Disambiguation of case suffixes in Basque

Victor Lascurain (1), Eneko Agirre (1), Mikel Lersundi (1)
Luboš Popelínský (2)
(1) University of the Basque Country, Donostia, Spain
`Email: bittor@web.de, eneko@si.ehu.es,
jialeaym@si.ehu.es`
(2) Faculty of Informatics, Masaryk University
Botanická 68, CZ-602 00 Brno, Czech Republic
`Email: popel@fi.muni.cz`

Mots-clefs – Keywords

désambiguisation morphologique, désambiguisation de la sens du mot, apprentissage inductive

morphological disambiguation, word-sense disambiguation, inductive learning

Résumé - Abstract

Le but de ce projet étais la classification automatique des cas grammaticaux dans la langue Basque. Pour réaliser ça on a appliqué l'apprentissage inductive (les systémes Tilde et Timbl). On emploie WordNet pour retrouver des mots et les hyperonyma des mots dans un contexte. L'exactitude etais plus haut que 70% pour Tilde et 63% en cas du Timbl.

The goal of this paper is to build a tool for automatic classification of grammatical cases in Basque. To achieve this goal we applied inductive learning techniques, namely systems Tilde and Timbl. We use WordNet for finding synsets and hyperonyms of words in a context. For Tilde we reached accuracy higher than 70% and for Timbl 63%.

1 Introduction

The goal of this paper is to build a tool for the automatic classification of case-suffixes in Basque. Basque is an agglutinative language and its case suffixes are more or less equivalent to prepositions, but they are also used to mark subject and objects of verbs. If we want to disambiguate the relation between the verb and the prepositional phrase it is really important to know the possibles interpretations of the Basque case-suffixes, as it is important in other languages to know the possibles interpretations of prepositions. This can be considered as a semantic disambiguation task.

All case-suffixes used in Basque need to be analysed. Some of them are more ambiguous than the others as it happens with prepositions in other languages. We have chosen *instrumental* because it is one of the most ambiguous, and more interesting from a disambiguation point of view. To clarify the task, Table 1 shows the possible interpretations of the instrumental case-suffix with examples in Basque and their translation into English.

Table 1: Possible interpretations of the instrumental case-suffix (-z).

	Basque	**English**
theme	Seguru nago horretaz Matematikaz asko daki	I'm sure of that He's an expert in maths
during-time	Arratsaldez lasai egon nahi dut Gauez egin dut	I like to relax of an evening I did it by night
instrument	Autobusez etorri naiz Belarra segaz moztu Euskaraz hitz egin	I have come by bus To cut grass with a scythe To speak in Basque
manner	Animali baten hestea betez egindako haragia Ahots ozen batez	A meat preparation made by filling an animal intestine In a loud voice
cause	Haren aitzakiez nekatuta nago Beldurrez zurbildu Kanpoan lan egitea baztertu zuenez, lan-aukera ederra galdu zuen	Sick of his excuses To turn white of fear In refusing to work abroad, she missed an excellent job opportunity
containing	Edalontzia ardoz beteta dago Txapelaz dagoen gizona Ilez estalia	The glass is full of wine The man with the beret on Cover in hair
matter	Armairua egurrez egina dago	The wardrobe is made of wood

The goal is then to classify each occurrence of the case suffix into one of the possible interpretations, – *theme*, *place*, *instrument*, etc. – taken into account the context. The approach we have used is based on learning from a set of hand-tagged occurrences of the instrumental case suffix, using inductive logic programming (Muggelton, 1992; Muggleton & De Raedt, 1994) (ILP) and instance-based learning (Zavřel & Daelmans, 1998) techniques. In order to test each of the approaches we have used 5-cross validation.

The context of each occurrence is annotated with ambiguous morphological tags. It means that for all context words we know all morphological readings but we do not know the right one. Besides WordNet[1] is used to generalise the words in the context. WordNet is a lexical reference system that organise English nouns, verbs, adjectives and adverbs into synonym sets, each representing one underlying lexical concept.

