s	Myfp1p
Myfp2p	Myfs1p	Myfs2p	Mymp1p	Mymp2p	Myms1p
Myms2p	Ncdms	Ncnms	Ncyfp	Ncyfs	Ncymp
Ncyyp	Ncyys	Nmnyp	Nmnys	Nonfp	Nonfs
Nonmp	Nonms	Noyfp	Noyfs	Noymp	Noyms
Npnfp	Npnfs	Npnmp	Npnms	P	Q!
Q”	Q(	Q)	Q,	Q-	Q.
Q...	Q:	Q;	Q?	Q‘!	Q‘?
Rp1pyy	Rp1syy	Rp2pyy	Rp2syy	Rp3paf	Rp3pam
Rp3pdy	Rp3saf	Rp3sam	Rp3sdy	Rp3yyy	Rt1pqf
Rt1pqm	Rt1sny	Rt1spy	Rt2pqf	Rt2pqm	Rt2sny
Rt2spy	Rt3pqf	Rt3pqm	Rt3sqf	Rt3sqm	Rt3sqy
Rt3ypy	Scfp	Scfs	Scfy	Scmp	Scms
Scmy	Scyp	Scys	Scyy	Spfp	Spfs
Spfy	Spmp	Spms	Spyp	Spys	Spyy
Tdfp	Tdfs	Tdmp	Tdms	Tnyp	Tnys
Tnyy	Tyfp	Tyfs	Tymp	Tyms	Tyyp
Tyys	V000f0	V000f0CT1	V000f0EP1	V000f0EP1CT1	V000f0EP1PCT1
V000f0PCT1	V000g0	V000g0CT1	V000g0EP1	V000g0EP1CT1	V000g0EP1PCT1
V000g0PCT1	V0p0pf	V0p0pfPCT2	V0p0pfPCT3	V0p0pm	V0p0pmPCT2

V0p0pmPCT3 V0s0pf V0s0pfPCT2 V0s0pfPCT3 V0s0pm V0s0pmCT2
V0s0pmPCT2 V0s0pmPCT3 V1pci0 V1pci0CT1 V1pci0PCT1 V1pei0
V1pei0CT1 V1pei0PCT1 V1pfi0 V1pfi0CT1 V1pfi0PCT1 V1pfs0
V1pfs0CT1 V1pfs0PCT1 V1pii0 V1pii0CT1 V1pii0PCT1 V1pis0
V1pis0CT1 V1pis0PCT1 V1ppi0 V1ppi0CT1 V1ppi0PCT1 V1pps0
V1pps0CT1 V1pps0PCT1 V1pss0 V1pss0CT1 V1pss0PCT1 V1sei0
V1sei0CT1 V1sei0PCT1 V1sfi0 V1sfi0CT1 V1sfi0PCT1 V1spi0
V1spi0CT1 V1spi0PCT1 V2pci0 V2pci0CT1 V2pci0PCT1 V2pei0
V2pei0CT1 V2pei0PCT1 V2pfi0 V2pfi0CT1 V2pfi0PCT1 V2pfs0
V2pfs0CT1 V2pfs0PCT1 V2pii0 V2pii0CT1 V2pii0PCT1 V2pis0
V2pis0CT1 V2pis0PCT1 V2ppi0 V2ppi0CT1 V2ppi0PCT1 V2ppm0
V2ppm0EP1 V2pps0 V2pps0CT1 V2pps0PCT1 V2pss0 V2pss0CT1
V2pss0PCT1 V2sci0 V2sci0CT1 V2sci0PCT1 V2sei0 V2sei0CT1
V2sei0PCT1 V2sfi0 V2sfi0CT1 V2sfi0PCT1 V2sfs0 V2sfs0CT1
V2sfs0PCT1 V2sii0 V2sii0CT1 V2sii0PCT1 V2sis0 V2sis0CT1
V2sis0PCT1 V2spi0 V2spi0CT1 V2spi0PCT1 V2spm0 V2spm0EP1
V2sps0 V2sps0CT1 V2sps0PCT1 V2sss0 V2sss0CT1 V2sss0PCT1
V3pci0 V3pci0CT1 V3pci0PCT1 V3pei0 V3pei0CT1 V3pei0PCT1
V3pfi0 V3pfi0CT1 V3pfi0PCT1 V3pfs0 V3pfs0CT1 V3pfs0PCT1
V3pii0 V3pii0CT1 V3pii0PCT1 V3pis0 V3pis0CT1 V3pis0PCT1
V3ppi0 V3ppi0CT1 V3ppi0PCT1 V3pps0 V3pps0CT1 V3pps0PCT1
V3pss0 V3pss0CT1 V3pss0PCT1 V3sei0 V3sei0CT1 V3sei0PCT1
V3sfi0 V3sfi0CT1 V3sfi0PCT1 V3spi0 V3spi0CT1 V3spi0PCT1
V3sps0 Vysci0 Vysci0CT1 Vysci0PCT1 Vysfs0 Vysfs0CT1
Vysfs0PCT1 Vysii0 Vysii0CT1 Vysii0PCT1 Vysis0 Vysis0CT1
Vysis0PCT1 Vysps0 Vysps0CT1 Vysps0PCT1 Vysss0 Vysss0CT1
Vysss0PCT1 Wi Wm Wn Wr Wv
Wy Y Ze00 Zefs Zemp Zems
Zeys Zeyy Zgfp Zgfs Zgms Zo00
Zoms

Appendix B

Crater tag set

The following list is the Crater project tag set. It consists of two columns: the first one is the tag, and the second one is a description of the tag, sometimes including word examples. The cardinal of the tag set is 475 tags.

Tag	Description (examples)
ACRNM	Acronym (`ADN`)
ADJCP	Plural general comparative adjective (`mayores`, `menores`)
ADJCS	Singular general comparative adjective (`mayor`, `menor`)
ADJGFP	Feminine plural general positive adjective
ADJGFS	Feminine singular general positive adjective
ADJGMP	Masculine plural general positive adjective
ADJGMS	Masculine singular general positive adjective
ADJSFP	Feminine plural general superlative adjective (`máximas`, `mínimas`)
ADJSFS	Feminine singular general superlative adjective (`máxima`, `mínima`)
ADJSMP	Masculine plural general superlative adjective (`máximos`, `mínimos`)
ADJSMS	Masculine singular general superlative adjective (`máximo`, `mínimo`, `grandísimo`)
ADVGR	Positive degree adverb (`muy`, `demasiado`, `mucho`)
ADVGRC	Comparative degree adverb (`más`, `menos`)
ADVGRS	Superlative degree adverb (`abundantísimamente`)
ADVINT	Interrogative adverb (`cómo`)
ADVL	Locative adverb underspecified for directionality (`abajo`)
ADVLD	Dynamic locative adverb (`adelante`)
ADVLE	Static locative adverb (`dentro`)
ADVLID	Interrogative dynamic locative adverb (`adónde`)
ADVLIN	Interrogative locative adverb (`dónde`)
ADVLP	Locative adverb with proximal deixis (`aquí`)
ADVLR	Locative adverb with remote deixis (`allí`)
ADVLRD	Relative dynamic locative adverb (`adonde`)
ADVLRE	Relative locative adverb underspecified for directionality (`donde`)
ADVMRE	Relative modal adverb (`como`)
ADVN	General adverb (`salvajemente`, `bien`, `probablemente`)
ADVNEG	General negative adverb (`tampoco`)
ADVT	Temporal adverb (`ahora`, `ayer`)
ADVTIN	Interrogative temporal adverb (`cuándo`)
ADVTNE	Negative temporal adverb (`nunca`)
ADVTRE	Relative temporal adverb (`cuando`)
ALFP	Plural letter of the alphabet (`As/Aes`, `bes`)
ALFS	Singular letter of the alphabet (`A`, `b`)
ARCAFS	Feminine singular indefinite article and cardinal capable of pronominal function (`una`)

ARCAMS	Masculine singular indefinite article and non pronominal cardinal (un)
ARQUFP	Feminine plural indefinite article and quantifier capable of pronominal function (unas)
ARQUMP	Masculine plural indefinite article and quantifier capable of pronominal function (unos)
ARTDFP	Feminine plural definite article (las)
ARTDFS	Feminine singular definite article (la)
ARTDMP	Masculine plural definite article (los)
ARTDMS	Masculine singular definite article (el)
ARTDNS	Neuter singular definite article (lo)
CARDFP	Plural femenine cardinal capable of pronominal function (doscientas)
CARDGU	Hyphenated cardinals (40-50, 1850-1990)
CARDMP	Plural masculine cardinal capable of pronominal function (doscientos)
CARDPS	Singular pronominal cardinal (uno)
CARDXP	Plural cardinal neutral for gender (dos, tres, mil)
CARDXS	Singular cardinal as digit (1)
CARNMP	Non pronominal plural masculine cardinal (veintiún)
CC	Coordinating conjunction (y, o)
CCAD	Adversative coordinating conjunction (pero)
CCNEG	Negative coordinating conjunction (ni)
CODE	Alphanumeric code
CQUE	que (as conjunction)
CSUBF	Subordinating conjunction that introduces finite clauses (apenas)
CSUBI	Subordinating conjunction that introduces infinite clauses (al)
CSUBX	Subordinating conjunction underspecified for subord-type (aunque)
DMDPFP	Pronominal feminine plural demonstrative with distal deixis (ésas)
DMDPFS	Pronominal feminine singular demonstrative with distal deixis (ésa)
DMDPMP	Pronominal masculine plural demonstrative with distal deixis (ésos)
DMDPMS	Pronominal masculine singular demonstrative with distal deixis (ése)
DMDPNS	Pronominal neuter singular demonstrative with distal deixis (eso)
DMDXFP	Feminine plural demonstrative (capable of pronominal function) with distal deixis (esas)
DMDXFS	Feminine singular demonstrative (capable of pronominal function) with distal deixis (esa)
DMDXMP	Masculine plural demonstrative (capable of pronominal function) with distal deixis (esos)
DMDXMS	Masculine singular demonstrative (capable of pronominal function) with distal deixis (ese)
DMPPFP	Pronominal feminine plural demonstrative with proximal deixis (éstas)
DMPPFS	Pronominal feminine singular demonstrative with proximal deixis (ésta)
DMPPMP	Pronominal masculine plural demonstrative with proximal deixis (éstos)
DMPPMS	Pronominal masculine singular demonstrative with proximal deixis (éste)
DMPPNS	Pronominal neuter singular demonstrative with proximal deixis (esto)
DMPXFP	Feminine plural demonstrative (capable of pronominal function) with proximal deixis (estas)
DMPXFS	Feminine singular demonstrative (capable of pronominal function) with proximal deixis (esta)
DMPXMP	Masculine plural demonstrative (capable of pronominal function) with proximal deixis (estos)
DMPXMS	Masculine singular demonstrative (capable of pronominal function) with proximal deixis (este)
DMRPFP	Pronominal feminine plural demonstrative with remote deixis (aquéllas)
DMRPFS	Pronominal feminine singular demonstrative with remote deixis (aquélla)
DMRPMP	Pronominal masculine plural demonstrative with remote deixis (aquéllos)

