Towards disambiguation in Czech corpora

Luboš Popelínský, Tomáš Pavelek, Tomáš Ptáčník
Natural Language Processing Laboratory
Faculty of Informatics, Masaryk University
Brno, Czech Republic
Email: {popel,xpavelek, ptacnik}@fi.muni.cz

May 12, 1999

Abstract

Lemma disambiguation means finding a basic word forms (e.g. nominative singular for nouns, infinitive for verbs). In Czech corpora it was observed that 10% of word positions have at least 2 lemmata. We present results of the most important subtasks of lemma disambiguation using combination of ILP and MBL techniques. We propose a way how to use lemma disambiguation rules learned with ILP system Progol to minimise a number of incorrectly disambiguated words. Although no knowledge on Czech grammar has been used the accuracy reaches 90% with some fraction of words remaining ambiguous. Our methods have been also proved valid for for another Czech corpus. A part of DESAM has been automatically disambiguated by another disambiguator resulting in a great number of incorrect tags. Exploiting our approach we can fix it with 88% accuracy. We display first results of tag disambiguation, too.

This is an extended version of the PKDD'99 submission
Luboš Popelínský, Tomáš Pavelek:
Mining lemma disambiguation rules from Czech corpora.

1 Disambiguation in Czech corpus

Disambiguation in inflective languages [1], including Czech, is a very challenging task because of their usefulness as well as its complexity. Czech corpus DESAM is now being built at Faculty of Informatics, Masaryk University – 66 grammatical categories like a part-of-speech, gender, case, number etc., about 2 000 tags, combinations of category-value couples. [2] It was observed [11] that there is in average 4.21 possible tags per word. It is impossible to perform the disambiguation task manually and any tool that can decrease the amount of human work is welcome.

It should be noticed here that the disambiguation task in Czech language is much more complex than in e.g. English also for another reason. For English there are tagged corpora covering a majority of common English sentences. Those corpora seem to be without errors. Grammar rules are known that covers a significant part of English sentence syntax. Unfortunately neither of those statements hold for Czech. It makes our task quite difficult.

[1]This work has been partially supported by VS97028 grant of Ministry of Education Natural Language Processing Laboratory

[2]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 done by combination of those categories.

2 DESAM corpus

DESAM, a corpus of Czech newspaper texts, which is now being disambiguated – partially manually, partially by means of different disambiguators – contains more than 1 000 000 word forms, about 130 000 different word forms, about 65 000 of them occuring more then once, and 1665 different tags. Characteristics of DESAM are in the following table [11].

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

Table 1 : Characteristics of DESAM Czech 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 analyser LEMMA [3] is able to recognise (each of them can have a number of both prefixes and suffixes).

3 Lemma disambiguation

Lemma disambiguation which we address here, means assigning to each word form its basic form – nominative singular for nouns, adjectives, pronouns and numerals, infinitive for verbs. E.g. in the sentence *Od rána je má Ivana se ženou.* (literarily *since morning my Ivana(female) is with (my) wife.*) each of words except for the preposition "od" has two basic forms. E.g. “rána” can be genitive of “ráno”(morning) as well as nominative of a substantive “rána”(bang). In the following example the tag k1gNnSc2 for the word “rána” (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.

```
Od      <l> od      <c> k7c2
rána    <l> ráno    <c> k1gNnSc2,k1gNnPc145
        <l> rána    <c> k1gFnSc1
je      <l> být     <c> k5eAp3nStPmIaI
        <l> on      <c> k3xPgNnSc4p3,k3xPgXnPc4p3
má      <l> můj     <c> k3xOgFnSc15p1,k3xOgNnPc145p1
        <l> mít     <c> k5eAp3nStPmIaI
Ivana   <l> Ivan    <c> k1gMnSc24
        <l> Ivana   <c> k1gFnSc1
se      <l> s       <c> k7c7
        <l> sebe    <c> k3xXnSc4
ženou   <l> žena    <c> k1gFnSc7
        <l> hnát    <c> k5eAp3nPtPmIaI
.
```

Pic. 1: Lemma ambiguities in a Czech sentence

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 the Czech vocabulary has at least 2 lemmata. The most frequent ambiguous word forms are **se** and **je** - see their frequencies compared with the most frequent Czech words.

