FI MU

Faculty of Informatics
Masaryk University

On Disambiguation in Czech Corpora

by

Luboš Popelínský
Tomáš Pavelek
Tomáš Ptáčník

FI MU Report Series **FIMU-RS-2000-07**

 October 2000

On Disambiguation in Czech Corpora

Luboš Popelínský, Tomáš Pavelek, Tomáš Ptáčník

Natural Language Processing Laboratory
Faculty of Informatics, Masaryk University in Brno
Email: {popel,xpavelek,ptacnik}@fi.muni.cz

Abstract

Lemma disambiguation means finding the basic word form, typically nominative singular for nouns or infinitive for verbs. We developed a multistrategy method for lemma disambiguation of unannotated text. The method is based on a combination of inductive logic programming and instance-based learning. We present results of the most important subtasks of lemma disambiguation for Czech language. Although no expert knowledge on Czech grammar has been used the accuracy reaches 90% with a fraction of words remaining ambiguous. We also display first results of tag disambiguation[1].

1 Introduction

Disambiguation in inflective languages is a very challenging task because of its usefulness as well as its complexity. First of all, we focus on lemma disambiguation in Czech language. Lemma disambiguation means assigning the basic word form to each word form – nominative singular for nouns, adjectives, pronouns and numerals, infinitive for verbs. E.g. in the sentence *Od rána je Ivana se ženou.* (literally *since (the) morning Ivana (female) is with (my) wife.*) each word except the preposition "od" has two basic forms. E.g. *rána* can be the genitive of *ráno*(morning) as well as the nominative of

[1]This is a substantially extended version of our LLL'99 contribution [Popelínský et al., 2000]. A brief summary of lemma disambiguation results may be also found in [Popelínský et al., 1999]

substantive *ráno*(bang). In Czech corpora it was observed that 10% of word positions – i.e. each 10th word of a text – have at least 2 lemmata and about 1% word forms of Czech vocabulary has at least 2 lemmata.

In this paper we focus on the most frequent ambiguous word forms *se* (reflexive pronoun or preposition) and *je* (*is* or *them*). In Tab. 1 we can see their frequency compared with the most frequent Czech words in DESAM corpus [Pala et al., 1997]. Disambiguation of the word forms *se* and *je* would be welcome as they are 3rd and 5th most frequent words in DESAM corpus. We will show how to employ machine learning techniques for building reliable disambiguators. Our multistrategy approach combines inductive logic programming(ILP) [Muggleton and De Raedt, 1994] – rules learned with P-Progol[2] – with instance-based learning [Mitchell, 1997]. We will also show how to disambiguate words that did not appear in the corpus.

Table 1: Most frequent words in DESAM corpus

a	22542	je	9553
v	17364	že	7565
se	15933	s	6344
na	13671	o	6059

a = *and* in English, v = *in,at,on*, na = *on*, že = the conj. *that*, o = the prep. *about*

2 Overview of the Method

We now briefly describe our method using the example in Tab. 2. In the example the tag k1gNnSc2 of word *ráno* (morning) means: part of speech (k) = noun (1), gender (g) = neutral (N), number (n) = singular (S) and case (c) = genitive (2). Lemmata and possible tags are prefixed by `<l>`, `<t>` respectively. The correct tags are highlighted. We want to find the correct lemma of the word form *se*. The learning set is built from unambiguously tagged sentences taken from the DESAM corpus. We exploit only tags. For *se* the above sentence is transformed into

```
ex([k1gFnSc1,k5eAp3nStPmIaI,k1gNnSc2,k7c2],[k1gFnSc7])
```

[2]http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.html

Table 2: Lemma ambiguities in a Czech sentence

Od	<l>	od	<t>	**k7c2**
rána	<l>	ráno	<t>	**k1gNnSc2**,k1gNnPc145
	<l>	rna	<t>	k1gFnSc1
je	<l>	být	<t>	**k5eAp3nStPmIaI**
	<l>	on	<t>	k3xPgNnSc4p3,k3xPgXnPc4p3
Ivana	<l>	Ivan	<t>	k1gMnSc24
	<l>	Ivana	<t>	**k1gFnSc1**
se	<l>	s	<t>	**k7c7**
	<l>	sebe	<t>	k3xXnSc4
ženou	<l>	žena	<t>	**k1gFnSc7**
	<l>	hnát	<t>	k5eAp3nPtPmIaI

The first argument of `ex/2` is built from tags in the left context in the reverse order, the second one contains tags of the word in the right context. As *se* is the preposition in the given context, it is the positive example for lemma *s* (preposition) and the negative example for lemma *sebe* (pronoun).