DMRPMS	Pronominal masculine singular demonstrative with remote deixis (`aquél`)
DMRPNS	Pronominal neuter singular demonstrative with remote deixis (`aquello`)
DMRXFP	Feminine plural demonstrative (capable of pronominal function) with remote deixis (`aquellas`)
DMRXFS	Feminine singular demonstrative (capable of pronominal function) with remote deixis (`aquella`)
DMRXMP	Masculine plural demonstrative (capable of pronominal function) with remote deixis (`aquellos`)
DMRXMS	Masculine singular demonstrative (capable of pronominal function) with remote deixis (`aquel`)
FO	Formula
INTPXP	Plural interrogative pronoun for humans neutral for gender (`quiénes`)
INTPXS	Singular interrogative pronoun for humans neutral for gender (`quién`)
INTXFP	Feminine plural interrogative capable of pronominal function (`cuántas`)
INTXFS	Feminine singular interrogative capable of pronominal function for inanimates (`cuánta`)
INTXMP	Masculine plural interrogative capable of pronominal function (`cuántos`)
INTXMS	Masculine and neuter singular interrogative capable of pronominal function for inanimates (`cuánto`)
INTXXP	Plural interrogative neutral for gender capable of pronominal function (`cuáles`)
INTXXS	Singular interrogative neutral for gender capable of pronominal function (`cuál`)
INTXXX	Interrogative capable of pronominal function neutral for gender and number (`qué`)
ITJN	Interjection (`oh`, `ja`)
NCFP	Feminine plural common noun (`mesas`, `manos`)
NCFS	Feminine singular common noun (`mesa`, `mano`)
NCMP	Masculine plural common noun (`libros`, `ordenadores`)
NCMS	Masculine singular common noun (`libro`, `ordenador`)
NEG	Negation
NMEAFP	Feminine plural unit of measure noun (`hectáreas`, `micras`)
NMEAFS	Feminine singular unit of measure noun (`hectárea`, `micra`)
NMEAMP	Masculine plural unit of measure noun (`metros`, `litros`)
NMEAMS	Masculine singular unit of measure noun (`metro`, `litro`)
NMON	Month nouns (singular and plural) (`diciembre(s)`)
NPAFP	Feminine plural proper anthroponymous noun (`Marías`)
NPAFS	Feminine singular proper anthroponymous noun (`María`)
NPAMP	Masculine plural proper anthroponymous noun (`Juanes`)
NPAMS	Masculine singular proper anthroponymous noun (`Juan`)
NPAXX	Proper anthroponymous noun neutral for gender and number (`Rodríguez`, `Sanchís`)
NPGP	Other plural proper nouns (`Aberri Egunas`)
NPGS	Other singular proper nouns (`Eutelsat`)
NPGX	Other proper nouns (default tag for unknown proper nouns)
NPOS	Singular proper collective nouns (`Iberia`)
NPTOP	Plural proper toponym or collective noun (`Coreas`)
NPTOS	Singular proper toponym or collective noun (`Madrid`)
NPTP	Plural proper toponym noun (`Pirineos`)
NPTS	Singular proper toponym noun (`Guadalquivir`)
NWEE	Weekday nouns (singular and plural) (`sábado(s)`)
ORDNMS	Masculine singular non pronominal ordinal (`primer`, `tercer`)
ORDXFP	Feminine plural ordinal capable of pronominal function (`primeras`, `segundas`)
ORDXFS	Feminine singular ordinal capable of pronominal function (`primera`, `segunda`)
ORDXMP	Masculine plural ordinal capable of pronominal function (`primeros`, `segundos`)
ORDXMS	Masculine singular ordinal capable of pronominal function (`primero`, `segundo`)
PAL	Portmanteau word formed by `a` and `el`

PDEL	Portmanteau word formed by `de` and `el`
PE	Foreign word
PNC	Unclassified word
PPC1P	Clitic personal pronoun, first person plural DO/IO (`nos`)
PPC1S	Clitic personal pronoun, first person singular DO/IO (`me`)
PPC2P	Clitic personal pronoun, second person plural DO/IO (`os`)
PPC2S	Clitic personal pronoun, second person singular DO/IO (`te`)
PPC3P	Clitic personal pronoun, third person plural DO/IO (`les`)
PPC3S	Clitic personal pronoun, third person singular DO/IO (`le`)
PPN1S	Personal pronoun, first person singular nominative (`yo`)
PPN2S	Personal pronoun, second person singular nominative (`tú`)
PPO3FP	Clitic personal pronoun, feminine third person plural DO (`las`)
PPO3FS	Clitic personal pronoun, feminine third person singular DO (`la`)
PPO3MP	Clitic personal pronoun, masculine third person plural DO (`los`)
PPO3XS	Clitic personal pronoun, masculine or neuter third person singular DO (`lo`)
PPOSFP	Feminine plural possessive pronoun (`tuyas`, `suyas`)
PPOSFS	Feminine singular possessive pronoun (`mía`, `tuya`)
PPOSMP	Masculine plural possessive pronoun (`míos`, `tuyos`)
PPOSMS	Masculine singular possessive pronoun (`tuyo`, `suyo`)
PPOSPP	Plural prenominal possessive pronoun (`mis`, `tus`, `sus`)
PPOSPS	Singular prenominal possessive pronoun (`mi`, `tu`, `su`)
PPP1S	Personal pronoun, first person singular oblique (`mí`)
PPP2S	Personal pronoun, second person singular oblique (`ti`)
PPP3X	Personal pronoun, third person neutral for number oblique (`sí`)
PPX1FP	Personal pronoun, feminine first person plural nominative or oblique (`nosotras`)
PPX1MP	Personal pronoun, masculine first person plural nominative or oblique (`nosotros`)
PPX2FP	Personal pronoun, feminine second person plural nominative or oblique (`vosotras`)
PPX2MP	Personal pronoun, masculine second person plural nominative or oblique (`vosotros`)
PPX3FP	Personal pronoun, feminine third person plural nominative or oblique (`ellas`)
PPX3FS	Personal pronoun, feminine third person singular nominative or oblique (`ella`)
PPX3MP	Personal pronoun, masculine third person plural nominative or oblique (`ellos`)
PPX3MS	Personal pronoun, masculine third person singular nominative or oblique (`él`)
PPX3NS	Personal pronoun, neuter third person singular nominative or oblique (`ello`)
PPXT2P	Personal pronoun, second person plural polite nominative or oblique (`ustedes`)
PPXT2S	Personal pronoun, second person singular polite nominative or oblique (`usted`)
PREP	Preposition
PREPN	Negative preposition (`sin`)
QUDF	Feminine plural distributive quantifier (`sendas`)
QUDM	Masculine plural distributive quantifier (`sendos`)
QUDX	Distributive quantifier neutral for gender (`cada`)
QUNFP	Feminine plural non pronominal quantifier (`ciertas`)
QUNFS	Feminine singular non pronominal quantifier (`cualquier`)
QUNMP	Masculine plural non pronominal quantifier (`ciertos`)
QUNMS	Masculine singular non pronominal quantifier (`algún`)
QUNNMS	Masculine singular non pronominal quantifier with negative polarity (`ningún`)
QUPA	Singular pronominal quantifier for humans (`alguien`)
QUPI	Singular pronominal quantifier for non-humans (`algo`)
QUPMUL	Singular pronominal quantifier that indicates multiples (`doble`, `triple`)
QUPNA	Singular pronominal quantifier for humans with negative polarity (`nadie`)
QUPNI	Masculine singular pronominal quantifier for inanimates with negative polarity (`nada`)
QUPNX	Masculine singular pronominal quantifier with negative polarity underspecified for animates and inanimates (`ninguno`)

QUXFP	Feminine plural quantifier capable of pronominal function (`todas`, `algunas`, `cualesquiera`)
QUXFS	Feminine singular quantifier capable of pronominal function (`toda`, `alguna`, `cualquiera`)
QUXMP	Masculine plural quantifier capable of pronominal function (`todos`, `algunos`, `cualesquiera`)
QUXMS	Masculine singular quantifier capable of pronominal function (`todo`, `alguno`, `cualquiera`)
QUXNFP	Femenine plural quantifier capable of pronominal function with negative polarity (`ningunas`)
QUXNFS	Feminine singular quantifier capable of pronominal function with negative polarity (`ninguna`)
QUXNMP	Masculine plural quantifier capable of pronominal function with negative polarity (`ningunos`)
RELPFP	Feminine plural possessive relative pronoun (`cuyas`)
RELPFS	Feminine singular possessive relative pronoun (`cuya`)
RELPMP	Masculine plural possessive relative pronoun (`cuyos`)
RELPMS	Masculine singular possessive relative pronoun (`cuyo`)
RELPXP	Pural relative pronoun for animates, neutral for gender (`quienes`)
RELPXS	Singular relative pronoun for animates, neutral for gender (`quien`)
RELXFP	Feminine plural relative pronoun capable of pronominal function (`cuantas`)
RELXFS	Feminine singular relative pronoun capable of pronominal function (`cuanta`)
RELXMP	Masculine plural relative pronoun capable of pronominal function (`cuantos`)
RELXMS	Masculine singular relative pronoun capable of pronominal function (`cuanto`)
ROMAN	Roman number (`XI`)
SE	`se` (as particle)
TRATF	Feminine noun of title (`Sra.`, `Dña.`, `Exma.`)
TRATM	Masculine noun of title (`Sr.`, `D.`, `Prof.`, `Exmo.`)
UMFX	Feminine unit of measurement, neutral for number (`pta.`)
UMMX	Masculine unit of measurement, neutral for number (`cm.`)
VECI1P	Verb estar. Indicative conditional tense first person plural
VECI1S	Verb estar. Indicative conditional tense first person singular
VECI2P	Verb estar. Indicative conditional tense second person plural
VECI2S	Verb estar. Indicative conditional tense second person singular
VECI3P	Verb estar. Indicative conditional tense thrid person plural
VECI3S	Verb estar. Indicative conditional tense third person singular
VEFI1P	Verb estar. Indicative future tense first person plural
VEFI1S	Verb estar. Indicative future tense first person singular
VEFI2P	Verb estar. Indicative future tense second person plural
VEFI2S	Verb estar. Indicative future tense second person singular
VEFI3P	Verb estar. Indicative future tense third person plural
VEFI3S	Verb estar. Indicative future tense third person singular
VEFS1P	Verb estar. Subjunctive future tense first person plural
VEFS1S	Verb estar. Subjunctive future tense first person singular
VEFS2P	Verb estar. Subjunctive future tense second person plural
VEFS2S	Verb estar. Subjunctive future tense second person singular
VEFS3P	Verb estar. Subjunctive future tense third person plural
VEFS3S	Verb estar. Subjunctive future tense third person singular
VEGER	Verb estar. Gerund
VEII1P	Verb estar. Indicative imperfect tense first person plural
VEII1S	Verb estar. Indicative imperfect tense first person singular
VEII2P	Verb estar. Indicative imperfect tense second person plural

VEII2S	Verb estar. Indicative imperfect tense second person singular
VEII3P	Verb estar. Indicative imperfect tense third person plural
VEII3S	Verb estar. Indicative imperfect tense third person singular
VEINF	Verb estar. Infinitive
VEIS1P	Verb estar. Subjunctive imperfect tense first person plural
VEIS1S	Verb estar. Subjunctive imperfect tense first person singular
VEIS2P	Verb estar. Subjunctive imperfect tense second person plural
VEIS2S	Verb estar. Subjunctive imperfect tense second person singular
VEIS3P	Verb estar. Subjunctive imperfect tense third person plural
VEIS3S	Verb estar. Subjunctive imperfect tense third person singular
VEPI1P	Verb estar. Indicative present tense first person plural
VEPI1S	Verb estar. Indicative present tense first person singular
VEPI2P	Verb estar. Indicative present tense second person plural
VEPI2S	Verb estar. Indicative present tense second person singular
VEPI3P	Verb estar. Indicative present tense third person plural
VEPI3S	Verb estar. Indicative present tense third person singular
VEPM2P	Verb estar. Imperative second person plural
VEPM2S	Verb estar. Imperative second person singular
VEPS1P	Verb estar. Subjunctive present tense first person plural
VEPS1S	Verb estar. Subjunctive present tense first person singular
VEPS2P	Verb estar. Subjunctive present tense second person plural
VEPS2S	Verb estar. Subjunctive present tense second person singular
VEPS3P	Verb estar. Subjunctive present tense third person plural
VEPS3S	Verb estar. Subjunctive present tense third person singular
VEPX	Verb estar. Past participle
VEXI1P	Verb estar. Indicative preterite tense first person plural
VEXI1S	Verb estar. Indicative preterite tense first person singular
VEXI2P	Verb estar. Indicative preterite tense second person plural
VEXI2S	Verb estar. Indicative preterite tense second person singular
VEXI3P	Verb estar. Indicative preterite tense third person plural
VEXI3S	Verb estar. Indicative preterite tense third person singular
VHCI1P	Verb haber. Indicative conditional tense first person plural
VHCI1S	Verb haber. Indicative conditional tense first person singular
VHCI2P	Verb haber. Indicative conditional tense second person plural
VHCI2S	Verb haber. Indicative conditional tense second person singular
VHCI3P	Verb haber. Indicative conditional tense third person plural
VHCI3S	Verb haber. Indicative conditional tense third person singular
VHFI1P	Verb haber. Indicative future tense first person plural
VHFI1S	Verb haber. Indicative future tense first person singular
VHFI2P	Verb haber. Indicative future tense second person plural
VHFI2S	Verb haber. Indicative future tense second person singular
VHFI3P	Verb haber. Indicative future tense third person plural
VHFI3S	Verb haber. Indicative future tense thrid person singular
VHFS1P	Verb haber. Subjunctive future tense first person plural
VHFS1S	Verb haber. Subjunctive future tense first person singular
VHFS2P	Verb haber. Subjunctive future tense second person plural
VHFS2S	Verb haber. Subjunctive future tense second person singular
VHFS3P	Verb haber. Subjunctive future tense third person plural
VHFS3S	Verb haber. Subjunctive future tense third person singular
VHGER	Verb haber. Gerund
VHII1P	Verb haber. Indicative imperfect tense first person plural
VHII1S	Verb haber. Indicative imperfect tense first person singular