[3] copyright Lingea Brno 1995

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

Table 2: Most frequent words in DESAM

We can see that disambiguation of the words “se” and “je” would be very useful as they are 3rd and 5th most frequent words in DESAM corpus. Actually the lemma disambiguation (almost always) leads to a disambiguation of sense. In the example, *má* means either "my" (daughter) or "has" (s/he has), *se* is either preposition "with" (my wife) or the reflexive pronoun (like "elle *se* lave" in French).

4 Inductive logic programming and NLP

Inductive logic programming (ILP) aims at finding first-order logic rules that covers positive examples (and uncovers negative ones) using given domain knowledge predicates. In the past, ILP has been applied for inflective languages in the field of morphology. LAI Ljubljana [2] applied ILP for generating the lemma from the oblique form of nouns as well as for generating the correct oblique form from the lemma, with average accuracy 91.5 % . Learning nominal inflections for Czech and Slovene (among others) is described in [7]. There are two works that strongly influenced our work. James Cussens [1] developed POS tagger for English that achieved per-word accuracy of 96.4 %. Martin Eineborg and Nikolaj Lindberg [3, 4] induced constraint grammar-like disambiguation rules for Swedish with accuracy 98%.

5 Overview of our method

We show how to employ inductive logic programming, namely Progol [10] system for building reliable disambiguators. For a given word we first learn two sets of disambiguation rules, one for each of two lemmata. E.g. (see the example above) for the word form `se` we will have a set of rules that succeed if the word is a preposition, and another set of rules that hold if `se` is a reflexive pronoun. As learning sets we used sentences from hand-disambiguated part of DESAM corpus. Than we take a set of sentences not used in the learning step that contains the given word form. Employing the morphological analyser LEMMA we find all possible tags for words that appear in the context of the given word. After we generate all combination of those tags. For the sentence above and a context of length 1 LEMMA returns following tags for the word forms *Ivana, ženou*

Ivana	*(se)*	*ženou*
k1gMnSc24		k1gFnSc7
k1gFnSc1		k5eAp3nPtPmIaI

and 4 combination of tags $data(LeftContext, RightContext)$

data([k1gMnSc24] , [k1gFnSc7]).
data([k1gFnSc1] , [k1gFnSc7]).
data([k1gMnSc24] , [k5eAp3nPtPmIaI]).
data([k1gFnSc1] , [k5eAp3nPtPmIaI]).

Now the two sets of rules are applied to the set of *data(LeftContext, RightContext)*. The set of rules wins which covers the set of *data/2* facts better. Let the number of covered facts be the right criterion. In our example, if it is a rule set for the lemma *se_preposition* that covers more facts, the word form *se* is tagged as a preposition in the sentence segment *Ivana se ženou.*

Let us notice that the number of *data/2* facts may be enormous. E.g. for right and left contexts of the length 3, and for the average number of possible tags per word 4.21 we have more than $4^3 * 4^3 = 4096$ combinations of tags, i.e. more than 4096 facts *data/2*.
■
The rest of the paper is organised as follows: In the Section 6 we explain show a way how to build basic domain knowledge if no sufficient linguistic knowledge is available. In the Section 7 we present results obtained with ILP syste, Progol for the most frequent lemma-ambiguous word form "se". Rule set accuracy on disambiguated context is displayed. Section 8 brings results of disambiguation when correct tags in a context were unknown. We introduce MBL algorithm that employs rule sets learned with Progol. The algorithm is sufficient for automatic disambiguation and minimise the number of incorrectly disambiguated words leaving some of the words unresolved. We show there, too, that our method works well when applied to other words with ambiguous lemma. Experiments with another Czech corpus are included. In Section 10 we display first results of tag disambiguation of nouns.