Domain knowledge predicates have been built automatically and no user defined predicates have been introduced. Domain knowledge predicates have the following form `p(Context, Focus, Condition)`, where `Focus` defines a subset of `Context`. E.g. `p(Left, first(2), always([k5,eA]))` succeeds if in the first two tags of the left context `k5, eA` always appears. For a given word form P-Progol learns two sets of disambiguation rules, one for each of two lemmata. Several examples of the learned rule are below.

```
pronoun(Left,Right) :-
  p(Right,first(2),somewhere(k5)).
pronoun(Left,Right) :-
  p(Right,first(1),always(c1)).
pronoun(Left,Right) :-
  p(Right,first(1),always(k6)),
  p(Left,first(1),always([k5,aI,eA])).

preposition(Left,Right) :-
  p(Right,first(1),always([k3,c7])),p(Right,first(1),always(x0)).
```

```
preposition(Left,Right) :-
  p(Right,first(2),always(c7)),p(Right,first(1),always(nP)).
```

Left context has been first reverted. Then `first(1)` for the left context means the neighbour of the word (that is to be disambiguated). The first rule displays the most common situation when reflexive pronoun *se* is followed by a verb (k5) – actually the verb appeared somewhere in next two words. The opposite word order is described by the third rule: a verb immediately precedes the reflexive pronoun. In the second set of rules, `preposition/2`, the instrumental (c7) almost always appears in the third argument of `p/3` because the preposition *se* is in the Czech language associated always with this case.

Then we took a set of sentences not used in the learning step, that contains the given word form. Employing morphological analyzer LEMMA (Pala and Ševeček P. 1995) we found all possible tags for the words that appear in the context of the given word. E.g. for *se* in the sentence in Tab. 2 and the context of length 1 the morphological analyzer returns following tags

je	*Ivana*	*(se)*	*ženou*
k5eAp3nStPmIaI	k1gMnSc24		k1gFnSc7
k3xPgNnSc4p3	k1gFnSc1		k5eAp3nPtPmIaI
k3xPgXnPc4p3			

Afterwards we generate all combinations of those tags (Tab. 3). Now we employ the method based on the k-nearest neighbour(kNN) approach. Both two sets of rules that have been learned with P-Progol – one for the case of preposition, the other for reflexive pronoun – are applied to the set of `data/2` facts. Two success rates (the number of correctly covered positive examples plus the number of correctly uncovered negative examples divided by the number of all examples), one for each of the rule sets, are used as coordinates in kNN model. The incoming sentence that is to be disambiguated is tagged in the same way, the `data/2` facts are generated for it, and two success rates are again computed. The chosen lemma has to be much more frequent than the other in the close neighborhood of the disambiguated sentence.

The rest of the paper is organized as follows. Section 3 describes DESAM corpus that was used for learning. In Section 4 we first explain what domain knowledge was used (Section 4.1). Then we present the results obtained with Progol for the most frequent lemma-ambiguous word form *se* (Section 4.2).

Table 3: Examples for kNN

```
data( [k1gMnSc24,k5eAp3nStPmIaI] , [k1gFnSc7] ).
data( [k1gFnSc1,k5eAp3nStPmIaI]  , [k1gFnSc7] ).
data( [k1gMnSc24,k5eAp3nStPmIaI] , [k5eAp3nPtPmIaI] ).
data( [k1gFnSc1,k5eAp3nStPmIaI]  , [k5eAp3nPtPmIaI]).
data( [k1gMnSc24,k3xPgNnSc4p3]   , [k1gFnSc7] ).
data( [k1gFnSc1,k3xPgNnSc4p3]    , [k1gFnSc7] ).
data( [k1gMnSc24,k3xPgNnSc4p3]   , [k5eAp3nPtPmIaI] ).
data( [k1gFnSc1,k3xPgNnSc4p3]    , [k5eAp3nPtPmIaI]).
data( [k1gMnSc24,k3xPgXnPc4p3]   , [k1gFnSc7] ).
              ...   ...   ...
```

Section 5 brings the results of disambiguation when correct tags in a context are unknown. In Section 6 we display the first results of tag disambiguation of nouns. We then discuss the obtained results in Section 7 and conclude with a summary of relevant works in Section 8.

3 Data Source

DESAM [Pala et al., 1997], the corpus of Czech newspaper texts that is now being built at Faculty of Informatics, Masaryk University, contains more than 1 000 000 word positions, about 130 000 different word forms, about 65 000 of them occurring more then once, and 1665 different tags. Characteristics of DESAM are in Tab. 4. DESAM is now being tagged – partially manually, partially by means of different disambiguators – into 66 grammatical categories like a part-of-speech, gender, case, number etc., about 2 000 tags, combinations of category-value couples[3]. It was observed [Pala et al., 1997] that there is in average 4.21 possible tags per word.