VHII2P	Verb haber. Indicative imperfect tense second person plural
VHII2S	Verb haber. Indicative imperfect tense second person singular
VHII3P	Verb haber. Indicative imperfect tense third person plural
VHII3S	Verb haber. Indicative imperfect tense third person singular
VHINF	Verb haber. Infinitive
VHIS1P	Verb haber. Subjunctive imperfect tense first person plural
VHIS1S	Verb haber. Subjunctive imperfect tense first person singular
VHIS2P	Verb haber. Subjunctive imperfect tense second person plural
VHIS2S	Verb haber. Subjunctive imperfect tense second person singular
VHIS3P	Verb haber. Subjunctive imperfect tense third person plural
VHIS3S	Verb haber. Subjunctive imperfect tense third person singular
VHPI1P	Verb haber. Indicative present tense first person plural
VHPI1S	Verb haber. Indicative present tense first person singular
VHPI2P	Verb haber. Indicative present tense second person plural
VHPI2S	Verb haber. Indicative present tense second person singular
VHPI3E	Verb haber. Indicative present tense third person singular existential
VHPI3P	Verb haber. Indicative present tense third person plural
VHPI3S	Verb haber. Indicative present tense third person singular
VHPM2P	Verb haber. Imperative second person plural
VHPM2S	Verb haber. Imperative second person singular
VHPS1P	Verb haber. Subjunctive present tense first person plural
VHPS1S	Verb haber. Subjunctive present tense first person singular
VHPS2P	Verb haber. Subjunctive present tense second person plural
VHPS2S	Verb haber. Subjunctive present tense second person singular
VHPS3P	Verb haber. Subjunctive present tense third person plural
VHPS3S	Verb haber. Subjunctive present tense third person singular
VHPXFP	Verb haber. Feminine plural past participle
VHPXFS	Verb haber. Feminine singular past participle
VHPXMP	Verb haber. Masculine plural past participle
VHPXMS	Verb haber. Masculine singular past participle
VHXI1P	Verb haber. Indicative preterite tense first person plural
VHXI1S	Verb haber. Indicative preterite tense first person singular
VHXI2P	Verb haber. Indicative preterite tense second person plural
VHXI2S	Verb haber. Indicative preterite tense second person singular
VHXI3P	Verb haber. Indicative preterite tense third person plural
VHXI3S	Verb haber. Indicative preterite tense third person singular
VLCI1P	Lexical verb. Indicative conditional tense first person plural
VLCI1S	Lexical verb. Indicative conditional tense first person singular
VLCI2P	Lexical verb. Indicative conditional tense second person plural
VLCI2S	Lexical verb. Indicative conditional tense second person singular
VLCI3P	Lexical verb. Indicative conditional tense third person plural
VLCI3S	Lexical verb. Indicative conditional tense third person singular
VLFI1P	Lexical verb. Indicative future tense first person plural
VLFI1S	Lexical verb. Indicative future tense first person singular
VLFI2P	Lexical verb. Indicative future tense second person plural
VLFI2S	Lexical verb. Indicative future tense second person singular
VLFI3P	Lexical verb. Indicative future tense third person plural
VLFI3S	Lexical verb. Indicative future tense thrid person singular
VLFS1P	Lexical verb. Subjunctive future tense first person plural
VLFS1S	Lexical verb. Subjunctive future tense first person singular
VLFS2P	Lexical verb. Subjunctive future tense second person plural
VLFS2S	Lexical verb. Subjunctive future tense second person singular

VLFS3P	Lexical verb. Subjunctive future tense third person plural
VLFS3S	Lexical verb. Subjunctive future tense third person singular
VLGER	Lexical verb. Gerund
VLII1P	Lexical verb. Indicative imperfect tense first person plural
VLII1S	Lexical verb. Indicative imperfect tense first person singular
VLII2P	Lexical verb. Indicative imperfect tense second person plural
VLII2S	Lexical verb. Indicative imperfect tense second person singular
VLII3P	Lexical verb. Indicative imperfect tense third person plural
VLII3S	Lexical verb. Indicative imperfect tense third person singular
VLINF	Lexical verb. Infinitive
VLIS1P	Lexical verb. Subjunctive imperfect tense first person plural
VLIS1S	Lexical verb. Subjunctive imperfect tense first person singular
VLIS2P	Lexical verb. Subjunctive imperfect tense second person plural
VLIS2S	Lexical verb. Subjunctive imperfect tense second person singular
VLIS3P	Lexical verb. Subjunctive imperfect tense third person plural
VLIS3S	Lexical verb. Subjunctive imperfect tense third person singular
VLPI1P	Lexical verb. Indicative present tense first person plural
VLPI1S	Lexical verb. Indicative present tense first person singular
VLPI2P	Lexical verb. Indicative present tense second person plural
VLPI2S	Lexical verb. Indicative present tense second person singular
VLPI3P	Lexical verb. Indicative present tense third person plural
VLPI3S	Lexical verb. Indicative present tense third person singular
VLPM2P	Lexical verb. Imperative second person plural
VLPM2S	Lexical verb. Imperative second person singular
VLPPFP	Lexical verb. Feminine plural present participle
VLPPFS	Lexical verb. Feminine singular present participle
VLPPMP	Lexical verb. Masculine plural present participle
VLPPMS	Lexical verb. Masculine singular present participle
VLPS1P	Lexical verb. Subjunctive present tense first person plural
VLPS1S	Lexical verb. Subjunctive present tense first person singular
VLPS2P	Lexical verb. Subjunctive present tense second person plural
VLPS2S	Lexical verb. Subjunctive present tense second person singular
VLPS3P	Lexical verb. Subjunctive present tense third person plural
VLPS3S	Lexical verb. Subjunctive present tense third person singular
VLPXFP	Lexical verb. Feminine plural past participle
VLPXFS	Lexical verb. Feminine singular past participle
VLPXMP	Lexical verb. Masculine plural past participle
VLPXMS	Lexical verb. Masculine singular past participle
VLXI1P	Lexical verb. Indicative preterite tense first person plural
VLXI1S	Lexical verb. Indicative preterite tense first person singular
VLXI2P	Lexical verb. Indicative preterite tense second person plural
VLXI2S	Lexical verb. Indicative preterite tense second person singular
VLXI3P	Lexical verb. Indicative preterite tense third person plural
VLXI3S	Lexical verb. Indicative preterite tense third person singular
VMCI1P	Modal verb. Indicative conditional tense first person plural
VMCI1S	Modal verb. Indicative conditional tense first person singular
VMCI2P	Modal verb. Indicative conditional tense second person plural
VMCI2S	Modal verb. Indicative conditional tense second person singular
VMCI3P	Modal verb. Indicative conditional tense third person plural
VMCI3S	Modal verb. Indicative conditional tense third person singular
VMFI1P	Modal verb. Indicative future tense first person plural
VMFI1S	Modal verb. Indicative future tense first person singular

VMFI2P	Modal verb. Indicative future tense second person plural
VMFI2S	Modal verb. Indicative future tense second person singular
VMFI3P	Modal verb. Indicative future tense third person plural
VMFI3S	Modal verb. Indicative future tense thrid person singular
VMFS1P	Modal verb. Subjunctive future tense first person plural
VMFS1S	Modal verb. Subjunctive future tense first person singular
VMFS2P	Modal verb. Subjunctive future tense second person plural
VMFS2S	Modal verb. Subjunctive future tense second person singular
VMFS3P	Modal verb. Subjunctive future tense third person plural
VMFS3S	Modal verb. Subjunctive future tense third person singular
VMGER	Modal verb. Gerund
VMII1P	Modal verb. Indicative imperfect tense first person plural
VMII1S	Modal verb. Indicative imperfect tense first person singular
VMII2P	Modal verb. Indicative imperfect tense second person plural
VMII2S	Modal verb. Indicative imperfect tense second person singular
VMII3P	Modal verb. Indicative imperfect tense third person plural
VMII3S	Modal verb. Indicative imperfect tense third person singular
VMINF	Modal verb. Infinitive
VMIS1P	Modal verb. Subjunctive imperfect tense first person plural
VMIS1S	Modal verb. Subjunctive imperfect tense first person singular
VMIS2P	Modal verb. Subjunctive imperfect tense second person plural
VMIS2S	Modal verb. Subjunctive imperfect tense second person singular
VMIS3P	Modal verb. Subjunctive imperfect tense third person plural
VMIS3S	Modal verb. Subjunctive imperfect tense third person singular
VMPI1P	Modal verb. Indicative present tense first person plural
VMPI1S	Modal verb. Indicative present tense first person singular
VMPI2P	Modal verb. Indicative present tense second person plural
VMPI2S	Modal verb. Indicative present tense second person singular
VMPI3P	Modal verb. Indicative present tense third person plural
VMPI3S	Modal verb. Indicative present tense third person singular
VMPM2P	Modal verb. Imperative second person plural
VMPM2S	Modal verb. Imperative second person singular
VMPS1P	Modal verb. Subjunctive present tense first person plural
VMPS1S	Modal verb. Subjunctive present tense first person singular
VMPS2P	Modal verb. Subjunctive present tense second person plural
VMPS2S	Modal verb. Subjunctive present tense second person singular
VMPS3P	Modal verb. Subjunctive present tense third person plural
VMPS3S	Modal verb. Subjunctive present tense third person singular
VMPX	Modal verb. Past participle
VMXI1P	Modal verb. Indicative preterite tense first person plural
VMXI1S	Modal verb. Indicative preterite tense first person singular
VMXI2P	Modal verb. Indicative preterite tense second person plural
VMXI2S	Modal verb. Indicative preterite tense second person singular
VMXI3P	Modal verb. Indicative preterite tense third person plural
VMXI3S	Modal verb. Indicative preterite tense third person singular
VSCI1P	Verb ser. Indicative conditional tense first person plural
VSCI1S	Verb ser. Indicative conditional tense first person singular
VSCI2P	Verb ser. Indicative conditional tense second person plural
VSCI2S	Verb ser. Indicative conditional tense second person singular
VSCI3P	Verb ser. Indicative conditional tense third person plural
VSCI3S	Verb ser. Indicative conditional tense third person singular
VSFI1P	Verb ser. Indicative future tense first person plural

VSFI1S	Verb ser. Indicative future tense first person singular
VSFI2P	Verb ser. Indicative future tense second person plural
VSFI2S	Verb ser. Indicative future tense second person singular
VSFI3P	Verb ser. Indicative future tense third person plural
VSFI3S	Verb ser. Indicative future tense thrid person singular
VSFS1P	Verb ser. Subjunctive future tense first person plural
VSFS1S	Verb ser. Subjunctive future tense first person singular
VSFS2P	Verb ser. Subjunctive future tense second person plural
VSFS2S	Verb ser. Subjunctive future tense second person singular
VSFS3P	Verb ser. Subjunctive future tense third person plural
VSFS3S	Verb ser. Subjunctive future tense third person singular
VSGER	Verb ser. Gerund
VSII1P	Verb ser. Indicative imperfect tense first person plural
VSII1S	Verb ser. Indicative imperfect tense first person singular
VSII2P	Verb ser. Indicative imperfect tense second person plural
VSII2S	Verb ser. Indicative imperfect tense second person singular
VSII3P	Verb ser. Indicative imperfect tense third person plural
VSII3S	Verb ser. Indicative imperfect tense third person singular
VSINF	Verb ser. Infinitive
VSIS1P	Verb ser. Subjunctive imperfect tense first person plural
VSIS1S	Verb ser. Subjunctive imperfect tense first person singular
VSIS2P	Verb ser. Subjunctive imperfect tense second person plural
VSIS2S	Verb ser. Subjunctive imperfect tense second person singular
VSIS3P	Verb ser. Subjunctive imperfect tense third person plural
VSIS3S	Verb ser. Subjunctive imperfect tense third person singular
VSPI1P	Verb ser. Indicative present tense first person plural
VSPI1S	Verb ser. Indicative present tense first person singular
VSPI2P	Verb ser. Indicative present tense second person plural
VSPI2S	Verb ser. Indicative present tense second person singular
VSPI3P	Verb ser. Indicative present tense third person plural
VSPI3S	Verb ser. Indicative present tense third person singular
VSPM2P	Verb ser. Imperative second person plural
VSPM2S	Verb ser. Imperative second person singular
VSPS1P	Verb ser. Subjunctive present tense first person plural
VSPS1S	Verb ser. Subjunctive present tense first person singular
VSPS2P	Verb ser. Subjunctive present tense second person plural
VSPS2S	Verb ser. Subjunctive present tense second person singular
VSPS3P	Verb ser. Subjunctive present tense third person plural
VSPS3S	Verb ser. Subjunctive present tense third person singular
VSPX	Verb ser. Past participle
VSXI1P	Verb ser. Indicative preterite tense first person plural
VSXI1S	Verb ser. Indicative preterite tense first person singular
VSXI2P	Verb ser. Indicative preterite tense second person plural
VSXI2S	Verb ser. Indicative preterite tense second person singular
VSXI3P	Verb ser. Indicative preterite tense third person plural
VSXI3S	Verb ser. Indicative preterite tense third person singular

Appendix C

Mapping CRATER tag set ⟼ GALENA extended tag set

The following list of pairs is the mapping CRATER tag set ⟼ GALENA extended tag set, which was applied to the ITU corpus in order to obtain our reference corpus for the experiment.