6 Domain knowledge

There is no complete description of Czech grammar. We decided to build domain knowledge predicates without any need of deep linguistic knowledge. We exploit only information about a form of particular tag(s) in a particular part of context. In following "word" means the word that is to be disambiguated.

6.1 Basic terms

General form of domain knowledge predicates is

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

where `Context` is a variable bound with either left or the right context, `Focus, Condition` are terms. `Focus` defines a subpart of the `Context`. `Condition` says what condition must hold on the `Focus`. The predicate `p(Context, Focus, Condition)` actually represents a class of predicates. Particular members of that class differ in focus predicates and conditions. E.g. a goal `p(X,first(4), somewhere([k1])` succeeds if somewhere in the 4 words of the context X that neighbours with the word there is a substantive (k1). The goal `p(X,first(2), always([c7,nS]))` succeeds if tags `c7,nS` appears in both of the first two words in the context `X` – e.g. a pronoun and a substantive in instrumental of singular like in "s tvým bratrem - (with) your brother".

Domain knowledge predicates are built in the following way.

`Focus` has a form

- `first(N)` (N=1..max_length) ... a sublist of the `Context` of length `N` neighbouring 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` ;

6.2 Specialisation operator

We do not allow predicates of the form "somewhere appears a tag T or a tag T1". Then the predicate

```
p(X,first(Max_length),somewhere([T])
```

is actually the most general in that class of predicates as it tests the most general condition `somewhere([T])` on the longest possible context. We can obtain other predicates by its specialisation. Specialisation operator consists of specialisation operations that are of two kinds. The first one decreases the length of the context by 1, the second one adds a tag into the `Condition`. Such operations ensure that the operator is locally complete. Let the context be not longer than `max_length` words and let a focus be a sublist (compact part of the context) neighbouring with the word that is to be disambiguated Then the maximal specialisations of `p(X,first(4),somewhere([T])` are
`p(X,first(3),somewhere([T])`
and `p(X,first(4),somewhere([T1,T2])`

The basic specialisation operations are following:

1. *spec*(`p(X,first(N),Y)`) = `p(X,first(N-1),Y)`
   for $1 \leq N \leq max\_length$;
2. *spec*(`p(X,Y,somewhere(Z))`) = `p(X,Y,somewhere([U|Z]))`
   if $\neg member(U, Z)$. ;
3. *spec*(`p(X,Y,always(Z))`) = `p(X,Y,always([U|Z]))`
   if $\neg member(U, Z)$. .

The specialisation operator is not proper: `p(X,first(1),somewhere(T))` and `p(X,first(1), always(T))` are equivalent. In the current version, all such couples `(Focus,Condition)` need to be explicitly known before building domain knowledge to save properness of the specialisation operator.

When building the current version of the domain knowledge specialisations (2) and (3) might be applied at most 3-times, i.e. max_length has been set to 3 and conditions on appearance of not more than 4 tags can be used.

We also performed experiments with broader class of predicates. Namely a `subset(Length)` has been added, i.e. any subset of the length might appear. However, this extension does not improve an accuracy significantly.

7 Learning disambiguation rules with Progol

Problem 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).

Learning sets For generation of learning sets we used the part of DESAM corpus which has been manually disambiguated. Left and right contexts have been set to 4 words. Negative examples have been built from sentences where a word has the second lemma.

Results Using P-Progol [10] version 2.2 we have learned rules for both of the two lemmata. It means that we have for each task 2 rule sets that might be complementary. However, we have found it usefull to use both of them. The results are in the following table. Number of sentences for each lemma is in the 2nd column. They have been split in 80:20 ratio and 80% of examples have been used for learning. We tested each rule set on the rest of data. The accuracy on the testing set is in the 3rd column.