For the first step – learning disambiguation rules with Progol – we used the part of DESAM that has been manually tagged (about 250 000 word po-

[3]E.g. for substantives, adjectives there are 4 basic grammatical categories. For pronouns 5 categories, for verbs 7 and for adverbs 3 categories, and some number of subcategories. The large number of tags is due to a combination of those categories.

Table 4: Characteristics of DESAM Czech corpus

Positions	1 247 594
Different word forms(tokens)	132 447
Word forms occurring once	67 059
Different lemmata	34 606
Lemmata occurring once	11 759
Different tags	1 665

sitions). This part of DESAM contains only a small fraction of incorrectly tagged words. The rest of DESAM has been exploited for the second step – kNN.

4 Learning Rules with Progol

4.1 Domain Knowledge

There is no complete formal description of Czech grammar. That is why any domain knowledge, even written by a linguist, is necessarily incomplete (does not cover all cases that have appeared in the corpus) or incorrect. Here we describe the systematic way of domain knowledge building without any need of linguistic knowledge. We only exploit the information about particular tags in a context. In the following text "word" means the word that is to be disambiguated. The general form of domain knowledge predicates is

```
p(Context, Focus, Condition)
```

where `Context` is a variable bound with either left context in a reverse order or with right context of the word, `Focus, Condition` are terms. `Focus` defines a subpart of the `Context`. `Condition` says what condition must hold on the `Focus`. `Focus` has a form

- `first(N)` (N=1..max_length) ... a sublist of the `Context` of length `N` neighboring with the word. `max_length` is a maximal length of a context.

`Condition` is an unary term whose argument is a list of tags

- `somewhere(List)` ... tags from the `List` appear somewhere in the `Context`;
- `always(List)` ... tags from the `List` appear in all positions in the `Context`.

E.g. a goal `p(X,first(4), somewhere([k1]))` succeeds if somewhere in the next four words of the context X there is a noun (k1). The goal `p(X,first(2), always([c7,nS]))` succeeds if tags `c7,nS` appear in each of the first two words in the context `X` – e.g. a pronoun and a noun in singular instrumental as in *s tvou sestrou – (with) your sister*. We also performed experiments with a broader class of predicates. E.g. predicates like `subset(Length)` – any subset of a context of the length `Length` – seemed to be promising. However, it does not significantly improve the accuracy.

The predicate `p(Context, Focus, Condition)` actually represents a class of predicates. Particular members of this class differ in focus predicates and conditions. It is important to notice that this class of predicates can be generated automatically from the set of grammatical categories and their values.

4.2 Results

We will demonstrate our method on disambiguation of the word form *se*. It may have either the lemma *s* (preposition like *with* in English) or the lemma *sebe* (reflexive pronoun *self*). In the manually tagged part of DESAM corpus there were together 3167 sentences with *se* (232 occurrences tagged as a preposition, 2935 ones tagged as a reflexive pronoun). 80% of examples were randomly chosen and used for learning. The left and right contexts have been set to 5 words. Untagged words in context have been tagged as 'unknown part-of-speech' (tag `kZ`). Negative examples were built from sentences where the word had the second lemma. Using P-Progol version 2.2 we have learned rules for both of the two lemmata. It means that for each task we obtained two rule sets that should be complementary. However, we have found it useful to use both of them. Results are in Tab. 5. The number of examples for each lemma is in the 2nd column. The average accuracy on the testing set from 5 runs is in the 3rd column. The learning time reached 14 hours. It is caused by the enormous number of 4536 literals that may be introduced in a rule body. It must be mentioned that the default accuracy, i.e. assigning

Table 5: Results of Progol

	#examples	accuracy(%)
preposition	232	94.48
pronoun	2935	92.84

the reflexive pronoun lemma to each occurrence of *se* is 92.7%. Then the rule accuracy 92.84 is not too impressive. We will show in the next section that even such "poor" rule set is usable for lemma disambiguation.

5 Lemma disambiguation

5.1 Data preparation

The learning and the testing sets contained sentences from DESAM that have not been used for learning the disambiguation rules. We removed all sentences with commas, dots, parentheses etc. This set contained 1635 sentences. All possible tags were found for each sentence employing LEMMA morphological analyzer. Then all variations of tags were generated for each sentence and the set of `data/2` facts was computed, as described in Section 2. We narrowed both left and right contexts to the length of 3 words to limit the number of *data/2* facts. E.g. for the right and left contexts of length 3, and for the average number of possible tags per word 4.21 we have obtained about $4.21^3 * 4.21^3 \doteq 5567$ combinations of tags.