ACRNM → Zgfs
ADJCP → Aypc
ADJCS → Aysc
ADJGFP → Afp0
ADJGFS → Afs0
ADJGMP → Amp0
ADJGMS → Ams0
ADJSFP → Afpc
ADJSFS → Afsc
ADJSMP → Ampc
ADJSMS → Amsc
ADVGR → Wm
ADVGRC → Wy
ADVGRS → Wn
ADVINT → Wv
ADVL → Wn
ADVLD → Wn
ADVLE → Wn
ADVLID → Wn
ADVLIN → Wv
ADVLP → Wn
ADVLR → Wn
ADVLRD → Wr
ADVLRE → Wr
ADVMRE → Wr
ADVN → Wy
ADVNEG → Wn
ADVT → Wn
ADVTIN → Wv
ADVTNE → Wn
ADVTRE → Wr
ALFP → Scfp
ALFS → Scfs
ARCAFS → Iyfs
ARCAMS → Idms
ARQUFP → Iyfp
ARQUMP → Iymp
ARTDFP → Dfp
ARTDFS → Dfs
ARTDMP → Dmp
ARTDMS → Dms
ARTDNS → Dns
CARDFP → Ncyfp
CARDGU → Ays0
CARDMP → Ncymp
CARDPS → Ncnms
CARDXP → Ncyyp
CARDXS → Ncyys
CARNMP → Ncdmp
CC → Cc
CCAD → Cs
CCNEG → Cc
CODE → Scms
CQUE → Cs
CSUBF → Cs
CSUBI → Cs
CSUBX → Cs
DMDPFP → Enfp
DMDPFS → Enfs
DMDPMP → Enmp
DMDPMS → Enms
DMDPNS → Eyns
DMDXFP → Eyfp
DMDXFS → Eyfs
DMDXMP → Eymp
DMDXMS → Eyms
DMPPFP → Enfp
DMPPFS → Enfs
DMPPMP → Enmp
DMPPMS → Enms
DMPPNS → Enns
DMPXFP → Eyfp
DMPXFS → Eyfs
DMPXMP → Eymp
DMPXMS → Eyms
DMRPFP → Enfp
DMRPFS → Enfs
DMRPMP → Enmp
DMRPMS → Enms
DMRPNS → Enns
DMRXFP → Eyfp
DMRXFS → Eyfs
DMRXMP → Eymp
DMRXMS → Eyms
FO → Zf
INTPXP → Gnyp
INTPXS → Gnys
INTXFP → Gyfp
INTXFS → Gyfs
INTXMP → Gymp
INTXMS → Gyms
INTXXP → Gyyp
INTXXS → Gyys
INTXXX → Gyyy
ITJN → Y
NCFP → Scfp
NCFS → Scfs
NCMP → Scmp
NCMS → Scms
NEG → Wn
NMEAFP → Scfp
NMEAFS → Scfs
NMEAMP → Scmp
NMEAMS → Scms
NMON → Scmy
NPAFP → Spfp
NPAFS → Spfs
NPAMP → Spmp
NPAMS → Spms
NPAXX → Spyy
NPGP → Spyp
NPGS → Spys
NPGX → Spyy
NPOS → Spys
NPTOP → Spyp
NPTOS → Spys
NPTP → Spyp
NPTS → Spys
NWEE → Scmy
ORDNMS → Nonms
ORDXFP → Nonfp
ORDXFS → Nonfs
ORDXMP → Nonmp
ORDXMS → Nonms
PAL → P Dms
PDEL → P Dms
PE → Ze00
PNC → U

PPC1P → Re1pyy
PPC1S → Re1syy
PPC2P → Re2pyy
PPC2S → Re2syy
PPC3P → Re3pdy
PPC3S → Re3sdy
PPN1S → Rt1sny
PPN2S → Rt2sny
PPO3FP → Re3paf
PPO3FS → Re3saf
PPO3MP → Re3pam
PPO3XS → Re3sam
PPOSFP → Mnfp2s
PPOSFS → Mnfs2s
PPOSMP → Mnmp2s
PPOSMS → Mnms3y
PPOSPP → Mdyp3s
PPOSPS → Mdys3s
PPP1S → Rt1spy
PPP2S → Rt2spy
PPP3X → Rt3ypy
PPX1FP → Rt1pqf
PPX1MP → Rt1pqm
PPX2FP → Rt2pqf
PPX2MP → Rt2pqm
PPX3FP → Rt3pqf
PPX3FS → Rt3sqf
PPX3MP → Rt3pqm
PPX3MS → Rt3sqm
PPX3NS → Rt3sqn
PPXT2P → Rt300y
PPXT2S → Rt300y
PREP → P
PREPN → P
QUDF → Idfp
QUDM → Idmp
QUDX → Iyys
QUNFP → Idfp
QUNFS → Idys
QUNMP → Idmp
QUNMS → Idms
QUNNMS → Idms
QUPA → Inms
QUPI → Inns
QUPMUL → Nmnys
QUPNA → Inms
QUPNI → Inns
QUPNX → Inms
QUXFP → Iyfp
QUXFS → Iyfs
QUXMP → Iymp
QUXMS → Iyms
QUXNFP → Iyfp
QUXNFS → Iyfs
QUXNMP → Iymp
RELPFP → Tdfp
RELPFS → Tdfs
RELPMP → Tdmp
RELPMS → Tdms
RELPXP → Tnyp
RELPXS → Tnys
RELXFP → Tyfp
RELXFS → Tyfs
RELXMP → Tymp
RELXMS → Tyms
ROMAN → Ncyyp
SE → Rp3yyy
TRATF → Afs0
TRATM → Ams0
UMFX → Scfy
UMMX → Scmy
VECI1P → V1pci0
VECI1S → V1sci0
VECI2P → V2pci0
VECI2S → V2sci0
VECI3P → V3pci0
VECI3S → V3sci0
VEFI1P → V1pfi0
VEFI1S → V1sfi0
VEFI2P → V2pfi0
VEFI2S → V2sfi0
VEFI3P → V3pfi0
VEFI3S → V3sfi0
VEFS1P → V1pfs0
VEFS1S → V1sfs0
VEFS2P → V2pfs0
VEFS2S → V2sfs0
VEFS3P → V3pfs0
VEFS3S → V3sfs0
VEGER → V000g0
VEII1P → V1pii0
VEII1S → V1sii0
VEII2P → V2pii0
VEII2S → V2sii0
VEII3P → V3pii0
VEII3S → V3sii0
VEINF → V000f0
VEIS1P → V1pss0
VEIS1S → V1sss0
VEIS2P → V2pss0
VEIS2S → V2sss0
VEIS3P → V3pss0
VEIS3S → V3sss0
VEPI1P → V1ppi0
VEPI1S → V1spi0
VEPI2P → V2ppi0
VEPI2S → V2spi0
VEPI3P → V3ppi0
VEPI3S → V3spi0
VEPM2P → V2ppm0
VEPM2S → V2spm0
VEPS1P → V1pps0
VEPS1S → V1sps0
VEPS2P → V2pps0
VEPS2S → V2sps0
VEPS3P → V3pps0
VEPS3S → V3sps0
VEPX → V0s0pm
VEXI1P → V1pei0
VEXI1S → V1sei0
VEXI2P → V2pei0
VEXI2S → V2sei0
VEXI3P → V3pei0
VEXI3S → V3sei0
VHCI1P → V1pci0
VHCI1S → V1sci0
VHCI2P → V2pci0
VHCI2S → V2sci0
VHCI3P → V3pci0
VHCI3S → V3sci0
VHFI1P → V1pfi0
VHFI1S → V1sfi0
VHFI2P → V2pfi0
VHFI2S → V2sfi0
VHFI3P → V3pfi0
VHFI3S → V3sfi0
VHFS1P → V1pfs0
VHFS1S → V1sfs0
VHFS2P → V2pfs0
VHFS2S → V2sfs0
VHFS3P → V3pfs0
VHFS3S → V3sfs0
VHGER → V000g0
VHII1P → V1pii0
VHII1S → V1sii0
VHII2P → V2pii0
VHII2S → V2sii0
VHII3P → V3pii0
VHII3S → V3sii0
VHINF → V000f0
VHIS1P → V1pss0
VHIS1S → V1sss0
VHIS2P → V2pss0
VHIS2S → V2sss0
VHIS3P → V3pss0
VHIS3S → V3sss0
VHPI1P → V1ppi0
VHPI1S → V1spi0
VHPI2P → V2ppi0
VHPI2S → V2spi0
VHPI3E → V3spi0
VHPI3P → V3ppi0
VHPI3S → V3spi0
VHPM2P → V2ppm0
VHPM2S → V2spm0
VHPS1P → V1pps0
VHPS1S → V1sps0
VHPS2P → V2pps0
VHPS2S → V2sps0
VHPS3P → V3pps0
VHPS3S → V3sps0
VHPXFP → V0p0pf
VHPXFS → V0s0pf
VHPXMP → V0p0pm
VHPXMS → V0s0pm
VHXI1P → V1pei0
VHXI1S → V1sei0
VHXI2P → V2pei0
VHXI2S → V2sei0
VHXI3P → V3pei0
VHXI3S → V3sei0
VLCI1P → V1pci0
VLCI1S → V1sci0
VLCI2P → V2pci0
VLCI2S → V2sci0
VLCI3P → V3pci0
VLCI3S → V3sci0
VLFI1P → V1pfi0
VLFI1S → V1sfi0
VLFI2P → V2pfi0
VLFI2S → V2sfi0
VLFI3P → V3pfi0
VLFI3S → V3sfi0
VLFS1P → V1pfs0
VLFS1S → V1sfs0
VLFS2P → V2pfs0
VLFS2S → V2sfs0
VLFS3P → V3pfs0
VLFS3S → V3sfs0
VLGER → V000g0
VLII1P → V1pii0
VLII1S → V1sii0
VLII2P → V2pii0
VLII2S → V2sii0
VLII3P → V3pii0
VLII3S → V3sii0
VLINF → V000f0
VLIS1P → V1pss0