The structure of this paper is following. In Section 2 we describe the data used for learning. Section 3 contains brief information on WordNet. Experiments with the ILP system Tilde are described in Section 4. In Section 5 we bring results obtained with the instance-based learner Timbl. We conclude with overview of relevant works and with concluding remarks.

[1] http://www.cogsci.princeton.edu/˜wn/

2 Data

The learning database contained 142 correctly classified examples of the target relation. These examples have been extracted from a monolingual Basque dictionary (Sarasola, 1996). Each example is a sentence containing a word inflected in the instrumental case. Each word in the sentence has been ambiguously morphologically tagged.

One example in a raw format shown in Figure 1 represents the sentence "Bazkaz hornitu.". Each word of the sentence is enclosed in "<>" and followed by a list of possible readings. The first word, "<Bazkaz>" has two possible readings. In each reading the first part is the lemma ("bazka" *pasture* or *grass* in English) followed by a list of tags. We exploit here only the morphological ones. For this word the interesting tags are "IZE" (noun), "DEK" (declined word) and "INS" (instrumental case). The second word ("hornitu", *feed* in English) has three possible readings. In this case we can see that the lemma is always followed by the tag "ADI" (verb), so this word has only verb readings.

Figure 1: An example of a sentence in a raw format.

```
"<Bazkaz>"
      "bazka"  IZE ARR DEK INS MG
      "bazka"  IZE ARR DEK INS NUMS MUGM
"<hornitu>"
      "hornitu"  ADI SIN AMM PART ASP BURU  NOTDEK
      "hornitu"  ADI SIN AMM PART DEK ABS MG
      "hornitu"  ADI SIN AMM PART  NOTDEK
"<$.>$"
      PUNT_PUNT
```

The data in this format was further transformed into the form of Prolog facts. Each example consists of three predicates, *position/1* (the word carrying the instrumental case), *leftCtx/1* (the list of words in the left context, in the reverse order, together with their morphological readings), and *rightCtx/1* (the list of words in the right context with their morphological readings). The transformed data can be seen in Figure 2. Each example in the learning set has been manually classified into one of seven different semantic categories. Their frequency is displayed in Table 2.

Figure 2: Sentence from the database. Prolog format.

```
begin(model(example1)). theme.
leftCtx([]).
rightCtx([word(hornitu,
        [[hornitu,adi,sin,amm,part,asp,buru,notdek],
        [hornitu,adi,sin,amm,part,dek,abs,mg],
        [hornitu,adi,sin,amm,part,notdek]])]).
position(word(bazkaz,
        [[bazka,ize,arr,dek,ins,mg,aorg,has_mai,def_hasi,notgelgen],
        [bazka,ize,arr,dek,ins,nums,mugm,aorg,has_mai,def_hasi,
          notgelgen]])).
end(model(example1)).
```

3 WordNet

The most important information for our task is the meaning of the word present in the relation the case suffix represents, usually a noun and a verb. As it is impossible to list every single word

Table 2: List of semantic categories and corresponding frequencies.

Class:	cause	containing	instrument	manner	matter	theme	time
Number:	5	23	31	41	7	29	6
Frequency:	0.03	0.16	0.21	0.28	0.05	0.20	0.04

pair that can be related to a given preposition or case, some way of generalisation from words to more abstract concepts will be useful. For this crucial task we exploited information from WordNet. The WordNet is a net made of words, or more exactly, synsets. A synset is a named collection of words that share a common meaning. Synsets in the net are related to each other in many ways. Regarding our work the *hyperonymy/hyponymy* relation is the most important. This relation defines a sub net inside the WordNet, which links synsets regarding to a *is-a* relation. This relation enables generalisation from specific words to more general concepts.

4 Learning with Tilde

Inductive logic programming (ILP) (Muggelton, 1992; Muggleton & De Raedt, 1994) is a machine learning technique that learns first order logic descriptions from a set of examples and a given background knowledge (Muggelton, 1992; Muggleton & De Raedt, 1994). We used the *Tilde* system which learns first order logic decision trees (Blockeel & Raedt, 1997).

Good background knowledge expressed in the form of a logic program is crucial for a good performance of any ILP system. We tested several different types of background knowledge predicates. A description of the characteristics of each of them as well as the obtained results are presented in the next paragraphs.