5.2 Method

From the rule sets learned by Progol we took two rule sets, one for each lemma, that displayed the highest success rate. First we tried to use the rule sets learned with Progol directly. However, the enormous ratio of unresolved cases – more than 30% – made us look for another method that would decrease those ratios without significant decrease of accuracy. The method based on a combination of ILP and kNN is described below.
50% of 1635 sentences were used for estimation of the parameters of kNN model in the following way. Both set of rules learned with Progol were run on `data/2`. Thus for each sentence we had received two success rates, i.e.

Table 6: kNN algorithm

1. Generate the set of instances. Count the number of cases that fall under each point.

2. For a sentence that is to be disambiguated generate `data/2`.

3. For `data/2` compute two success rate $x^{'}, y^{'}$, one for each set of rules.

4. Find the nearest neighbor of $(x^{'}, y^{'})$ in the learning set. Let n_1, n_2 be the numbers of cases corresponding to the first and the second lemma felt into this point.

5. lemma := **if** $n_1 > n_2 \ \wedge x_1^{'} > t_1$ **then** $lemma_1$ **else**
   if $n_1 < n_2 \ \wedge x_2^{'} > t_2$ **then** $lemma_2$
   else $unresolved$

(x_1, x_2) coordinates in a two-dimensional space. For each point (x_1, x_2) we computed the number of sentences for each lemma that "fall" under that point. Let n_1 be the number of sentences with the first lemma, n_2 the number of sentences with the second for this point. For a sentence that was to be disambiguated we again generated the set of `data/2` and we computed two success rates $x^{'}, y^{'}$. Then we found the nearest neighbor (x, y) to the point $x^{'}, y^{'}$ employing the Euclidean distance. If for point (x, y) e.g. n_1 was greater than n_2 we had expected that the first lemma would be the correct one. We observed that if a success rate is very small the word cannot be correctly disambiguated. Thus the correct lemma was assigned using the rules in Tab. 6. Values of (t_1, t_2) were tested in the range (0,0)..(1,1). The best settings of thresholds on the learning set was $t_1 = 0$, $t_2 = 0.8$. Different values of k were explored, too, but no increase of k results in any increase of accuracy.

5.3 se

Results of disambiguation can be found in Tab. 7. The column #ex contains a number of examples. The next three columns displays the number of correctly and incorrectly disambiguated sentences and the accuracy. Last two columns display the absolute and the relative numbers of unresolved cases.

When the rule sets were used directly, the accuracy was quite high – 93.3% for preposition and 99.1% for pronoun – but the number of unresolved sentences reached 47.3% and 31.9%. For the method described above, the number of unresolved cases decreased to 1/3, from 47.3% to 12.5% for the first lemma. For the second lemma the improvement reached 1/2, from 31.9% to 14.2%. At the same time accuracy decreased only slightly – from 93.3% and 99.1% for the first algorithm to 93.0% and 97.5% for the second one. The time needed for the disambiguation of one sentence was 6 seconds on average, very rarely it was more than 10 seconds. If the disambiguation lasted more than 30 seconds, the process was killed. It concerned less than 2% of cases.

Table 7: Results of kNN algorithm

			disambiguation			unresolved	
		#ex	correct	wrong	accur.(%)	#	%
preposition	learn	99	80	4	97.5	17	17.2
	test	112	93	7	93.0	14	12.5
	test(PDTB)	218	168	34	84.2	16	7.3
pronoun	learn	297	214	2	99.1	82	27.6
	test	310	236	6	97.5	44	14.2
	test(PDTB)	254	212	17	92.7	25	9.9

We also tested the soundness of our approach on the Prague Dependency Tree Bank (PDTB) corpus that is under construction at the Charles University in Prague. The learning data were taken from the corpus DESAM. The results in Tab. 7 displays the accuracy 84.2% for the first lemma and 92.7% for the second one. The number of unresolved sentences is even smaller than for DESAM.

5.4 je

The second most frequent word with an ambiguous lemma is the word form *je*. It may be either one of the plural forms of the pronoun *on* (*them* in English) or the 3rd person of the verb *být* (like *is* in English). Applying the same approach as above we obtained results for the testing set shown in

Tab. 8. The accuracy is again very high but the number of unresolved cases is not acceptable.