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

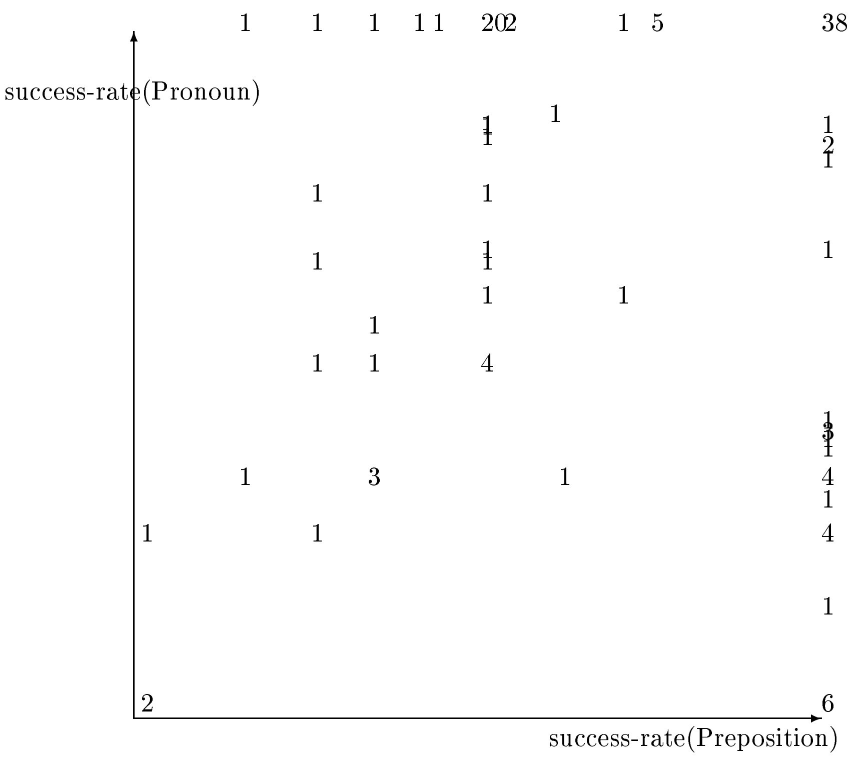
Table 3 : Results of learning

Run times Run times reached 14 hours when spliting examples in 80:20 ratio. It is caused by an enormous number of 4536 literals that may appear in a rule body. We tested, too, whether the accuracy would significantly decrease when less number of examples would be used for learning. It was observed that even 10% examples is enough for learning theories with accuracy about 90%. It means that the ratio of splitting example set into training and testing data did not decrease significantly the accuracy. It may imply that sentences in the corpus are similar to each other (we are not so courageous to say "representative") In the case of 10% learning set, run times was not greater than 5 minutes.