VLIS1S → V1sss0
VLIS2P → V2pss0
VLIS2S → V2sss0
VLIS3P → V3pss0
VLIS3S → V3sss0
VLPI1P → V1ppi0
VLPI1S → V1spi0
VLPI2P → V2ppi0
VLPI2S → V2spi0
VLPI3P → V3ppi0
VLPI3S → V3spi0
VLPM2P → V2ppm0
VLPM2S → V2spm0
VLPPFP → V0p0pf
VLPPFS → V0s0pf
VLPPMP → V0p0pm
VLPPMS → V0s0pm
VLPS1P → V1pps0
VLPS1S → V1sps0
VLPS2P → V2pps0
VLPS2S → V2sps0
VLPS3P → V3pps0
VLPS3S → V3sps0
VLPXFP → V0p0pf
VLPXFS → V0s0pf
VLPXMP → V0p0pm
VLPXMS → V0s0pm
VLXI1P → V1pei0
VLXI1S → V1sei0
VLXI2P → V2pei0
VLXI2S → V2sei0
VLXI3P → V3pei0
VLXI3S → V3sei0
VMCI1P → V1pci0
VMCI1S → V1sci0
VMCI2P → V2pci0
VMCI2S → V2sci0
VMCI3P → V3pci0
VMCI3S → V3sci0
VMFI1P → V1pfi0
VMFI1S → V1sfi0
VMFI2P → V2pfi0
VMFI2S → V2sfi0
VMFI3P → V3pfi0
VMFI3S → V3sfi0
VMFS1P → V1pfs0
VMFS1S → V1sfs0
VMFS2P → V2pfs0
VMFS2S → V2sfs0
VMFS3P → V3pfs0
VMFS3S → V3sfs0
VMGER → V000g0
VMII1P → V1pii0
VMII1S → V1sii0
VMII2P → V2pii0
VMII2S → V2sii0
VMII3P → V3pii0
VMII3S → V3sii0
VMINF → V000f0
VMIS1P → V1pss0
VMIS1S → V1sss0
VMIS2P → V2pss0
VMIS2S → V2sss0
VMIS3P → V3pss0
VMIS3S → V3sss0
VMPI1P → V1ppi0
VMPI1S → V1spi0
VMPI2P → V2ppi0
VMPI2S → V2spi0
VMPI3P → V3ppi0
VMPI3S → V3spi0
VMPM2P → V2ppm0
VMPM2S → V2spm0
VMPS1P → V1pps0
VMPS1S → V1sps0
VMPS2P → V2pps0
VMPS2S → V2sps0
VMPS3P → V3pps0
VMPS3S → V3sps0
VMPX → V0s0pm
VMXI1P → V1pei0
VMXI1S → V1sei0
VMXI2P → V2pei0
VMXI2S → V2sei0
VMXI3P → V3pei0
VMXI3S → V3sei0
VSCI1P → V1pci0
VSCI1S → V1sci0
VSCI2P → V2pci0
VSCI2S → V2sci0
VSCI3P → V3pci0
VSCI3S → V3sci0
VSFI1P → V1pfi0
VSFI1S → V1sfi0
VSFI2P → V2pfi0
VSFI2S → V2sfi0
VSFI3P → V3pfi0
VSFI3S → V3sfi0
VSFS1P → V1pfs0
VSFS1S → V1sfs0
VSFS2P → V2pfs0
VSFS2S → V2sfs0
VSFS3P → V3pfs0
VSFS3S → V3sfs0
VSGER → V000g0
VSII1P → V1pii0
VSII1S → V1sii0
VSII2P → V2pii0
VSII2S → V2sii0
VSII3P → V3pii0
VSII3S → V3sii0
VSINF → V000f0
VSIS1P → V1pss0
VSIS1S → V1sss0
VSIS2P → V2pss0
VSIS2S → V2sss0
VSIS3P → V3pss0
VSIS3S → V3sss0
VSPI1P → V1ppi0
VSPI1S → V1spi0
VSPI2P → V2ppi0
VSPI2S → V2spi0
VSPI3P → V3ppi0
VSPI3S → V3spi0
VSPM2P → V2ppm0
VSPM2S → V2spm0
VSPS1P → V1pps0
VSPS1S → V1sps0
VSPS2P → V2pps0
VSPS2S → V2sps0
VSPS3P → V3pps0
VSPS3S → V3sps0
VSPX → V0s0pm
VSXI1P → V1pei0
VSXI1S → V1sei0
VSXI2P → V2pei0
VSXI2S → V2sei0
VSXI3P → V3pei0
VSXI3S → V3sei0

Appendix D

PERL scripts and utilities

The following pieces of PERL source code are the main scripts which implement all the steps of each experiment, and the main utilities for different tasks like changing text files from one syntax or format to another, or the calculus of features of corpora and lexica, or the building of directory trees for data, or the calculus of scores, ...

D.1 crater_corpus_to_brill_corpus.prl

The syntax of the original ITU CORPUS is a file with lines

```
form1_tag1_lemma1 form2_tag2_lemma2 ... formn_tagn_lemman
```

with tag1, tag2, ..., tagn in the CRATER TAG SET. Therefore, this script simply replace character _ with character /, and applies the mapping of appendix C in order to translate tags from CRATER TAG SET to GALENA EXTENDED TAG SET. We do not directly include the script because its size is too large.

D.2 sentence_line_corpus_to_word_line_corpus.prl

```
#!/usr/bin/perl
#
# sentence_line_corpus_to_word_line_corpus.prl
#
#    We assume <corpus> a corpus with lines
#
#       form1/tag1/lemma1 form2/tag2/lemma2 ... formn/tagn/lemman ./Q./.
#
#       (lines could also end with ?/Q?/? or .../Q.../... or another
#        punctuation mark, and thus each line is a sentence)
#
#    This program uses <corpus> in order to generate another
#    corpus with lines
#
#       form1/tag1/lemma1
#       form2/tag2/lemma2
#       ...
#       formn/tagn/lemman
#       ./Q./.
```

```
#
#    Usage:
#
#       cat <corpus> | sentence_line_corpus_to_word_line_corpus.prl
#

while ($line = <STDIN>) {
    @words = split (" ", $line);

    foreach $w (@words) {
        print "$w\n";
    }
}
```

D.3 brill_corpus_to_galena_corpus.prl

```
#!/usr/bin/perl
#
# brill_corpus_to_galena_corpus.prl
#
#    We assume <corpus> a training corpus with lines
#
#       form1/tag1/lemma1 form2/tag2/lemma2 ... formn/tagn/lemman ./Q./.
#
#       (lines could also end with ?/Q?/? or .../Q.../... or another
#        punctuation mark, and thus each line is a sentence)
#
#    This program uses <corpus> in order to generate another
#    training corpus in the Galena syntax, that is,
#
#       => *["form1", (tag1), "lemma1"]
#       => *["form2", (tag2), "lemma2"]
#       ...
#       => *["formk", (tagk), "lemmak"]
#
#    k and n could be different, because Galena syntax needs to join
#    compound tenses and to split verbal forms with enclitic pronouns,
#    and this program takes this feature into account.
#
#    These are all the possible cases:
#
#       1       2       3       4       5       6       7
#    ------- ------- ------- ------- ------- ------- -------
#    EP1     CT1     EP1CT1  PCT1    PCT1    EP1PCT1 EP1PCT1
#            CT2     CT2     PCT2    PCT2    PCT2    PCT2
#                                    PCT3            PCT3
#
#    In 1, 3, 6 and 7 (an infinitive or a gerund with 1 enclitic pronoun),
#    split in this way:
#                                                      ["se",  (Re3yyy), "'el"]
#                                                      ["lo",  (Re3sam), "'el"]
```

```
#                                                       ["la",  (Re3saf), "'el"]
#       ["form - enclitic", (tag - EP1), "lemma"]  +    ["los", (Re3pam), "'el"]
#                                                       ["las", (Re3paf), "'el"]
#                                                       ["le",  (Re3sdy), "'el"]
#                                                       ["les", (Re3pdy), "'el"]
#
#    In 2, 3, 5 and 7 (compound tenses), join in this way:
#
#                               in 4th letter of the tagCT1, replace
#                                          p with P
#                                          e with E
#       ["formCT1 formCT2", (              i with I                  ), "lemmaCT2"]
#                                          s with S
#                                          f with F
#                                          c with C,
#           or in 5th letter (compound infinitive and compound gerund), replace
#                                          f with F,
#                                          g with G,
#                                    and remove CT1 or PCT1
#
#    Usage:
#
#       cat <corpus> | sentence_line_corpus_to_word_line_corpus.prl
#                    | brill_corpus_to_galena_corpus.prl
#

sub shift_lines {
    my $n = shift(@_);

    if ($n == 1) {
        $line1 = $line2;
        $line2 = $line3;
        $line3 = <STDIN>; chomp ($line3);
        return;
    }

    if ($n == 2) {
        $line1 = $line3;
        $line2 = <STDIN>; chomp ($line2);
        $line3 = <STDIN>; chomp ($line3);
        return;
    }

    $line1 = <STDIN>; chomp ($line1);
    $line2 = <STDIN>; chomp ($line2);
    $line3 = <STDIN>; chomp ($line3);
}

sub split_form_enclitic {
    my $form = shift (@_);
    my $enclitic = "";
    my @chars = split (//, $form);
```

```
    if ($chars[$#chars] eq "s") {
        $form = join ("", @chars[0..($#chars - 3)]);
        $enclitic = join ("", @chars[($#chars - 2)..$#chars]);
    } else {
        $form = join ("", @chars[0..($#chars - 2)]);
        $enclitic = join ("", @chars[($#chars - 1)..$#chars]);
    }
    return ($form, $enclitic);
}

sub print_enclitic {
    my $enclitic = shift (@_);
  SWITCH: {
      if ($enclitic eq "se")  { print "=> *[\"se\", (Re3yyy), \"'el\"]\n";  last SWITCH;}
      if ($enclitic eq "lo")  { print "=> *[\"lo\", (Re3sam), \"'el\"]\n";  last SWITCH;}
      if ($enclitic eq "la")  { print "=> *[\"la\", (Re3saf), \"'el\"]\n";  last SWITCH;}
      if ($enclitic eq "los") { print "=> *[\"los\", (Re3pam), \"'el\"]\n"; last SWITCH;}
      if ($enclitic eq "las") { print "=> *[\"las\", (Re3paf), \"'el\"]\n"; last SWITCH;}
      if ($enclitic eq "le")  { print "=> *[\"le\", (Re3sdy), \"'el\"]\n";  last SWITCH;}
      if ($enclitic eq "les") { print "=> *[\"les\", (Re3pdy), \"'el\"]\n"; last SWITCH;}
  }
}

shift_lines (3);

while ($line1 ne "") {
    ($form1, $tag1, $lemma1) = split ("/", $line1);
    ($form2, $tag2, $lemma2) = split ("/", $line2);
    ($form3, $tag3, $lemma3) = split ("/", $line3);

  CASES: {
      if (($tag1 =~ /EP1PCT1/) and ($tag2 =~ /PCT2/)) {
          if ($tag3 =~ /PCT3/) {
              # CASE 7
              $tag1 =~ s/EP1PCT1//;
              $tag2 =~ s/PCT2//;
              $tag3 =~ s/PCT3//;
              ($form1, $pronoun) = split_form_enclitic ($form1);
              @tagtemp = split(//, $tag1);
              if ($tagtemp[3] eq "0") {$tagtemp[4] = uc ($tagtemp[4])}
                  else {$tagtemp[3] = uc ($tagtemp[3])};
              $tag1 = join ("", @tagtemp);
              $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
              $form2 =~ s/_/ /g; $lemma2 =~ s/_/ /g;
              $form3 =~ s/_/ /g; $lemma3 =~ s/_/ /g;
              print "=> *[\"$form1 $form2\", ($tag1), \"$lemma2\"]\n";
              print_enclitic ($pronoun);
              print "=> *[\"$form3\", ($tag3), \"$lemma3\"]\n";
              shift_lines (3);
          } else {
              # CASE 6
              $tag1 =~ s/EP1PCT1//;
```

```
        $tag2 =~ s/PCT2//;
        ($form1, $pronoun) = split_form_enclitic ($form1);
        $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
        $form2 =~ s/_/ /g; $lemma2 =~ s/_/ /g;
        print "=> *[\"$form1\", ($tag1), \"$lemma1\"]\n";
        print_enclitic ($pronoun);
        print "=> *[\"$form2\", ($tag2), \"$lemma2\"]\n";
        shift_lines (2);
    }
    last CASES;
}

if (($tag1 =~ /PCT1/) and ($tag2 =~ /PCT2/)) {
    if ($tag3 =~ /PCT3/) {
        # CASE 5
        $tag1 =~ s/PCT1//;
        $tag2 =~ s/PCT2//;
        $tag3 =~ s/PCT3//;
        @tagtemp = split(//, $tag1);
        if ($tagtemp[3] eq "0") {$tagtemp[4] = uc($tagtemp[4])}
            else {$tagtemp[3] = uc($tagtemp[3])};
        $tag1 = join("", @tagtemp);
        $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
        $form2 =~ s/_/ /g; $lemma2 =~ s/_/ /g;
        $form3 =~ s/_/ /g; $lemma3 =~ s/_/ /g;
        print "=> *[\"$form1 $form2\", ($tag1), \"$lemma2\"]\n";
        print "=> *[\"$form3\", ($tag3), \"$lemma3\"]\n";
        shift_lines (3);
    } else {
        # CASE 4
        $tag1 =~ s/PCT1//;
        $tag2 =~ s/PCT2//;
        $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
        $form2 =~ s/_/ /g; $lemma2 =~ s/_/ /g;
        print "=> *[\"$form1\", ($tag1), \"$lemma1\"]\n";
        print "=> *[\"$form2\", ($tag2), \"$lemma2\"]\n";
        shift_lines (2);
    }
    last CASES;
}

if (($tag1 =~ /EP1CT1/) and ($tag2 =~ /CT2/)) {
    # CASE 3
    $tag1 =~ s/EP1CT1//;
    @tagtemp = split(//, $tag1);
    if ($tagtemp[3] eq "0") {$tagtemp[4] = uc($tagtemp[4])}
        else {$tagtemp[3] = uc($tagtemp[3])};
    $tag1 = join("", @tagtemp);
    ($form1, $pronoun) = split_form_enclitic($form1);
    $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
    $form2 =~ s/_/ /g; $lemma2 =~ s/_/ /g;
    print "=> *[\"$form1 $form2\", ($tag1), \"$lemma2\"]\n";
```

```
            print_enclitic ($pronoun);
            shift_lines (2);
            last CASES;
        }

        if (($tag1 =~ /CT1/) and ($tag2 =~ /CT2/)) {
            # CASE 2
            $tag1 =~ s/CT1//;
            @tagtemp = split(//, $tag1);
            if ($tagtemp[3] eq "0") {$tagtemp[4] = uc($tagtemp[4])}
                else {$tagtemp[3] = uc($tagtemp[3])};
            $tag1 = join("", @tagtemp);
            $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
            $form2 =~ s/_/ /g; $lemma2 =~ s/_/ /g;
            print "=> *[\"$form1 $form2\", ($tag1), \"$lemma2\"]\n";
            shift_lines (2);
            last CASES;
        }

        if ($tag1 =~ /EP1/) {
            # CASE 1
            $tag1 =~ s/EP1//;
            ($form1, $pronoun) = split_form_enclitic($form1);
            $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
            print "=> *[\"$form1\", ($tag1), \"$lemma1\"]\n";
            print_enclitic ($pronoun);
            shift_lines (1);
            last CASES;
        }

        # NOTHING SPECIAL
        $tag1 =~ s/EP1//;
        $tag1 =~ s/PCT1//;
        $tag1 =~ s/PCT2//;
        $tag1 =~ s/PCT3//;
        $tag1 =~ s/CT1//;
        $tag1 =~ s/CT2//;
        $form1 =~ s/_/ /g; $lemma1 =~ s/_/ /g;
        print "=> *[\"$form1\", ($tag1), \"$lemma1\"]\n";
        shift_lines (1);
        last CASES;
    }
}
```

D.4 divide_corpus_in_two_randomly.prl

```
#!/usr/bin/perl
#
# divide_corpus_in_two_randomly.prl
#
#    We assume <corpus> a corpus with lines
```

```
#
#       form1/tag1/lemma1 form2/tag2/lemma2 ... formn/tagn/lemman ./Q./.
#
#       (lines could also end with ?/Q?/? or .../Q.../... or another
#        punctuation mark, and thus each line is a sentence)
#
#    This program uses <corpus> in order to generate two corpora: one
#    with <n> lines (or sentences) randomly chosen, and the other
#    with the rest of the lines (or sentences).
#
#    Usage:
#
#       divide_corpus_in_two_randomly.prl <corpus> <n> <corpus-1> <corpus-2>
#

($#ARGV + 1 == 4) or die "\nUsage: $0 <corpus> <n> <corpus-1> <corpus-2>\n\n";
open (XX, "$ARGV[0]") or die "Can't open $ARGV[0]: $!\n";
($ARGV[1] > 0) or die "$ARGV[1] isn't a number\n";
open (YY, ">$ARGV[2]") or die "Can't open $ARGV[2]: $!\n";
open (ZZ, ">$ARGV[3]") or die "Can't open $ARGV[3]: $!\n";

$total = 0;
while (<XX>) {
    $total++;
}
close(XX);
($ARGV[1] <= $total) or die "Can't take $ARGV[1] lines from a file of $total lines\n";

open (XX, "$ARGV[0]") or die "Can't open $ARGV[0]: $!\n";
$wanted = $ARGV[1];

srand;
while ($wanted > 0) {
    $r = int (rand $total) + 1;
    if ($rand_table{$r} eq "") {
        $rand_table{$r} = 1;
        $wanted--;
    }
}

$l = 1;
while (<XX>) {
    if ($rand_table{$l} ne "") {
        print YY $_;
    } else {
        print ZZ $_;
    }
    $l++;
}

close(YY);
close(ZZ);
```

D.5 form_tag_lemma_to_form_tag.prl

```
#!/usr/bin/perl
#
# form_tag_lemma_to_form_tag.prl
#
#    We assume <corpus> a corpus with lines
#
#       form1/tag1/lemma1 form2/tag2/lemma2 ... formn/tagn/lemman ./Q./.
#
#       (lines could also end with ?/Q?/? or .../Q.../... or another
#        punctuation mark, and thus each line is a sentence)
#
#    This program uses <corpus> in order to generate another
#    corpus with lines
#
#       form1/tag1 form2/tag2 ... formn/tagn ./Q.
#
#    Usage:
#
#       cat <corpus> | form_tag_lemma_to_form_tag.prl
#

while ($line = <STDIN>) {
    @words = split(" ", $line);

    $beforelast = $#words - 1;
    foreach $w (@words[0..$beforelast]) {
        ($form, $tag, $lemma) = split ("/", $w);
        print "$form/$tag ";
    }

    ($form, $tag, $lemma) = split ("/", $words[$#words]);
    print "$form/$tag\n";
}
```

D.6 form_tag_to_form.prl

```
#!/usr/bin/perl
#
# form_tag_to_form.prl
#
#    We assume <corpus> a corpus with lines
#
#       form1/tag1 form2/tag2 ... formn/tagn ./Q.
#
#       (lines could also end with ?/Q? or .../Q... or another
#        punctuation mark, and thus each line is a sentence)
#
#    This program uses <corpus> in order to generate another
#    corpus with lines
```

```
#
#       form1 form2 ... formn .
#
#    Usage:
#
#       cat <corpus> | form_tag_to_form.prl
#

while ($line = <STDIN>) {
    @words = split (" ", $line);

    $beforelast = $#words - 1;
    foreach $w (@words[0..$beforelast]) {
        ($form, $tag) = split ("/", $w);
        print "$form ";
    }

    ($form, $tag) = split ("/", $words[$#words]);
    print "$form\n";
}
```

D.7 corpus_to_lexicon.prl

```
#!/usr/bin/perl
#
# corpus_to_lexicon.prl
#
#    We assume <corpus> a corpus with lines
#
#       form1/tag1/lemma1 form2/tag2/lemma2 ... formn/tagn/lemman ./Q./.
#
#    This program uses <corpus> in order to generate a lexicon with lines
#
#       form1 tag1
#       form2 tag2
#       ...
#       formn tagn
#
#    Usage:
#
#       cat <corpus> | corpus_to_lexicon.prl
#

while ($line = <STDIN>) {
    @words = split (" ", $line);
    foreach $w (@words) {
        ($form, $tag, $lemma) = split ("/", $w);
        print "$form $tag\n";
    }
}
```

D.8 features_lexicon.prl

```
#!/usr/bin/perl
#
# features_lexicon.prl
#
#    We assume <lex> a lexicon with lines
#
#       form tag1 tag2 ... tagn
#
#    This program calculates the number of lines and provides
#    statistics about the number of ambiguities in them.
#
#    Usage:
#
#       cat <lex> | features_lexicon.prl
#

while ($line = <STDIN>) {
    $n++;
    @data = split (" ", $line);
    $ambiguities{$#data}++;
}

foreach $num (sort keys %ambiguities) {
    $p += $num * $ambiguities{$num};
    print "*with $num tags: $ambiguities{$num} forms\n";
}

print "\nlexicon size: $n forms, $p posibilities\n";
```

D.9 features_corpus.prl

```
#!/usr/bin/perl
#
# features_corpus.prl
#
#    We assume <lex> a lexicon with lines
#
#       form tag1 tag2 ... tagn
#
#    and <corpus> and corpus with lines
#
#       form1/tag1 form2/tag2 ... formn/tagn ./Q.
#
#       (lines could also end with ?/Q? or .../Q... or another
#        punctuation mark, and thus each line is a sentence)
#
#    This program calculates the number of words in the corpus and provides
#    statistics about the number of ambiguities in them.
#
```

```
#    Usage:
#
#       features_corpus.prl <lex> <corpus>
#

($#ARGV + 1 == 2) or die "\nUsage: $0 <lex> <corpus>\n\n";
open (LL, "$ARGV[0]") or die "Can't open $ARGV[0]: $!\n";
open (XX, "$ARGV[1]") or die "Can't open $ARGV[1]: $!\n";

print STDERR "Loading lexicon...\n";
while ($lineLL = <LL>) {
    @fts = split (" ", $lineLL);
    $lexicon{$fts[0]} = $#fts;
}

print STDERR "Calculating features...\n";

while ($lineXX = <XX>) {
    @wordsXX = split(" ", $lineXX);

    foreach $i (0..$#wordsXX) {
        $n++;
        ($formXX, $tagXX) = split ("/", $wordsXX[$i]);
        $features{$lexicon{$formXX}}++;
    }
}

foreach $num (sort keys %features) {
    $p += $num * $features{$num};
    print "*with $num tags: $features{$num} forms\n";
}

print "\ncorpus size: $n words, $p posibilities\n";
```

D.10 tagged_words_to_lexicon.prl

```
#!/usr/bin/perl
#
# tagged_words_to_lexicon.prl
#
#    We assume <lex1> <lex2> ... <lexn> lexica with lines
#
#       form tag1 tag2 ... tagn
#
#    This program replace lines which have the same form with only one line
#    which contains the form and the union of the tags, sorted by frequency
#    from highest to lowest.
#
#    Usage:
#
#       cat <lex1> <lex2> ... <lexn> | sort | tagged_words_to_lexicon.prl
```

```
#

sub numeric_sort { $a <=> $b }

sub sort_tags {
    my %tag_occurrencies;
    my %occurrencies_tag;
    my $tags;

    foreach $t (@_) {
        $tag_occurrencies{$t}++;
    }
    foreach $t (keys %tag_occurrencies) {
        $occurrencies_tag{$tag_occurrencies{$t}} .= " $t";
    }
    foreach $n (sort numeric_sort keys %occurrencies_tag) {
        $tags = $occurrencies_tag{$n} . $tags;
    }
    return ($tags);
}

$line = <STDIN>;
@data = split (" ", $line);
$old_form = $data[0];
shift (@data);
$old_tags = join (" ", @data);

while ($line = <STDIN>) {
   @data = split(" ", $line);
   $form = $data[0];
   shift (@data);
   $tags = join (" ", @data);

   if ($form eq $old_form) {
       $old_tags .= " $tags";
   } else {
       @union_tags = sort_tags (split (" ", $old_tags));
       print "$old_form@union_tags\n";
       $old_form = $form;
       $old_tags = $tags;
   }
}

@union_tags = sort_tags (split (" ", $old_tags));
print "$old_form@union_tags\n";
```

D.11 prepare_data_for_experiment.prl

```
#!/usr/bin/perl
#
#    Usage:
```

```
#
#       prepare_data_for_experiment.prl <serial_number_of_the_experiment>
#

($#ARGV + 1 == 1) or die "\nUsage: $0 <serial_number_of_the_experiment>\n\n";

$NLPDIR = "/home/grana/suiza";

##############################################################################
print "*** PEPARING DATA FOR A DISAMBIGUATION EXPERIMENT\n\n";
print "*** Creating directories\n";

$command = "mkdir $NLPDIR/test/test$ARGV[0]1 $NLPDIR/test/test$ARGV[0]2
            $NLPDIR/test/test$ARGV[0]3 $NLPDIR/test/test$ARGV[0]4
            $NLPDIR/test/test$ARGV[0]5";
print "$command\n\n";
`$command`;

##############################################################################
print "*** Extracting 5000 random sentecenses from ITU Corpus\n";

$command = "$NLPDIR/bin/divide_corpus_in_two_randomly.prl
            $NLPDIR/data/CORPUS_ITU_form_tag_lemma 5000 CORPUS-1 CORPUS-2";
print "$command\n\n";
`$command`;

##############################################################################
print "*** Splitting random sentences in files of 1000 sentences\n";

$command = "split -l 1000 CORPUS-1 CORPUS-1";
print "$command\n\n";
`$command`;

$command = "rm -f CORPUS-1";
print "$command\n\n";
`$command`;

##############################################################################
print "*** Generating TRAINING_CORPUS_ZERO for Brill\n";

$command = "cat $NLPDIR/data/TRAINING_CORPUS_ZERO_form_tag_lemma |
            $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            TRAINING_CORPUS_ZERO_form_tag";
print "$command\n\n";
`$command`;

$command = "cp TRAINING_CORPUS_ZERO_form_tag $NLPDIR/test/test$ARGV[0]1";
print "$command\n\n";
`$command`;

$command = "cp TRAINING_CORPUS_ZERO_form_tag $NLPDIR/test/test$ARGV[0]2";
print "$command\n\n";
```

```
`$command`;

$command = "cp TRAINING_CORPUS_ZERO_form_tag $NLPDIR/test/test$ARGV[0]3";
print "$command\n\n";
`$command`;

$command = "cp TRAINING_CORPUS_ZERO_form_tag $NLPDIR/test/test$ARGV[0]4";
print "$command\n\n";
`$command`;

$command = "cp TRAINING_CORPUS_ZERO_form_tag $NLPDIR/test/test$ARGV[0]5";
print "$command\n\n";
`$command`;

$command = "rm -f TRAINING_CORPUS_ZERO_form_tag";
print "$command\n\n";
`$command`;

##############################################################################
print "*** Generating TRAINING_CORPUS_ONE for Brill\n";

$command = "cat CORPUS-1aa | $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            $NLPDIR/test/test$ARGV[0]1/TRAINING_CORPUS_ONE_form_tag";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab |
            $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            $NLPDIR/test/test$ARGV[0]2/TRAINING_CORPUS_ONE_form_tag";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab CORPUS-1ac |
            $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            $NLPDIR/test/test$ARGV[0]3/TRAINING_CORPUS_ONE_form_tag";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab CORPUS-1ac CORPUS-1ad |
            $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            $NLPDIR/test/test$ARGV[0]4/TRAINING_CORPUS_ONE_form_tag";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab CORPUS-1ac CORPUS-1ad CORPUS-1ae |
            $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            $NLPDIR/test/test$ARGV[0]5/TRAINING_CORPUS_ONE_form_tag";
print "$command\n\n";
`$command`;

##############################################################################
print "*** Generating TRAINING_CORPUS_ONE for Galena\n";
```

```
$command = "cat CORPUS-1aa |
            $NLPDIR/bin/sentence_line_corpus_to_word_line_corpus.prl |
            $NLPDIR/bin/brill_corpus_to_galena_corpus.prl >
            $NLPDIR/test/test$ARGV[0]1/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "gzip $NLPDIR/test/test$ARGV[0]1/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab |
            $NLPDIR/bin/sentence_line_corpus_to_word_line_corpus.prl |
            $NLPDIR/bin/brill_corpus_to_galena_corpus.prl >
            $NLPDIR/test/test$ARGV[0]2/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "gzip $NLPDIR/test/test$ARGV[0]2/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab CORPUS-1ac |
            $NLPDIR/bin/sentence_line_corpus_to_word_line_corpus.prl |
            $NLPDIR/bin/brill_corpus_to_galena_corpus.prl >
            $NLPDIR/test/test$ARGV[0]3/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "gzip $NLPDIR/test/test$ARGV[0]3/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab CORPUS-1ac CORPUS-1ad |
            $NLPDIR/bin/sentence_line_corpus_to_word_line_corpus.prl |
            $NLPDIR/bin/brill_corpus_to_galena_corpus.prl >
            $NLPDIR/test/test$ARGV[0]4/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "gzip $NLPDIR/test/test$ARGV[0]4/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "cat CORPUS-1aa CORPUS-1ab CORPUS-1ac CORPUS-1ad CORPUS-1ae |
            $NLPDIR/bin/sentence_line_corpus_to_word_line_corpus.prl |
            $NLPDIR/bin/brill_corpus_to_galena_corpus.prl >
            $NLPDIR/test/test$ARGV[0]5/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;
```

```
$command = "gzip $NLPDIR/test/test$ARGV[0]5/TRAINING_CORPUS_ONE_galena";
print "$command\n\n";
`$command`;

$command = "rm -f CORPUS-1aa CORPUS-1ab CORPUS-1ac CORPUS-1ad CORPUS-1ae";
print "$command\n\n";
`$command`;

##############################################################################
print "*** Generating CORPUS_TWO for Brill\n";

$command = "cat CORPUS-2 | $NLPDIR/bin/form_tag_lemma_to_form_tag.prl >
            CORPUS_TWO_form_tag";
print "$command\n\n";
`$command`;

$command = "cat CORPUS_TWO_form_tag | $NLPDIR/bin/form_tag_to_form.prl >
            CORPUS_TWO_form";
print "$command\n\n";
`$command`;

$command = "cp CORPUS_TWO_form_tag CORPUS_TWO_form $NLPDIR/test/test$ARGV[0]1";
print "$command\n\n";
`$command`;

$command = "cp CORPUS_TWO_form_tag CORPUS_TWO_form $NLPDIR/test/test$ARGV[0]2";
print "$command\n\n";
`$command`;

$command = "cp CORPUS_TWO_form_tag CORPUS_TWO_form $NLPDIR/test/test$ARGV[0]3";
print "$command\n\n";
`$command`;

$command = "cp CORPUS_TWO_form_tag CORPUS_TWO_form $NLPDIR/test/test$ARGV[0]4";
print "$command\n\n";
`$command`;

$command = "cp CORPUS_TWO_form_tag CORPUS_TWO_form $NLPDIR/test/test$ARGV[0]5";
print "$command\n\n";
`$command`;

$command = "rm -f CORPUS_TWO_form_tag";
print "$command\n\n";
`$command`;


##############################################################################
print "*** Generating CORPUS_TWO for Galena\n";

$command = "cat CORPUS-2 |
            $NLPDIR/bin/sentence_line_corpus_to_word_line_corpus.prl |
            $NLPDIR/bin/brill_corpus_to_galena_corpus.prl >
```

```
            CORPUS_TWO_galena_form_tag_lemma";
print "$command\n\n";
`$command`;

$command = "gzip CORPUS_TWO_galena_form_tag_lemma";
print "$command\n\n";
`$command`;

$command = "$NLPDIR/bin/replace_string.prl CORPUS_TWO_form _ ' ' >
            CORPUS_TWO_galena_form";
print "$command\n\n";
`$command`;

$command = "gzip CORPUS_TWO_galena_form";
print "$command\n\n";
`$command`;

$command = "rm -f CORPUS_TWO_form CORPUS-2";
print "$command\n\n";
`$command`;

################################################################################
print "*** End\n";
```

D.12 experiment.prl

```
#!/usr/bin/perl
#
#    Usage:
#
#       experiment.prl
#

$NLPDIR = "/home/grana/suiza/Brill";
$BRILLORIGINALDIR = "/home/grana/suiza/Brill/RULE_BASED_TAGGER_V1.14";

sub set_start_time {
    $start_second = time;
    $start_second_cpu = (times)[2];
    $start_second_sys = (times)[3];
}

sub print_time {
    use integer;
    my $sec = time - $start_second;
    my ($hour, $sec_temp) = ($sec / 3600, $sec % 3600);
    my ($min, $sec) = ($sec_temp / 60, $sec_temp % 60);
    my $sec_cpu = (times)[2];
    $sec_cpu = $sec_cpu - $start_second_cpu;
    my ($hour_cpu, $sec_temp_cpu) = ($sec_cpu / 3600, $sec_cpu % 3600);
    my ($min_cpu, $sec_cpu) = ($sec_temp_cpu / 60, $sec_temp_cpu % 60);
```

```
    my $sec_sys = (times)[3];
    $sec_sys = $sec_sys - $start_second_sys;
    my ($hour_sys, $sec_temp_sys) = ($sec_sys / 3600, $sec_sys % 3600);
    my ($min_sys, $sec_sys) = ($sec_temp_sys / 60, $sec_temp_sys % 60);
    printf "%02d:%02d:%02d(date) %02d:%02d:%02d(cpu) %02d:%02d:%02d(sys)",
            $hour, $min, $sec, $hour_cpu, $min_cpu, $sec_cpu, $hour_sys,
            $min_sys, $sec_sys;
    no integer;
}

set_start_time ();
$step = 0;
print "\n*** BRILL TRAINING PROCESS\n";

###########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Splitting TRAINING_CORPUS_ZERO + TRAINING_CORPUS_ONE in two rand parts\n";

$command = "cat TRAINING_CORPUS_ZERO_form_tag TRAINING_CORPUS_ONE_form_tag |
            $BRILLORIGINALDIR/Utilities/divide-in-two-rand.prl CORPUS-1 CORPUS-2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "ln -s CORPUS-1 TAGGED-CORPUS";
print "$command\n";
`$command`;

###########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Untagging TRAINING_CORPUS_ZERO + TRAINING_CORPUS_ONE\n";

$command = "cat TRAINING_CORPUS_ZERO_form_tag TRAINING_CORPUS_ONE_form_tag |
            $BRILLORIGINALDIR/Utilities/tagged-to-untagged.prl > UNTAGGED-CORPUS";
print "$command\n";
`$command`;

###########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Creating BIGWORDLIST, a list of words occurring in UNTAGGED-CORPUS,
       sorted by decreasing frequency\n";

$command = "cat UNTAGGED-CORPUS | $BRILLORIGINALDIR/Utilities/wordlist-make.prl |
            sort +1 -rn | awk '{ print \$1}' > BIGWORDLIST";
print "$command\n";
`$command`;

###########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Creating SMALLWORDTAGLIST, a list of the form (word tag count) listing
       the number of times a word is tagging with a tag in TAGGED-CORPUS\n";
```

```
$command = "cat TAGGED-CORPUS | $BRILLORIGINALDIR/Utilities/word-tag-count.prl |
            sort +2 -rn > SMALLWORDTAGLIST";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Creating BIGBIGRAMLIST, a list of all word pairs occurring in UNTAGGED-CORPUS\n";

$command = "cat UNTAGGED-CORPUS | $BRILLORIGINALDIR/Utilities/bigram-generate.prl |
            awk '{ print \$1,\$2}' > BIGBIGRAMLIST";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Running the RULE LEARNER (this process is very slow)\n";

$command = "$BRILLORIGINALDIR/Learner_Code/unknown-lexical-learn.prl BIGWORDLIST
            SMALLWORDTAGLIST BIGBIGRAMLIST 300 LEXRULEOUTFILE";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Learning CONTEXTUAL CUES\n";

$command = "ln -s CORPUS-2 TAGGED-CORPUS-2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "cat TRAINING_CORPUS_ZERO_form_tag TRAINING_CORPUS_ONE_form_tag >
            TAGGED-CORPUS-ENTIRE";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Creating TRAINING.LEXICON\n";

$command = "cat TAGGED-CORPUS |
            $BRILLORIGINALDIR/Utilities/make-restricted-lexicon.prl > TRAINING.LEXICON";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Creating FINAL.LEXICON\n";
$command = "cat TAGGED-CORPUS-ENTIRE |
            $BRILLORIGINALDIR/Utilities/make-restricted-lexicon.prl > FINAL.LEXICON";
print "$command\n";
```

```
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Preparing to run\n";

$command = "cat TAGGED-CORPUS-2 |
            $BRILLORIGINALDIR/Utilities/tagged-to-untagged.prl > UNTAGGED-CORPUS-2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "cp $BRILLORIGINALDIR/Bin_and_Data/start-state-tagger .";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "cp $BRILLORIGINALDIR/Bin_and_Data/final-state-tagger .";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$BRILLORIGINALDIR/Bin_and_Data/tagger TRAINING.LEXICON UNTAGGED-CORPUS-2
             BIGBIGRAMLIST LEXRULEOUTFILE /dev/null -w BIGWORDLIST
             -i DUMMY-TAGGED-CORPUS > /dev/null";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Learning contextual rules\n";

$command = "$BRILLORIGINALDIR/Bin_and_Data/contextual-rule-learn TAGGED-CORPUS-2
            DUMMY-TAGGED-CORPUS CONTEXT-RULEFILE TRAINING.LEXICON";
print "$command\n";
`$command`;

$command = "ln -s FINAL.LEXICON LEXICON_BRILL";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Tagging CORPUS_TWO_form with LEXICON_BRILL\n";

$command = "$BRILLORIGINALDIR/Bin_and_Data/tagger LEXICON_BRILL CORPUS_TWO_form
            BIGBIGRAMLIST LEXRULEOUTFILE CONTEXT-RULEFILE -s 1000 >
            CORPUS_THREE_form_tag.brill";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
```

```
$command = "$NLPDIR/bin/scores.prl LEXICON_BRILL CORPUS_TWO_form_tag
            CORPUS_THREE_form_tag.brill > scores_brill";
print "$command\n";
`$command`;

##########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Tagging CORPUS_TWO_form with LEXICON_BRILL_GALENA\n";

$command = "$BRILLORIGINALDIR/Utilities/combine-lexicons.prl LEXICON_BRILL
            $NLPDIR/data/LEXICON_GALENA > LEXICON_BRILL_GALENA";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$BRILLORIGINALDIR/Bin_and_Data/tagger LEXICON_BRILL_GALENA
            CORPUS_TWO_form BIGBIGRAMLIST LEXRULEOUTFILE CONTEXT-RULEFILE -s 1000 >
            CORPUS_THREE_form_tag.brill_galena";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$NLPDIR/bin/scores.prl LEXICON_BRILL_GALENA CORPUS_TWO_form_tag
            CORPUS_THREE_form_tag.brill_galena > scores_brill_galena";
print "$command\n";
`$command`;

##########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Tagging CORPUS_TWO_form with LEXICON_BRILL_ITU\n";

$command = "$BRILLORIGINALDIR/Utilities/combine-lexicons.prl LEXICON_BRILL
            $NLPDIR/data/LEXICON_ITU > LEXICON_BRILL_ITU";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$BRILLORIGINALDIR/Bin_and_Data/tagger LEXICON_BRILL_ITU
            CORPUS_TWO_form BIGBIGRAMLIST LEXRULEOUTFILE CONTEXT-RULEFILE -s 1000 >
            CORPUS_THREE_form_tag.brill_itu";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$NLPDIR/bin/scores.prl LEXICON_BRILL_ITU CORPUS_TWO_form_tag
            CORPUS_THREE_form_tag.brill_itu > scores_brill_itu";
print "$command\n";
`$command`;

##########################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Tagging CORPUS_TWO_form with LEXICON_BRILL_ITU_GALENA\n";
```

```
$command = "$BRILLORIGINALDIR/Utilities/combine-lexicons.prl LEXICON_BRILL
            $NLPDIR/data/LEXICON_ITU_GALENA > LEXICON_BRILL_ITU_GALENA";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$BRILLORIGINALDIR/Bin_and_Data/tagger LEXICON_BRILL_ITU_GALENA
            CORPUS_TWO_form BIGBIGRAMLIST LEXRULEOUTFILE CONTEXT-RULEFILE -s 1000 >
            CORPUS_THREE_form_tag.brill_itu_galena";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$NLPDIR/bin/scores.prl LEXICON_BRILL_GALENA CORPUS_TWO_form_tag
            CORPUS_THREE_form_tag.brill_itu_galena > scores_brill_itu_galena";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f CORPUS_THREE_form_tag";
print "$command\n";
`$command`;

##############################################################################
$step++; print "\n*** "; print_time (); print " STEP $step: ";
print "Calculating word transitions\n";

print "\n    "; print_time (); print "\n";
$command = "$NLPDIR/bin/word_transitions.prl LEXICON_BRILL CORPUS_TWO_form_tag
            CORPUS_THREE_form_tag.brill > wt1";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "$NLPDIR/bin/word_transitions.prl LEXICON_BRILL_ITU
            CORPUS_TWO_form_tag CORPUS_THREE_form_tag.brill_itu > wt2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "paste wt1 wt2 | sort | uniq -c >
            word_transitions_from_LEXICON_BRILL_to_LEXICON_BRILL_ITU";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f d2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
```

```
$command = "$NLPDIR/bin/word_transitions.prl LEXICON_BRILL_ITU_GALENA
            CORPUS_TWO_form_tag CORPUS_THREE_form_tag.brill_itu_galena > wt2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "paste wt1 wt2 | sort | uniq -c >
            word_transitions_from_LEXICON_BRILL_to_LEXICON_BRILL_ITU_GALENA";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f wt1 wt2";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f CORPUS_THREE_form_tag.brill";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f CORPUS_THREE_form_tag.brill_galena";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f LEXICON_BRILL_GALENA";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f CORPUS_THREE_form_tag.brill_itu";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f LEXICON_BRILL_ITU";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f CORPUS_THREE_form_tag.brill_itu_galena";
print "$command\n";
`$command`;

print "\n    "; print_time (); print "\n";
$command = "rm -f LEXICON_BRILL_ITU_GALENA";
print "$command\n";
`$command`;

##############################################################################
```

```
print "\n*** "; print_time (); print " END\n";
```

D.13 scores.prl

```
#!/usr/bin/perl
#
# scores.prl
#
#    We assume <corpus-1> and <corpus-2> two corpora with lines
#
#       form1/tag1 form2/tag2 ... formn/tagn ./Q.
#
#       (lines could also end with ?/Q? or .../Q... or another
#        punctuation mark, and thus each line is a sentence)
#
#    and <lex> a lexicon with lines
#
#       form tag1 tag2 ... tagn
#
#    This program uses <corpus-1> as a reference corpus, uses <corpus-2>
#    as untagged(<corpus-1>) retagged with the last version of the tagger,
#    uses <lex> as the lexicon used by the tagger in that retagging process,
#    and provides scores of success and failure in the disambiguation process.
#
#    Scores calculus:
#
#       Notation:
#          + means success  (we use s in the program)
#          - means failure  (we use f in the program)
#          # means cardinal (we use c in the program)
#
#                                                              \
#                                              \               |
#       Out of vocabulary forms:                |              |
#          OOVF+                                | => #OOVF     |
#          OOVF-                                |              |
#                                              /               |
#                                                              |
#       In vocabulary forms:                                   |
#                                              \               |
#          Non-ambiguous forms:                 |              |
#             NAF+                              | => #NAF      | => #F
#             NAF-                              |              |
#                                              /               |
#                                                              |
#                                              \               |
#          Ambiguous forms:                     |              |
#             AF+                               | => #AF       |
#             AF-                               |              |
#                                              /               |
#                                                             /
```

```
#
#       Rates:
#
#                (OOVF+) + (NAF+) + (AF+)
#          S1 = ------------------------
#                            #F
#
#
#                (OOVF+) + (AF+)
#          S2 = ---------------
#                   #F - #NAF
#
#
#    Usage:
#
#       scores.prl <lex> <corpus-1> <corpus-2>
#

($#ARGV + 1 == 3) or die "\nUsage: $0 <lex> <corpus-1> <corpus-2>\n\n";
open (LL, "$ARGV[0]") or die "Can't open $ARGV[0]: $!\n";
open (XX, "$ARGV[1]") or die "Can't open $ARGV[1]: $!\n";
open (YY, "$ARGV[2]") or die "Can't open $ARGV[2]: $!\n";

$OOVFs = 0;  $NAFs = 0;  $AFs = 0;    $Fc = 0;
$OOVFf = 0;  $NAFf = 0;  $AFf = 0;
$OOVFc = 0;  $NAFc = 0;  $AFc = 0;

print STDERR "Loading lexicon...\n";
while ($lineLL = <LL>) {
    @fts = split (" ", $lineLL);
    $lexicon{$fts[0]} = $#fts;
}

print STDERR "Calculating scores...\n";
$line = 1;
while ($lineXX = <XX>) {
    $lineYY = <YY>;
    @wordsXX = split(" ", $lineXX);
    @wordsYY = split(" ", $lineYY);

    foreach $i (0..$#wordsXX) {
        ($formXX, $tagXX) = split ("/", $wordsXX[$i]);
        ($formYY, $tagYY) = split ("/", $wordsYY[$i]);
        if ($formXX eq $formYY) {
            if ($lexicon{$formXX} eq "") {
                if ($tagXX eq $tagYY) {
                    $OOVFs++;
                } else {
                    $OOVFf++;
                }
            } else {
                if ($tagXX eq $tagYY) {
```

```
                    if ($lexicon{$formXX} == 1) {
                        $NAFs++;
                    } else {
                        $AFs++;
                    }
                } else {
                    if ($lexicon{$formXX} == 1) {
                        $NAFf++;
                    } else {
                        $AFf++;
                    }
                }
            }
        } else {
            print STDERR "*** error in line $line, word $i:\n";
            print STDERR "    forms \"$formXX\" and \"$formYY\" do not match\n";
        }
    }
    $line++;
}

$OOVFc = $OOVFs + $OOVFf;
$NAFc  = $NAFs  + $NAFf;
$AFc   = $AFs   + $AFf;
$Fc    = $OOVFc + $NAFc + $AFc;

if ($OOVFc == 0) { $OOVFsp = "-"; } else { $OOVFsp = $OOVFs * 100 / $OOVFc; }
if ($OOVFc == 0) { $OOVFfp = "-"; } else { $OOVFfp = $OOVFf * 100 / $OOVFc; }
$NAFsp = $NAFs * 100 / $NAFc;
$NAFfp = $NAFf * 100 / $NAFc;
$AFsp = $AFs * 100 / $AFc;
$AFfp = $AFf * 100 / $AFc;

$S1 = ($OOVFs + $NAFs + $AFs) * 100 / $Fc;
$S2 = ($OOVFs + $AFs) * 100 / ($Fc - $NAFc);

print "OOVF+: $OOVFs ($OOVFsp)\n";
print "OOVF-: $OOVFf ($OOVFfp)\n";
print "NAF+: $NAFs ($NAFsp)\n";
print "NAF-: $NAFf ($NAFfp)\n";
print "AF+: $AFs ($AFsp)\n";
print "AF-: $AFf ($AFfp)\n";

print "S1 = $S1\n";
print "S2 = $S2\n";

print STDERR "Done...\n";
close(LL);
close(XX);
close(YY);
```

D.14 word_transitions.prl

```
#!/usr/bin/perl
#
# word_transitions.prl
#
#    We assume <corpus-1>, <corpus-2> and <corpus-r> three corpora with lines
#
#       form1/tag1 form2/tag2 ... formn/tagn ./Q.
#
#       (lines could also end with ?/Q? or .../Q... or another
#        punctuation mark, and thus each line is a sentence)
#
#    and <lex-1> and <lex-2> two lexica with lines
#
#       form tag1 tag2 ... tagn
#
#    This program uses <corpus-r> as a reference corpus,
#    uses <corpus-1> as untagged(<corpus-r>) retagged using <lex-1>,
#    uses <corpus-2> as untagged(<corpus-r>) retagged using <lex-2>,
#    and provides scores for word transitions among all the differents
#    counters, from the first test to the second one.
#
#    Example:
#
#       103953 AF+     AF+
#          319 AF+     AF-
#          523 AF-     AF+
#         6478 AF-     AF-
#        30633 NAF+    AF+
#          225 NAF+    AF-
#       138706 NAF+    NAF+
#          360 NAF-    AF+
#         2020 NAF-    AF-
#         2329 OOVF+   AF+
#          243 OOVF+   AF-
#        20644 OOVF+   NAF+
#         3491 OOVF-   AF+
#          868 OOVF-   AF-
#        10445 OOVF-   NAF+
#
#    Usage:
#
#       word_transitions.prl <lex-1> <corpus-1> > <wt1>
#       word_transitions.prl <lex-2> <corpus-2> > <wt2>
#       paste <wt1> <wt2> | sort | uniq -c
#       rm -f <wt1> <wt2>
#

($#ARGV + 1 == 3) or die "\nUsage: $0 <lex> <corpus-1> <corpus-2>\n\n";
open (LL, "$ARGV[0]") or die "Can't open $ARGV[0]: $!\n";
open (XX, "$ARGV[1]") or die "Can't open $ARGV[1]: $!\n";
```

```
open (YY, "$ARGV[2]") or die "Can't open $ARGV[2]: $!\n";

print STDERR "Loading lexicon...\n";
while ($lineLL = <LL>) {
    @fts = split (" ", $lineLL);
    $lexicon{$fts[0]} = $#fts;
}

print STDERR "Calculating scores...\n";
$line = 1;
while ($lineXX = <XX>) {
    $lineYY = <YY>;
    @wordsXX = split(" ", $lineXX);
    @wordsYY = split(" ", $lineYY);

    foreach $i (0..$#wordsXX) {
        ($formXX, $tagXX) = split ("/", $wordsXX[$i]);
        ($formYY, $tagYY) = split ("/", $wordsYY[$i]);
        if ($formXX eq $formYY) {
            if ($lexicon{$formXX} eq "") {
                if ($tagXX eq $tagYY) {
                    print "OOVF+\n";
                } else {
                    print "OOVF-\n";
                }
            } else {
                if ($tagXX eq $tagYY) {
                    if ($lexicon{$formXX} == 1) {
                        print "NAF+\n";
                    } else {
                        print "AF+\n";
                    }
                } else {
                    if ($lexicon{$formXX} == 1) {
                        print "NAF-\n";
                    } else {
                        print "AF-\n";
                    }
                }
            }
        } else {
            print STDERR "*** error in line $line, word $i:\n";
            print STDERR "    forms \"$formXX\" and \"$formYY\" do not match\n";
        }
    }
    $line++;
}

print STDERR "Done...\n";
close(LL);
close(XX);
close(YY);
```