Table 8: Results for *je*

		disambiguation			unresolved	
	#ex	correct	wrong	accuracy(%)	#	%
pronoun	31	16	1	94.1	14	45.0
verb	30	28	3	91.3	2	6.1

5.5 Disambiguation of unknown words

Our method was also used for disambiguation of unknown words (not existing in the corpus). In [Pavelek and Popelínský, 2000] we defined similarity classes for lemma-ambiguous words in the terms of grammatical categories. Then we are able to recognize easily that a word belongs to a particular similarity class just using the LEMMA morphological analyzer. E.g. the word form *Jana* can be either the accusative of the masculine *Jan* or nominative feminine *Jan*. Such first names form one of the similarity classes: their members have the same part-of-speech (noun) and they differ in gender (masculine/feminine). Tab. 9 displays summary results for first names. Description of the method can be found in [Pavelek and Popelínský, 2000].

Table 9: First names

		disambiguation			unresolved	
	#ex	correct	wrong	accuracy(%)	#	%
masculine	97	68	3	95.8	26	26.8
feminine	44	19	5	79.2	20	45.5

6 Tag Disambiguation

We performed first experiments with tag disambiguation of nouns as well. We display the results for the class of words with inflectional paradigm *pán*

(*gentleman*, *master*) in Tab. 10. Complete results for other inflectional paradigms can be found in Appendix. In the corpus DESAM there are 9268 items with that paradigm. The used domain knowledge only contained the predicate

```
remove(LeftContext,RightContext,Tag)
```

that removes the tag `Tag` from the set of possible tags if the word has the left context `LeftContext` and the right context `RightContext` (as in [Cussens, 1997]). For each of ambiguous cases – *pána*, *pánovi*, *páni*, *pány* – 100 examples were used for learning disambiguation rules with Progol, the rest for testing. The context of length 1 was taken in the account. Then for each sentence in the test set, the facts `data/2` were prepared in the same way as for the lemma disambiguation (i.e. all combination of tags etc.).

In Tab. 10 the first two columns display a word form and possible tags for this word form. The third column contains the number of cases in the DESAM corpus. The accuracy of the learned rules for the test set (with known tags in the context) is in the fourth column. Exploiting the learned rules for disambiguation of unannotated text, the number and the percentage of tags that were removed are displayed in the 5th and the 6th columns. The last column contains the relative number of the correctly removed tags. The average accuracy 90.6% is promising. The other method based on hidden Markov chains shows the accuracy 81.6% [Pala et al., 1997]

7 Discussion

Fixing DESAM. A part of DESAM was automatically disambiguated by different methods. Later a quite significant number of tags was found incorrect. This holds very often also for the case of *se*: the word was incorrectly tagged as preposition instead of reflexive pronoun when a noun phrase in instrumental followed the word *se*. With our method, 85 cases out of 59 were correctly fixed, for 8 cases our method failed, and 18 cases were unrecognized, which is 88.1% accuracy.

Smaller learning set. We tested, too, whether the accuracy would significantly decrease when the smaller number of examples would be used for learning. Instead 80:20 ratio we used 20% of sentences for learning the theory for *se* (40 positive examples) and 10% of sentences of lemma *sebe* (210

Table 10: Results for pattern *pán*

Word	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
pána	k1gMnSc2 k1gMnSc4	4037	85.2	2248	55.6	80.6
pánovi	k1gMnSc3 k1gMnSc6	665	94.5	377	56.7	94.7
páni	k1gMnPc1 k1gMnPc5	3044	96.5	2827	92.9	99.5
pány	k1gMnPc4 k1gMnPc7	1522	85.4	1198	78.7	87.6
Total		9268	90.4	6650	71.0	90.6

positive examples) for learning the theory for *sebe*. The reason for the difference was just the speed of Progol. For the first theory the average learning time did not take more than 11 minutes, for the second one it was about 6 minutes. We repeated again the learning session five times and then we chose for each lemma the theory with the maximum success rate. The results of disambiguation are displayed in Tab. 11. The accuracy did not decrease significantly – the decrease is equal to 2.5% for the first lemma, and 0.5% for the second one. The number of unresolved cases increased only for the second lemma to 8.4%. It means that the ratio of splitting of the example set into training and testing data did not lower the accuracy significantly.

DESAM corpus. DESAM is still not large enough. It does not contain all Czech word forms – compare 132 000 different word forms in DESAM with 164 000 stems of Czech words that a morphological analyzer LEMMA is able to recognize (each of them can have a number of both prefixes and suffixes). Thus DESAM does not contain a representative set of Czech sentences. In addition DESAM contains some errors, i.e. incorrectly tagged words. Another problem is that the significant amount of word positions (words as well as commas, semicolons etc.) are untagged. For the word form *se* nearly one fifth of words in the context are untagged (16,8%) and 93.4% of contexts

Table 11: Results for the smaller learning set

			disambiguation			unresolved	
		#ex	correct	wrong	accuracy(%)	#	%
preposition	learn	99	86	3	96.6	10	11.1
	test	112	95	10	90.5	12	12.0
pronoun	learn	297	241	2	99.2	64	21.6
	test	310	240	5	98.0	70	22.6

contain an untagged word. It is similar for other classes of words with an ambiguous lemma (see Table 12).