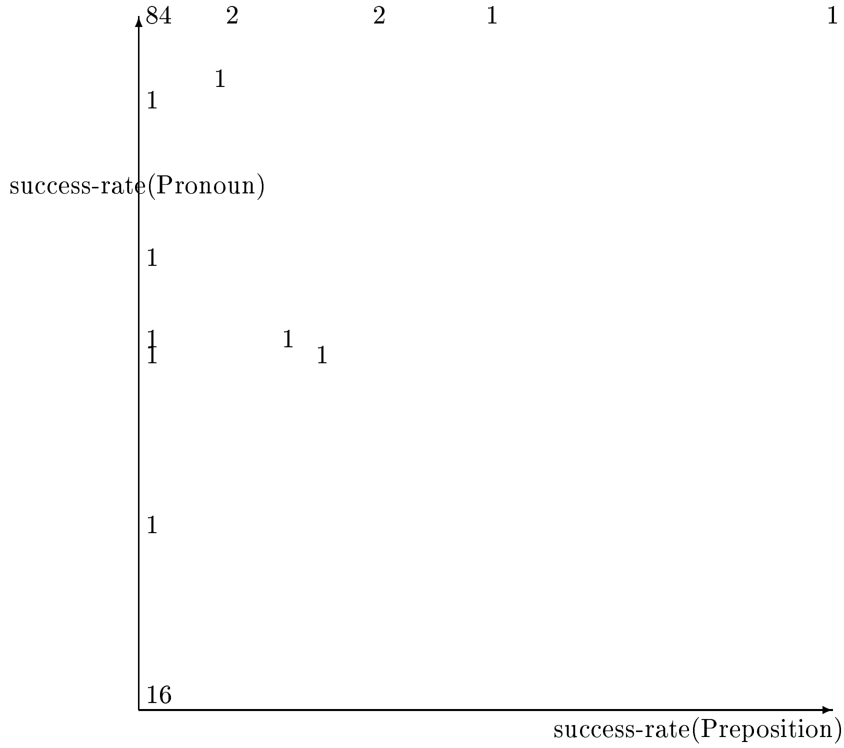
8 Disambiguation

8.1 MBL learning sets

The goal now was to find such a criterion that would allow to find the correct lemma for the word "se". We have chosen k-nearest neighbour method [8]. A part of DESAM corpus have been used as learning set that contained sentences not used for learning the disambiguation rules. With LEMMA morphological analyser first all possible grammatical categories have been found for each sentence. Then all variations of categories have been generated for each sentence. Success rates of theories of both lemmata have been computed. Thus we have received (x,y)-coordinates. The new example is than classified into class(lemma) of the nearest neighbour(s) in that learning set. The learning sets are in the following pictures. The point (0,0) is in the left-bottom corner, (1,1) is in the upper-right corner.



Pic. 2: Preposition cases



Pic. 3: Reflexive pronoun cases

8.2 Learning kNN parameters

has been performed as follows:

1. k nearest points is found, where the distance between 2 points $(x, y), (x_i, y_i)$ is computed as Euclidian distance.
2. For those k points, sums of number of occurences of lemma_1 (s_1) and lemma_2 (s_2) in those points are computed.

Parameters of kNN has to be found, namely

k . . .	a number of nearest neighbours
w_{lemma_1}	weight of the success rate of the first lemma
w_{lemma_2}	weight of the success rate of the second lemma

On new sentences we tested values of k in the range 1..10, (w_x, w_y) in the range (0,0)..(1,1) and the best triple(k, w_x, w_y) has been used for neighbours calculations. It was observed that increasing value of k did not increase an accuracy of disambiguation. Thus or all experiments bellow k was set to 1.

8.3 Finding the correct lemma

From the corpus we have chosen new sentences that have not been used so far. We limited both left and right context to the length of 3 words. Then we removed all sentences that contained commas, dots, parentheses etc. Again with LEMMA morphological analyser all possible grammatical categories have been found for each sentence and all variations of categories have been generated for each sentence.

Then both theories learned with Progol have been run on that data so that for each sentence we had two success rates (i.e. relative number of examples covered by the particular theory) (x, y). The nearest point (x_i, y_i) has been found as well as the number of instances with lemma "s" (s_1) and the number of instances with the lemma "sebe" (s_2). The correct lemma has been assigned using the rules below:

lemma $= if\ w_{lemma_1} * s_1 > s_2\ then\ lemma_1$
lemma $= if\ s_1 < w_{lemma_2} * s_2\ then\ lemma_2$
lemma $=$ *unresolved* otherwise

The best settings of $lemma_i$ was $lemma_1 = lemma_2 = 1$. Results are in the Table 4.

	#ex	accuracy(%)	unresolved(%)
preposition	238	89.8	1.3
reflexive pronoun	242	94.8	3.3

Table 4 : Results MBL for DESAM

8.4 Testing on another corpus

We have tested the properness of our approach on Prague Tree Bank corpus that is under construction in Charles University in Prague.

	#ex	accuracy(%)	unresolved(%)
preposition	218	84.2	7.3
reflexive pronoun	254	92.7	0.9

Table 5 : Results MBL for PTB

The results show very good accuracy although all the learning data have been taken from another source – corpus DESAM. We claim that our approach is capable to find real dependencies in Czech language.

8.5 Fixing DESAM

A part of DESAM has been automatically disambiguated by hand-coded rules [14]. However, a quite significant number of tags has been found incorrect. It happened very often also in the case of "se": the word have been incorrectly tagged as preposition instead of a reflexive pronoun. In 85 cases, 18 has been unrecognised, 59 have been correctly fixed and on 8 cases our method failed, what means 88.1 % accuracy.

8.6 Case of "je" ambiguity

The second most frequent word with an ambiguous lemma is the word "je" . It may be either one of plural form of a pronoun "on" (like "them" in English) or 3rd person of the verb "být" (like "is" in English). Applying the same approach as above we have obtained results that follows.

	"on"	"být"
#examples for learning theories	151	2440
rule set accuracy(%)	94.7	91.4
disambiguation accuracy(%)	100.0	91.3
covered : correctly/incorrectly	17/0	28/3
unresolved sentences(%)	45.0 (14 cases)	6.1 (2 cases)
$weight_lemma$	(0.63, 1)	(0.1,0.7)

Table 6 : Results for "je"

9 Comprehensibility of rules

It need to be stressed that the richer domain knowledge brings, beside the higher accuracy, a better comprehensibility. For instance for the reflexive pronoun/preposition **"se"** the obtained rules appear to be interesting from the linguistic point of view. In the case of preposition the rule

```
preposition(Left_Context,Right_Context) :-
      p(Right_Context,first(2),always([c7])),
      p(Right_Context,first(2),always([nS])).
```

means that very often after that preposition there are two words in instrumental and singular – e.g. substantive, adjective, pronoun or numeral and it is not important what part-of-speech they belong to. It is interesting, too, that no similar rule has been learned that tests plural (e.g. `p(Left_Context,first(2),always([nP]))`): is is surprising as no knowledge exists about preference singular to plural for this preposition!

10 Tag disambiguation

Similar approach has been performed for tag disambiguation of nouns. We will display results for the class of words with pattern "pán" (gentleman, master). In this case the used domain knowledge contained only predicates `remove(LeftContext,RightContext, Tag)` like in [1] and the context of the length 1 have been taken in account.

In the Table 7., 1st column displays a word form and possible tags for the particular word form. 2nd column contains the number of examples extracted from DESAM corpus. Then 100 examples have been used for learning disambiguation rules with Progol. Accuracy of the learned rules on the rest of data is in the 3rd column. Then data have been prepared for tag disambiguation by the same way as for lemma disambiguation (i.e. all combination of tags etc.). Exploiting the learned rules, some tags have been removed – see 4th column. The last column contains a relative number of correctly removed tags.

Word form	Tags	#ex	accuracy on desamb.context	#tags removed	accuracy
pána	k1gMnSc2 k1gMnSc4	4037	85.2	2248	80.6
pánovi	k1gMnSc3 k1gMnSc6	665	94.5	1042	98.1
páni	k1gMnPc1 k1gMnPc5	3044	96.5	2827	99.5
pány	k1gMnPc4 k1gMnPc7	1522	85.4	1522	87.6

Table 7: Pattern "pán"

11 Conclusion

So far statistical techniques (accuracy 81.64%) and neural nets (75.47%) were applied to DESAM [11]. See also [5, 6, 18] for other results with another Czech corpus. It should be pointed out that our results are not quite comparable as we focus only on lemma disambiguation.

Our approach differs significantly from the earlier works [1, 3]. We do not exploit any information on particular words as in [3]. Such knowledge would improve an accuracy significantly. Neither we use any hand-coded grammatical domain knowledge as in [1].

The presented results are the first ones obtained with ILP techniques in disambiguation of inflective languages, as we know. We described the way of building domain knowledge if no sufficient linguistic knowledge is available. Even though the Czech corpus is under development (and therefore it contains incorrectly tagged words) the theories learned with Progol overcome 92% accuracy on testing data. We presented an algorithm for automatic lemma disambiguation that reached an accuracy nearly 90% with a very small amount of words leaving ambiguous.

Our approach can be used for disambiguation unknown words (not existing in a corpus). We defined similarity classes on lemma-ambiguous words in terms of grammatical categories. It allows to easily recognise that a word belongs to a similarity class using output of LEMMA morphological analyser. I E.g. the word form "Jana" can be either accusative of masculine "Jan" or nominative of feminine "Jana". Such first names forms one of the similarity classes: their member have same the part-of-speech tag (noun) and they differs in a gender (masculine/feminine). First results can be found in [13].

Concerning future work, the project that is coming to its (hopefully successful) end consists in tag disambiguation. Next we plan to extend the domain knowledge with some basic predicates on noun phrases [16] and verb valencies [15]. It seems to be possible to use our approach for other inflective languages as well.

Acknowledgements

We would like to thank to Karel Pala, Pavel Rychlý as well as our colleagues from 1998 ILP courses for fruitful discussions and assistance, namely to Radek Sedláček and Robert Král.

References

[1] Cussens J.: Part-of-Speech Tagging using Progol. In Proc. of ILP'97, LNAI 1297, Springer-Verlag 1997.

[2] Džeroski S., Erjavec T.: Induction of Slovene Nominal Paradigms. In Proc. of ILP'97, LNAI 1297, Springer-Verlag 1997.

[3] Eineborg M., Lindberg N.: Induction of Constraint Grammar rules using Progol. In Proc. of ILP'98, 1998.

[4] Lindberg N., Eineborg M.: Learning Constraint Grammar-style disambiguation rules using Inductive Logic Programming. In: COLING/ACL'98.

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

[6] Hajič J., Hladká B.: Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In Proceedings of EACL 1998.

[7] Manandhar S., Džeroski S., Erjavec T.: Learning multilingual morphology with CLOG. In Proc. of ILP'98, 1998.

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

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

[10] Muggleton S.: Inverse Entailment and Progol. New Generation Computing Journal, 13:245-286, 1995.

[11] K. Pala , P. Rychlý and P. Smrž: DESAM - annotated corpus for Czech. In Plášil F., Jeffery K.G.(eds.): Proceedings of SOFSEM'97, Milovy, Czech Republic. LNCS 1338, Springer-Verlag 1997.

[12] K. Pala , P. Rychlý and P. Smrž: DESAM - Approaches to Disambiguation. Tech. Report FIMU-RS-97-09. `http://www.fi.muni.cz/informatics/reports/`

[13] Pavelek T., Popelínský L.: Towards lemma disambiguation: Similarity classes (submitted)

[14] Puža V.: Syntactic Analysis of Natural Language Focused on Corpora Tagging. Master thesis, Faculty of Informatics, Masaryk University, Brno 1997. (in Czech).

[15] Smrž P., Horák A.: Determining Type of TIL Construction with Verb Valency Analyser. In Proceedings of SOFSEM'98, LNCS Springer-Verlag 1998.

[16] Smrž P., Žáčková E.: New Tools for Disambiguation of Czech Texts. In [17].

[17] Sojka P., Matoušek V., Pala K., Kopeček I.: Text, Speech, Dialogue. Proceedings of the 1st Workshop on Text, Speech, Dialogue - TSD'98, Brno, Czech Republic, Sept. 1998.

[18] Zavrel J., Daelemans W.: Recent Advances in Memory-Based Part-of-Speech Tagging. ILK/Computaional Linguistics, Tilburg University, 1998.