4.1 Morphological predicates

The first set of predicates is composed of only simple morphological predicates. There are two different types of predicate, *exists* and *forall* predicates. The "*exists/1*" predicate checks whether a given morphological tag is present in at least one of the readings of one of the words in the example. The "*forall/1*" predicate checks whether a given morphological tag is present in all the readings of at least one word in the example. The accuracy of this classifier is around 47%. All the results have been obtained by running 5–cross validation.

A second experiment was done with a slightly modified set of predicates. Namely "*exists/2*" and "*forall/2*" predicates were added. They do the same checks as their arity 1 equivalents but for a pair of tags instead for a single one. The accuracy increased to 55%.

4.2 Semantic (WordNet) predicates

In order to increase accuracy semantic information from WordNet has been introduced into the background knowledge. This semantic information was used in both possible ways, either alone, or in combination with the morphological information. The new predicates have the form *hasSynset(X)* and *hasHyperonym(X)* till 3rd level up in the hyperonymy hierarchy. The predicate *has_synset(Synset)* succeeds if a word, member of the given *Synset*, is present in the sentence. The predicate *has_Hyperonym(Hyperonym)* succeeds if a word belongs to the *Hyperonym*. For example, given the sentence "*Let's dance the war dance*" the predicate "*hasHyperonym(ASynset)*" would succeed for *ASynset = synset of ritual dance* but not for *ASynset = synset of social dancing*. In order to improve accuracy and to decrease learning time further improvement has been performed based on the following observations:

- The word in instrumental case is usually a noun or an associated determinant. The noun to which the determinant is associated is usually the first noun to the left from this determinant. The determinant does not modify the classification.
- When the word in instrumental case is a noun (or determinant) it defines a relation between the noun and the nearest verb to the right.

Accuracy of finding the most significant words can be seen in Table 3. Then the semantic predicates are applied only to these important words.

Table 3: Finding the most significant words.

Type	Hit	Fail	Unknown	Total	Accuracy
Verb:	108	5	27	140	77.4
Noun:	105	7	0	112	93.8

There is a description of the new predicates.

- *nearestNounNotVerbNotDet/1*: looks for a word with at least one noun reading and which has no determinant neither verb readings. It first looks in the position, then in the left context and then in the right context, returning the first word found. For example, in the sentence *Etxe (house) batez (of a) jabetu (become the owner)* the goal *nearestNounNotVerbNotDet(Word)* success only for *Word = house*.
- *nearestVerbNotDet/1*: looks for a word with at least one verb reading and which has no determinant readings. It first looks in the position, then in the right context and then in the left context, returning the first word found. Using the same sentence as above as an example the predicate *nearestVerbNotDet(Word)* success only for *Word = jabetu*.

The following combinations of the semantic predicates and the morphological predicates were used:

- In all cases only synsets and the first level in the hyperonymy hierarchy are used, i.e. only *nearestVerNotDetHasHyperonym/1*, *nearestNounNotVerbNotDetHasHyperonym/1*, *nearestVerbNotDetHasSynset/1* and *nearestVervNotDetHasHyperHyperonym/1* predicates are provided as background knowledge.
- In the first case *exists/1* and *forall/1* are used. The accuracy in this case is 56 %.

- In the second case only *exists/2* is used. This reduction is due to the available machine resources. In this case the accuracy is 59 %.

It demonstrates that the WordNet predicates do provide some valuable information, even when applied to ambiguously tagged text.

4.3 Refinement of the semantic predicates

When introducing the WordNet predicates a new source of ambiguity has been introduced. The words in the context are not semantically disambiguated and for that reason we can not remove any synset. Because of that we have to add all the possible semantic interpretations that a word can bear. In this section we explain how we tried to overcome this problem.

The problem cannot be completely solved without manual disambiguation. However, some improvements can be done if using the frequency of a given synset as measure for it “goodness”. We make the assumption that if two different words have a common synset (or hyperonym) it is more likely that this synset(hyperonym) is the right one. For example, given the sentences “sitting on the chair” and “sitting on the bank” we assume that the correct interpretation of “bank” is that of “chair” and not that of “credit institution”. We implement this in our application by removing all the synsets which appear less than N times in the database.

If to compare with the results displayed in the previous paragraph, in the first case the accuracy 57% while in the second it is about 60%. So there is slight improvement about 1 %. It is important that also the learning time was a bit smaller. It can be important when processing bigger data.

4.4 Leaving implicit class

The result of the last experiment is in Table 4. We can see that the biggest discrepancy between expected classification and the learned one concerns the class `instrument`. When we have a look to a typical result of learning (below) we can see that the default category (i.e. category that is used if no rule fires for the classified example) is again `instrument`. An example of output of Tilde is below.

```
class([time]):-nearestNounNotVerbNotDetHasHyperonym(s09065837),!.
% 6.0/6.0=1.0.
class([theme]):-nearestVerbNotDetHasSynset(s00527673),!.
% 13.0/13.0=1.0.
...
class([instrument]).
% 12.0/22=0.545
```

So we decided to remove the last clause from the learned rules. On one side it results in decrease of recall, in other side accuracy increased up to 10%. Namely for the two cases mentioned the accuracy increased up to 70.4% (recall 69.0) and 71.1% (recall 68.3).

Table 4: Results with arity 2 morphological predicates and refined WordNet.

REAL / PRED	cause	containing	instrument	manner	matter	theme	time	total
cause	**0**	0	1	1	0	3	0	5
containing	0	**21**	2	0	0	0	0	23
instrument	0	3	**7**	15	1	3	1	30
manner	0	1	5	**29**	1	5	0	41
matter	0	3	1	1	**1**	1	0	7
theme	1	1	1	4	0	**23**	0	30
time	0	0	0	0	0	1	**5**	6
total	1	29	17	50	3	36	6	142

5 Learning with Timbl

Timbl [2] (Zavřel & Daelmans, 1998) is a program implementing several instance-based, or Memory-Based, learning techniques. Timbl stores a representation of the training set explicitly in memory, and classifies new cases by extrapolation from the most similar stored cases.

5.1 Learning data

The propositional representation available for ILP had to be re-coded into the format required for Timbl. First, morphological information has been removed and the WordNet predicates have been rebuilt. In Timbl, each example is seen as a chain of comma separated items. So for each word in a given sentence, one of its lemmas is randomly chosen and the word/lemma pairs are written in a comma separated list of a given length. The *word in position* is always on the 5^{th} position of the chain. As it may happen, that not all the examples are long enough, the missing ones are filled with underline characters. The category comes after the chain followed by a dot. An example is shown in Figure 3.

Figure 3: Two sentences in the Timbl format.

```
_,_,_,_,_,_,_,_,zauriz,zauri,betea,bete,_,_,_,_,_,_,containing.
_,_,_,_,_,_,_,_,gauaz,gau,zaintzen,zaindu,zizkiena,edun,_,_,_,_,time.
```

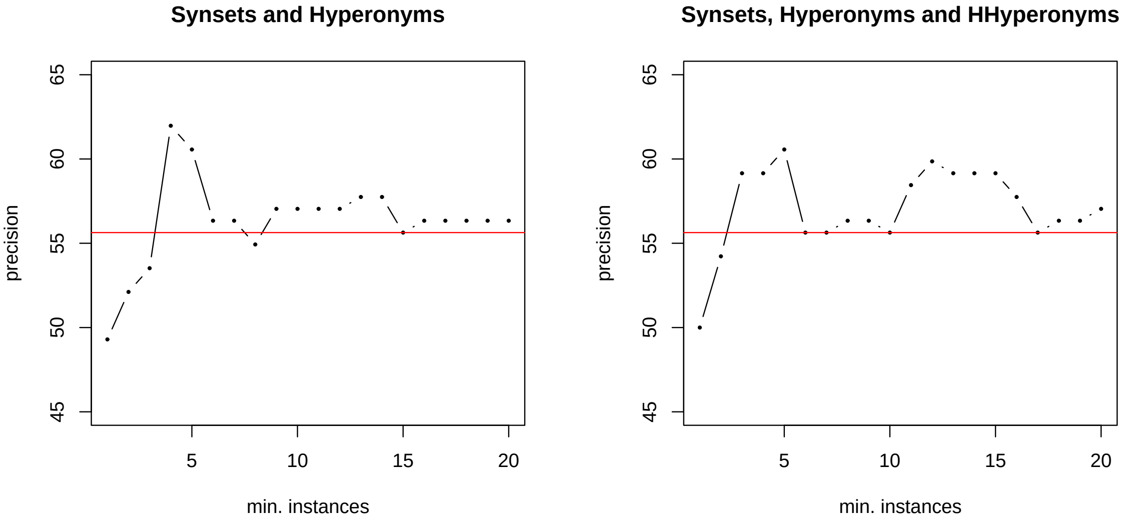
For each column pair (word/lemma) the union of the synsets of each line is found. If this set contains N elements, then N extra binary attributes are added to the database. For a given example the value of one of these binary attributes is true if and only if the synset it represents is a synset of the word in the given column.

5.2 Results

Two different set of experiments have been performed. They differ in the depth employed in the hyperonymy tree. In the first case only synsets and hyperonyms are considered. In the second

[2] http://ilk.kub.nl/software.html

Figure 4: Results for Timbl



one hyper-hyperonyms have also been used. Timbl was learned five times. In the first time the database is used without modification. In the next ones the synsets/hyperonyms which are true (following the schema above) for less than N examples are removed, for $N \in \{1..20\}$. The results can be seen in Figure 4. In both graphs, the horizontal line in the middle shows accuracy 55.6%, the case when no WordNet information has been exploited.

6 Related work

Agirre et al. (Agirre *et al.*, 2002) presented preliminary experiments in the use of translation equivalences to disambiguate prepositions or case suffixes. The core of the method is to find translations of the occurrence of the target preposition or case suffix, and assign the intersection of their set of interpretations. Given a table with prepositions and their possible interpretations, the method is fully automatic. The method was tested on the occurrences of the Basque instrumental case -z in the definitions of a Basque dictionary, looking for the translations in the definitions from 3 Spanish and 3 English dictionaries. The method is able to disambiguate with 94.5% accuracy 2.3% of those occurrences (up to 91). The ambiguity is reduced from 7 readings down to 3.1.

There has been many works that apply ILP for morphological disambiguation. Cussens (Cussens, 1997) developed POS tagger for English that achieved per-word accuracy of 96.4 %. Eineborg and Lindberg induced constraint grammar-like disambiguation rules for Swedish with the accuracy of 98%. In (Džeroski & Erjavec, 1997) ILP was applied for generating the lemma from the oblique form of nouns as well as for generating the correct oblique form from the lemma, with the average accuracy 91.5 % . Learning nominal inflections for Czech and Slovene (among others) is described in (Manandhar *et al.*, 1998). In (Cussens *et al.*, 1999), first steps in morphosyntactic tagging of Slovene are described. The obtained accuracy 86.6% is comparable with our results of tag disambiguation that varied between 80% and 98%. In (Nepil *et al.*, 2001; Žáčková & Popelínský, 2000) we brought first results for morphological tagging in Czech with means of ILP. We did not employ any lexical statistics and we did not use any hand-crafted domain knowledge.

7 Conclusion

The results are not good enough for automatic disambiguation of cases in Basque. However, some conclusions can already be made. When using only simple morphological predicates an accuracy varied between 47% and 55%. When we introduce semantic predicates they produce a small improvement 59%. We can improve these results a little bit more by refining the semantic predicates trying to remove ambiguity as described in Section 4.3. When removing the implicit rule we reached accuracy higher than 70% with decrease of recall to 68–69%.

This fact seems to confirm that the information derived from the WordNet is important but does not mean that the morphological information should be automatically discarded. In the experiments described in Section 4.2 the morphological information is important for finding the so called ”important” words (the words for which the WordNet predicates are applied). The morphological information is also used in *verbInPosition* and *adverbInPosition* predicates. Nevertheless we got the best results making no use of the morphological knowledge in the experiments described in Section 5.

Regarding to the experiments with Timbl, trying to find an explanation for the particular form the two curves show in Figure 4 is interesting. When we add the WordNet information the accuracy falls down about 6% in both cases and then it increases steadily to meet its peak value for $N \approx 5$. At this point the tendency changes and accuracy becomes worse. There is a possible explanation. When we added the synset information to the data we also add a lot of noise. By adding all the synsets of a word the only thing we do is adding all the possible semantic interpretations of a given word. When we restrict the minimum number of examples in which a synset must be present (the N parameter) data become less ambiguous. Those synsets which belong to different words have better chances to survive. The accuracy increases until we begin to destroy more information than noise. As N moves from 1 to 20 there is a balance between noise and information. The first peak could be due to the situation in which the synset and hyperonym information weigh more than the ambiguity they introduced. From that moment we begin to destroy information, so the curve sinks. The second peak belongs to the hyper-hyperonyms, which should be more common and thus are removed later. When this happens the second peak collapses.

Acknowledgements

Most of this work was done when Victor Lascurain and Luboš Popelínský worked with machine learning group at University of Freiburg, Germany. We would like to thank to Luc De Raedt and Stefan Kramer for their kind assistance. Luboš Popelínský has been partially supported by the grant of Czech Ministry of Education MŠMT 143300003. Eneko Agirre and Mikel Lersundi have been partially supported by the European Commission (MEANING project IST-2001-34460) and MCYT (Hermes project TIC-2000-0335-C03-03).

References

AGIRRE E., LERSUNDI M. & MARTÍNEZ D. (2002). A multilingual approach to disambiguate prepositions and case suffixes. In *ACL Workshop: Word Sense Disanbiguation: recent successes and future directions*.

BLOCKEEL H. & RAEDT L. D. (1997). *Top-down Induction of Logical Decision Trees*. Rapport interne, Ktholieke Universiteit Leuven. Department of Computer Science.

CUSSENS J. (1997). Part-of-speech tagging using Progol. In *Inductive Logic Programming: Proceedings of the 7th International Workshop (ILP-97). LNAI 1297*, p. 93–108: Springer.

CUSSENS J., DŽEROSKI S. & ERJAVEC T. (1999). Morphosyntactic tagging of Slovene using Progol. In S. DŽEROSKI & P. FLACH, Eds., *Inductive Logic Programming: Proc. of the 9th International Workshop (ILP-99)*, Bled, Slovenia: Springer-Verlag.

CUSSENS J. & (EDS.) S. D. (2000). *Learning language in Logic*. Springer.

DŽEROSKI S. & ERJAVEC T. (1997). Induction of Slovene nominal paradigms. In *Inductive Logic Programming: Proceedings of the 7th International Workshop (ILP-97). LNAI 1297*, p. 141–148: Springer.

LAVRAČ N. & DŽEROSKI S. (1994). *Inductive Logic Programming: Tecniques and Applications*. Ellis Horwood.

MANANDHAR S., DŽEROSKI S. & ERJAVEC T. (1998). Learning multilingual morphology with CLOG. In *Inductive Logic Programming: Proceedings of the 8th International Conference (ILP-98)*: Springer.

MUGGELTON S. (1992). *Inductive Logic Programming*. Academic Press.

MUGGLETON S. & DE RAEDT L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, **19/20**, 629–679.

NEPIL M., POPELIINSKY L. & ŽÁČKOVÁ E. (2001). Part-of-speech tagging by means of shallow parsing, ilp and active learning. In *Proceedings of the Third Learning Language in Logic (LLL) Workshop, Strasbourg, France*.

NIENHUYS-CHENG S.-H. & DE WOLF R. (1997). *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

SARASOLA I. (1996). *Euskal Hiztegia*. Gipuzkoako Kutxa.

ŽÁČKOVÁ E. & POPELÍNSKÝ L. (2000). Automatic tagging of compound verb groups in Czech corpora. In *Text, Speech and Dialogue: Proceedings of TSD'2000 Workshop, LNAI*: Springer.

ZAVŘEL J. & DAELMANS W. (1998). *Recent Advances in Memory-Based Part-of-Speech Tagging*. Rapport interne, ILK/Computaional Linguistics, Tilburg University.