Table 12: Frequency of untagged words(left/right context of 5 words)

Class of		Untagged words		Incomplete contexts	
word form	#occurrences	#	%	#	%
se	3167	5323	16.8	2957	93.4
je	2525	4718	18.7	2400	95.0
first names	124	219	17.7	109	87.9

8 Relevant Works and Conclusion

Cussens [Cussens, 1997] developed POS tagger for English that achieved per-word accuracy of 96.4%. Eineborg and Lindberg (Eineborg and Lindberg, 1998, Lindberg and Eineborg 1998, Lindberg and Eineborg 1999) induced constraint grammar-like disambiguation rules for Swedish with the accuracy of 98%. Our approach differs significantly in two points. We do not exploit any information on particular words like in [Eineborg and Lindberg, 1998]. Such knowledge would improve accuracy significantly. Neither do we use any hand-crafted grammatical domain knowledge as in [Cussens, 1997].
Inductive logic programming has not been applied for lemma disambiguation in inflectional languages yet. However, ILP has been successfully used for solving different subtasks of morphological (or morphosyntactic) analysis of

inflectional languages. In [Džeroski and 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%. It must be stressed that we did not employ any lexical statistics and we did not use any handcrafted domain knowledge. However, our method concerned only a subtask of morphosyntactic disambiguation – tag disambiguation of nouns.
Concerning morphosyntactic disambiguation in Czech corpora, statistical techniques (accuracy 81.64%) and neural nets (75.47%) have been applied to DESAM [Pala et al., 1997]. See also [Hajič and Hladká, 1997a], [Hajič and Hladká, 1997b], [Zavrel and Daelmans, 1998] for other results with another Czech corpus. It should be pointed out that our results are not comparable with these works because we focus only on subtasks of morphological disambiguation.
The lemma disambiguation task is not solved here completely. The main reason for that is the size of the Czech corpora DESAM and PDTB. Both corpora are still too small and therefore the size of learning sets is not very often sufficient for disambiguation. We demonstrated that our method can be successfully used for ambiguous words that are frequent in corpora. Even if developed for the Czech language, the method is actually language-independent. It can be used for other Slavic languages without significant modifications.

Acknowledgements

We thank a lot to James Cussens and Sašo Džeroski for their comments. Thanks are also due to anonymous referees of LLL'99 workshop. We would like to thank to Karel Pala and Olga Štěpánková for their help with earlier versions of this paper, and to our colleagues Pavel Rychlý, Radek Sedláček and Robert Král for fruitful discussions and assistance. This work has been partially supported by VS97028 grant of Ministry of Education of the Czech Republic "Natural Language Processing Laboratory" and ESPRIT ILP^2 Project.

References

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

[Cussens et al., 1999] Cussens, J., Džeroski, S., and Erjavec, T. (1999). Morphosyntactic tagging of Slovene using Progol. In Džeroski, S. and Flach, P., editors, *Inductive Logic Programming: Proc. of the 9th International Workshop (ILP-99)*, Bled, Slovenia. Springer-Verlag.

[Džeroski and Erjavec, 1997] Džeroski, S. and Erjavec, T. (1997). Induction of Slovene nominal paradigms. In *Inductive Logic Programming: Proceedings of the 7th International Workshop (ILP-97). LNAI 1297*, pages 141–148. Springer.

[Eineborg and Lindberg, 1998] Eineborg, M. and Lindberg, N. (1998). Induction of constraint grammar-rules using Progol. In *Inductive Logic Programming: Proceedings of the 8th International Conference (ILP-98).* Springer.

[Hajič and Hladká, 1997a] Hajič, J. and Hladká, B. (1997a). Probabilistic and rule-based tagger of an inflective language – a comparison. In *Proceedings of the 5th Conf. on Applied Natural Language Processing,*, pages 111–118.

[Hajič and Hladká, 1997b] Hajič, J. and Hladká, B. (1997b). Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of EACL,*.

[Lindberg and Eineborg, 1998] Lindberg, N. and Eineborg, M. (1998). Learning constraint grammar-style disambiguation rules using inductive logic programming. In *Proc. COLING/ACL98.*

[Lindberg and Eineborg, 1999] Lindberg, N. and Eineborg, M. (1999). Improving part of speech disambiguation rules by adding linguistic knowledge. In Džeroski, S. and Flach, P., editors, *Inductive Logic Programming: Proc. of the 9th International Workshop (ILP-99)*, Bled, Slovenia. Springer-Verlag.

[Manandhar et al., 1998] Manandhar, S., Džeroski, S., and Erjavec, T. (1998). Learning multilingual morphology with CLOG. In *Inductive Logic Programming: Proceedings of the 8th International Conference (ILP-98).* Springer.

[Mitchell, 1997] Mitchell, T.M.: Machine Learning. McGraw Hill, New York, 1997.

[Muggleton and De Raedt, 1994] Muggleton S. and De Raedt L.: Inductive Logic Programming: Theory And Methods. J. Logic Programming 1994:19,20:629-679.

[Pala and Ševeček, 1995] Pala K. and Ševeček P. (1995). Lemma morphological analyser. User manual. Lingea Brno.

[Pala et al., 1997] Pala, K., Rychlý, P., and Smrž, P. (1997). DESAM - annotated corpus for czech. In *In Plášil F., Jeffery K.G.(eds.): Proceedings of SOFSEM'97, Milovy, Czech Republic. LNCS 1338*, pages 60–69. Springer.

[Pavelek and Popelínský, 2000] Pavelek, T. and Popelínský L. (2000). On lemma disambiguation: Similarity classes. In *In Proceedings of Summer School on Information Systems, Ruprechtov.* Technical University Brno.

[Popelínský et al., 1999] Popelínský L. and Pavelek T. Mining lemma disambiguation rules from Czech corpora In Proc. of 3rdEuropean Conference PKDD'99, Prague Czech Republic 1999. LNCS 1704 pp.498–503, Springer-Verlag 1999.

[Popelínský et al., 2000] Popelínský L., Pavelek T., Ptáčník. T. (2000). Towards disambiguation in Czech corpora. In Proc. of the 1st Learning Language in Logic Workshop LLL99, Bled, 1999.

[Zavrel and Daelmans, 1998] Zavrel, J. and Daelmans, W. (1998). Recent advances in memory-based part-of-speech tagging. Technical report, ILK/Computaional Linguistics, Tilburg University.

Appendix

This appendix contains the results performed by Tomáš Ptáčník. All theories were learned using context of the length of 1.

Table 13: Results for pattern *pán*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
pána	k1gMnSc2					
	k1gMnSc4	4037	85.20	2248	55.64	80.60
pánovi	k1gMnSc3					
	k1gMnSc6	665	94.51	377	56.69	94.69
páni	k1gMnPc1					
	k1gMnPc5	3044	96.50	2827	92.87	99.50
pány	k1gMnPc4					
	k1gMnPc7	1522	85.44	1198	78.71	87.56
Total		9268	90.41	6650	70.98	90.59

Table 14: Results for pattern *muž*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
muže	k1gMnSc2 k1gMnSc4 k1gMnPc4	2072	77.32	1099	26.51	84.53
muži	k1gMnSc3 k1gMnSc5 k1gMnSc6 k1gMnPc1 k1gMnPc5 k1gMnPc7	2742	84.21	4819	35.15	96.85
Total		4814	80.77	5918	30.83	90.69

Table 15: Results for pattern *předseda*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
předsedy	k1gMnSc2 k1gMnPc4 k1gMnPc7	471	85.09	145	15.39	90.34
předsedovi	k1gMnSc3 k1gMnSc6	73	86.79	33	45.21	93.94
předsedové	k1gMnPc1 k1gMnPc5	447	100.00	433	96.87	100.00
Total		991	90.63	611	52.49	94.76

Table 16: Results for pattern *soudce*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
soudce	k1gMnSc1 k1gMnSc2 k1gMnSc4 k1gMnSc5 k1gMnPc4	766	84.88	165	5.37	98.18
soudci	k1gMnSc3 k1gMnSc6 k1gMnPc1 k1gMnPc5 k1gMnPc7	441	85.45	33	1.87	100.00
Total		1207	85.17	198	3.62	99.09

Table 17: Results for pattern *hrad*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
hrad	k1gInSc1 k1gInSc4	28718	73.07	15762	54.82	55.54
hradu	k1gInSc2 k1gInSc3 k1gInSc6	28658	94.21	40400	70.44	93.25
hrady	k1gInPc1 k1gInPc4 k1gInPc5 k1gInPc7	11398	86.85	18492	54.07	92.59
Total		68774	84.71	74654	59.78	80.46

Table 18: Results for pattern *stroj*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
stroj	k1gInSc1 k1gInSc4	2134	73.14	1055	49.41	57.35
stroje	k1gInSc2 k1gInPc1 k1gInPc4 k1gInPc5	2478	85.49	908	12.21	100.00
stroji	k1gInSc3 k1gInSc5 k1gInSc6 k1gInPc7	1078	96.86	702	21.71	100.00
Total		5690	85.16	2665	27.78	85.78

Table 19: Results for pattern *žena*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
ženy	k1gFnSc2 k1gFnPc1 k1gFnPc4 k1gFnPc5	21159	88.77	10756	16.94	99.99
ženě	k1gFnSc3 k1gFnSc6	8668	96.74	4639	53.50	97.93
Total		29827	92.76	15395	35.22	98.96

Table 20: Results for pattern *růže*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
růže	k1gFnSc1					
	k1gFnSc2					
	k1gFnSc5					
	k1gFnPc1					
	k1gFnPc4					
	k1gFnPc5	13415	81.64	1764	2.63	99.38
růži	k1gFnSc3					
	k1gFnSc4					
	k1gFnSc6	7599	86.93	5595	36.81	85.29
růží	k1gFnSc7					
	k1gFnPc2	3378	90.04	2527	74.81	98.18
Total		24392	86.20	9886	38.08	94.28

Table 21: Results for pattern *píseň*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
píseň	k1gFnSc1					
	k1gFnSc4	7046	77.36	4499	63.72	78.15
písně	k1gFnSc2					
	k1gFnPc1					
	k1gFnPc4					
	k1gFnPc5	1616	88.85	854	17.62	100.00
písni	k1gFnSc3					
	k1gFnSc5					
	k1gFnSc6	2958	97.07	2760	46.64	100.00
písní	k1gFnSc7					
	k1gFnPc2	3302	83.95	2729	82.65	83.36
Total		14922	86.81	10842	52.65	90.38

Table 22: Results for pattern *kost*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
kost	k1gFnSc1					
	k1gFnSc4	7047	77.36	4499	63.71	78.15
kosti	k1gFnSc2					
	k1gFnSc3					
	k1gFnSc5					
	k1gFnSc6					
	k1gFnPc1					
	k1gFnPc4					
	k1gFnPc5	14596	91.85	4013	9.10	95.04
kostí	k1gFnSc7					
	k1gFnPc2	6504	84.08	2853	86.40	80.37
Total		28147	84.43	11365	53.07	84.52

Table 23: Results for pattern *město*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
město	k1gNnSc1					
	k1gNnSc4					
	k1gNnSc5	5437	74.18	3854	35.44	94.68
města	k1gNnSc2					
	k1gNnPc1					
	k1gNnPc4					
	k1gNnPc5	5400	90.23	2135	13.18	99.81
městu	k1gNnSc3					
	k1gNnSc6	422	76.34	253	59.95	93.68
Total		11259	80.25	6242	36.19	96.06

Table 24: Results for pattern *moře*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
moře	k1gNnSc1					
	k1gNnSc2					
	k1gNnSc4					
	k1gNnSc5					
	k1gNnPc1					
	k1gNnPc4					
	k1gNnPc5	704	82.62	266	6.30	99.62
moři	k1gNnSc3					
	k1gNnSc6					
	k1gNnPc7	444	96.32	74	13.60	100.00
Total		1148	89.47	340	9.95	99.81

Table 25: Results for pattern *kuře*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
kuře	k1gNnSc1					
	k1gNnSc4					
	k1gNnSc5	410	77.53	3153	28.99	90.96
kuřata	k1gNnPc1					
	k1gNnPc4					
	k1gNnPc5	176	90.46	0	0.00	0.00
kuřeti	k1gNnSc3					
	k1gNnSc6	4	50.00	4	100.00	75.00
Total		590	72.66	3157	43.00	55.32

Table 26: Results for pattern *stavení*

Word form	Tags	#ex	accuracy on test data	tags removed #	tags removed %	disambiguation accuracy
stavení	k1gNnSc1					
	k1gNnSc2					
	k1gNnSc3					
	k1gNnSc4					
	k1gNnSc5					
	k1gNnSc6					
	k1gNnPc1					
	k1gNnPc2					
	k1gNnPc4					
	k1gNnPc5	24955	86.03	519	0.27	100.00
stavením	k1gNnSc7					
	k1gNnPc3	2351	96.45	3	0.13	100.00
Total		27306	91.24	522	0.20	100.00

Publications in the FI MU Report Series are in general accessible via WWW and anonymous FTP:

`http://www.fi.muni.cz/informatics/reports/`
`ftp ftp.fi.muni.cz (cd pub/reports)`

Copies may be also obtained by contacting:

**Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic**