Elektrotehniški vestnik 65(5): 296–302, 1998
Electrotechnical Review, Ljubljana, Slovenija

Inductive Learning of Multilingual Morphology

Sašo Džeroski, Tomaž Erjavec

Jožef Stefan Institute, Department of Intelligent Systems,
Jamova 39, SI-1000 Ljubljana, Slovenia
E-mail: Saso.Dzeroski@ijs.si, Tomaz.Erjavec@ijs.si

Abstract. The paper presents results of learning nominal inflections of English, Romanian, Czech, Slovene, and Estonian using inductive logic programming. The same approach had previously been tested on the problem of inducing rules for forming the past tense of English verbs. The languages discussed here have, unlike English, rich inflectional morphology, and the paper reports the result of using the MULTEXT-East multilingual tagged corpus to induce rules for the synthesis and analysis of the inflectional paradigms of nouns and adjectives. The resulting rule-set can be used, especially in conjunction with a tagger, to constrain the possible lemmas or morphosyntactic interpretations of word-forms, or to generate word-forms from lemmas, for five typologically very different languages.

Key words: machine learning, natural language processing, multilingual morphology

Induktivno učenje večjezičnega oblikoslovja

Povzetek. Članek predstavi rezultate učenja pregibanj samostalniških besed angleškega, romunskega, češkega, slovenskega in estonskega jezika z metodami induktivnega logičnega programiranja. Ista metoda je bila pred tem uporabljena za indukcijo pravil za tvorjenje angleških glagolov v pretekliku. Jeziki, ki jih obravnavamo tu, imajo, za razliko od angleščine, bogato oblikoslovje in članek predstavi rezultate uporabe MULTEXT-East večjezičnega označenega korpusa za induciranje pravil za sintezo (tvorjenje) in analizo oblikoslovnih paradigm samostalnikov in pridevnikov. Naučeno množico pravil je mogoče, posebej še v kombinaciji z oblikoslovnim označevalnikom, uprabiti za omejtev možnih gesel oz. oblikoslovnih oznak besednih oblik, oziroma za tvorjenje besednih oblik iz gesel, za pet tipološko zelo različnih jezikov.

Ključne besede: strojno učenje, procesiranje naravnega jezika, večjezično oblikoslovje

1 Introduction

Machine learning methods have been recently applied to a variety of tasks within the area of natural language processing [3,13]. Systems that learn relations, called inductive logic programming (ILP) [9] systems, have been applied to tasks such as learning to parse [10] and learning part-of-speech tagging [1]. Learning of morphological structure has also been attempted [11], but experiments have focussed on relatively small samples of English.

This paper is concerned with using inductive logic programming to learn a significant portion of the morphology of five languages, which are, furthermore, morphologically very dissimilar: Romanian is a Romance (weakly inflecting) language, Czech and Slovene Slavic languages, (highly inflecting), and Estonian a Finno-Ugric (agglutinative) language. The ILP system FOIDL [11] is used to learn morphological rules for producing the inflectional forms of nouns and adjectives given the base form (the *lemma*). Rules are also learned for deducing the lemma from these inflectional forms. Thus, rules for both morphological synthesis and analysis are learned.

The training and testing data are taken from the MULTEXT-East tagged corpus [5], and converted to Prolog encoding, as explained in Section 2. FOIDL is used to learn rules for synthesizing and analyzing the noun and adjective forms of the five languages. A brief overview of FOIDL and its application to learning synthesis rules for the past tense of English verbs is given in Section 3. Section 4 describes the MULTEXT-East corpus experiments with FOIDL. Discussed here are the experimental setup, the induced rules for the synthesis and the analysis tasks, and their performance on unseen text. Section 5 concludes with a discussion and some directions for further work.

2 The Data

The EU-funded MULTEXT-East project developed corpora, lexica and tools for six Central and East-European languages; the project reports and samples of results are available at *http://nl.ijs.si/ME/CD*. The centerpiece of the corpus is the novel "1984" by George Orwell, in the English original and translations.

In previous work [4] we reported results on learning the noun paradigms on the MULTEXT-East Slovene-language lexicon, with 200 random entries used for training, and the rest of the lexicon for testing. However, the

Received 15 september 1998
Accepted 10 november 1998

word-forms from a tagged corpus give a more realistic view of the language as compared to random samples from a lexicon. In a purely lexical data-set, comparison between languages is also hampered by differences in the structure of lexicons, these often reflecting the methods used in their construction more than linguistic differences. Such problems are, to a certain extent, neutralised by using validated parallel-corpus data.

For the experiments reported here, the first three parts of the '1984' were taken for training, and the fourth part (Appendix: "The Principles of Newspeak") for testing. The '1984' corpus is encoded in SGML, in the Corpus Encoding Specification DTD [6], with the non-ASCII characters encoded as SGML entities.

For word-forms (<token> elements of <type = word>), the markup in this corpus includes their orthography and annotation for the context disambiguated, as well as ambiguous (i.e. <lex>ical) linguistic annotation. The linguistic annotation contains the lemmas (<base>s) and morphosyntactic descriptions (<msd>s). What is in the tagger literature commonly known as the *ambiguity class* [2] corresponds to the union of a token's <lex> contained <msd> elements.

The example below shows the markup for the Slovene *članki / papers* word-form, taken from the Appendix of the novel:

```
<tok type=WORD>
  <orth>&ccaron;lanki</orth>
  <disamb>
    <base>&ccaron;lanek</base><msd>Ncmpn</msd>
  </disamb>
  <lex>
    <base>&ccaron;lanek</base><msd>Ncmpi</msd>
  </lex>
  <lex>
    <base>&ccaron;lanek</base><msd>Ncmpn</msd>
  </lex>
</tok>
```

The MSDs are structured and more detailed than is commonly assumed for tagsets; they are compact string representations of a simplified kind of feature structures — the formalism was defined in the MULTEXT project [7]. The first letter encodes the part of speech (Noun, Adjective), while the letters following give the value of the position determined attribute. Each part of speech defines its appropriate attributes and their values, acting as a kind of feature-structure type or sort. The above two MSDs expand to the following full descriptions:

>>>>	Ncmpi	>>>>	Ncmpn
PoS:	Noun	PoS:	Noun
Type:	common	Type:	common
Gender:	masculine	Gender:	masculine
Number:	plural	Number:	plural
Case:	instrumental	Case:	nominative

The MSD grammar for the MULTEXT-East languages is defined in the MULTEXT-East tables of attribute values; Appendix I. gives the tables for Nouns and Adjectives, for the five languages that we consider in this paper. The 'x' in the language column defines a certain attribute-value to be appropriate for the language in question. Further constraints are placed on the MSDs of the particular languages by specifying legal combinations of features and, ultimately, by hand-validation of the MSDs actually appearing in the lexicons.

In case a certain attribute is not appropriate (1) for a language, (2) for the particular combination of features, or (3) for the the word in question, this is marked by a hyphen in the attribute's position. Estonian nouns, for example, are not marked for gender and a `Noun common (no gender) singular translative` is written as `Nc-s4`.

For our experiments, triplets were extracted from the tagged corpus, consisting of the <orth> element (the word-form), and the lexical, undisambiguated <base> elements (the possible lemmas) with their accompanying <msd> elements, thus using a setting similar to the setting one obtains prior to tagging. As mentioned, we considered only the triples with noun and adjective MSDs.

Each triplet gives rise to two examples, one for synthesis and one for analysis. The examples have the form `syn_msd(base,orth)` and `ana_msd(orth,base)`. Within the learning setting of inductive logic programming, `syn_msd` and `ana_msd` are relations or predicates, that consist of all pairs (lemma, word-form), resp. (word-form, lemma) that have the same morphosyntactic description. A set of rules has to be learned for each of these predicates or concepts.

Encoding-wise, the MSD's part-of-speech is decapitalised and hyphens are converted to underscores. The word-forms and lemmas are encoded as lists of characters, with non-ASCII characters encoded as SGML entities. In this way, the generated examples comply with Prolog syntax. For illustration, the triplet <orth>članki</orth>, <base>članek</base>, <msd>Ncmpn</msd> gives rise to the following two examples:

```
syn_n0mpn([ccaron,l,a,n,e,k],[ccaron,l,a,n,k,i]).
ana_n0mpn([ccaron,l,a,n,k,i],[ccaron,l,a,n,e,k]).
```

Certain attributes have (almost) no effect on the inflectional behaviour of the word. We generalise over their values in the predicates, and indicate this by a 0 for the value of the vague attribute. In particular, and for all languages, proper and common nouns (Nc, Np) are collapsed to n0, and all Types of adjectives (f, s, o) to a0. Furthermore, only adjectives in the positive degree (a0p) are considered.

Below we give the numbers of generalised MSDs for the complete noun and adjective paradigms for each of the five languages, as well as illustrative examples of MSDs for each language.

```
en:  (1 = 1 N + 0 A)
n00p        any-gender plural

ro (30 = 15 N + 15 A):
n0fpoy      feminine plural oblique +definiteness
a0pmsrn     masculine singular direct -definite.

cs: (91 = 45 N + 46 A)
a0pfdi___c feminine dual instrumental compound
n0fsv       feminine singular vocative
n0mpn__n    masculine plural nominative -animate
n0mpn__y    masculine plural nominative +animate
```

```
sl: (108 = 54 N + 54 A)
a0pfda      positive feminine dual accusative
n0nsg       neuter singular genitive

et (58 = 29 N + 29 A):
a_p_p1      plural partitive
n0_sw       singular essive
```

English has only the plural of nouns to account for in both the Noun and the Adjective paradigms, and only this MSD was used for the dataset. For the other languages complete paradigms were modeled, including the base forms themselves. The languages have a varying numbers of MSDs, depending on their inflectional complexity. Lowest is Romanian (30), with an impoverished case system, similar to other Romance languages, next comes Estonian (58), which has a large number of cases but does not distinguish gender. The two Slavic languages, known for their heavy inflection are highest (Czech 91, Slovene 108), with the large number of Slovene concepts accounted for by the Cartesian product of its three genders, three numbers (unlike Czech, the dual is productive), and six cases.

3 FOIDL and Learning Past Tense

FOIDL [11] is a system for learning relations, i.e., an inductive logic programming (ILP) [9] system. Unlike most other ILP approaches that learn unordered sets of Horn clauses, FOIDL learns first-order decision lists, i.e., ordered lists of clauses. First-order decision lists seem to be a very appropriate formalism for representing linguistic knowledge, as they allow for an elegant way of representing exceptions to general rules. The exceptions are placed before the general rules in a decision list: the first applicable rule from the list is used when treating new cases. The data structure produced by FOIDL thus implements a version of the Elsewhere Condition, well known in morphological theory [8].

Another important feature of FOIDL is its ability to learn decision lists from positive examples only. Most other ILP systems rely on negative examples to avoid overly general hypotheses. FOIDL, on the other hand, uses an output completeness assumption. It states that the training set contains all of the corresponding output patterns for every unique input pattern that it contains. The ability to learn from positive data only is also very important for almost all linguistic applications of ILP.

Like all ILP systems, FOIDL can use background knowledge defining predicates relevant for learning the target predicate. While some ILP systems, e.g., FOIL [12], can only use extensional background knowledge, consisting of ground facts only, FOIDL can use intensional background knowledge consisting of Prolog clauses. Predicates for list processing, relevant for learning linguistic concepts, can thus be represented concisely and efficiently.

FOIDL has been successfully applied to the problem of learning rules for forming the past tense of English verbs [11]. Three different variants of the problem have been addressed: phonetic representation, phonetic representation of regular verbs only, and orthographic representation. For illustration, a positive example for the last case is `past([s,l,e,e,p],[s,l,e,p,t])`. In all three cases, the list processing predicate `split(A,B,C)`, which splits a list `A` into two nonempty lists `B` and `C`, was used as background knowledge. This predicate is defined as:

```
split([X,Y|Z],[X],[Y|Z]).
split([X|Y],[X|Z],W) :- split(Y,Z,W).
```

The first argument (present tense of the verb) of the target predicate `past` is an input argument and the second (past tense) is an output argument. FOIDL was given the opportunity to use constant prefixes and suffixes in its rules. An excerpt from a first-order decision list learned by FOIDL from 250 examples is given below.

```
past(A,B) :- split(A,C,[e,p]),
             split(B,C,[p,t]), !.
...
past(A,B) :- split(B,A,[d]),
             split(A,C,[e]), !.
past(A,B) :- split(B,A,[e,d]).
```

Note the ! (cut) in the clauses: this indicates that only the first applicable clause from the list should be used when treating new examples. The FOIDL output, however, does not list the cuts; the FOIDL output in the next Section is shown in its original form, thus without cuts. When treating new cases, it is interpreted as a decision list, i.e., as if the cuts were there.

Given 1392 examples of the English past tense, FOIDL used up to 500 examples for learning and the remainder for testing. Given 500 examples for learning, FOIDL achieved an accuracy of approximately 85% on the testing set. The running time on a SUN SPARC 10 was approximately 8 hours.

4 Experiments and Results

Our decision to use FOIDL for learning the five-language declensions was based on its properties listed in the previous Section and its success on learning the paradigms for Slovene nouns [4]. While the original English past tense experiment with FOIDL involved only synthesis, it is, in general, more useful to analyze word-forms (determine their lemma) than synthesize them. Thus, two sets of experiment were performed with FOIDL, the first concerning synthesis and the second analysis.

For each MSD, a set of rules for the predicate `syn_msd` was induced: the induced rules generate the oblique form from a given lemma. The input and output arguments of the `syn_msd` predicate are switched for the `ana_msd` predicate: the task of FOIDL was to learn rules that produce the base form of the word given the oblique form. Apart from exchanging the input and the output, the set-up for the synthesis and analysis experiments was identical. There were altogether 288 MSDs in the five languages and FOIDL was run 288 times for synthesis and analysis each, i.e., twice for each MSD.

As has been mentioned, the training sets were taken from the first three parts of '1984'. Due to FOIDL's computational efficiency limits and the large number

of relations to be induced, the 200 (most frequent) examples were chosen for training for each MSD, where more than 200 were available. The Appendix of the novel was used for the test set. While the whole '1984' has approx. 100.000 words, the Appendix has only approx. 4.000. It therefore happens that certain rare MSDs were not represented in the test set (6 for Romanian, 8 for Slovene).

The set-up for the experiment was as for the orthographic past tense learning experiment: for synthesis the training data were encoded as Prolog facts of the form `syn_msd(lemma,oblique)` and for analysis as Prolog facts of the form `ana_msd(oblique,lemma)`. In both cases, the first argument of each target predicate is an input argument and the second is an output argument. The predicate `split` was used as background knowledge. Constant prefixes and suffixes were allowed in the rules.

4.1 Results

For a start, let us take a look at an example set of rules induced by FOIDL for the task of synthesising the genitive singular of Slovene feminine nouns. The rule set consists of five exceptions and three generalizations and is listed below.

```
syn_nOfsg([r,a,v,a,n],[r,a,v,n,i]).
syn_nOfsg([p,e,s,e,m],[p,e,s,m,i]).
syn_nOfsg([o,b,u,t,e,v],[o,b,u,t,v,e]).
syn_nOfsg([m,i,s,e,l],[m,i,s,l,i]).
syn_nOfsg([m,a,t,i],[m,a,t,e,r,e]).
syn_nOfsg(A,B) :- split(B,C,[n,i]),
                  split(A,C,[e,n]).
syn_nOfsg(A,B) :- split(B,C,[e]),
                  split(A,C,[a]).
syn_nOfsg(A,B) :- split(B,A,[i]).
```

From the bottom up, the first rule describes the formation of genitive for feminine nouns of the canonical second declension where *i* is added to the nominative singular (lemma) to obtain the genitive. The second rule deals with the canonical first declension, where the lemma ending *a* is replaced by *e* to obtain the genitive. Finally, the third rule deals with nouns of the second declension that exhibit a common phonological alteration in Slovene, whereby a schwa in the last syllable is deleted in word-forms with a non-null ending (see also below).

The 288 programs learned by FOIDL for the synthesis and analysis concepts show varying degrees of success in capturing the relevant morphological generalizations. As has been already mentioned, FOIDL works only with the orthographic representation of the words, which does not contain enough information to induce the correct rules for synthesizing or analyzing the paradigm forms in all cases.

Table 1 gives an overview of the results obtained by testing the induced programs. For each language, the results on the synthesis and analysis tasks are listed, first for nouns and adjectives separately (N* and A*), then aggregated. The first entry is the total number of generalised MSDs (predicates) over which the summary has been done. The second entry is the total number of clauses in the FOIDL decision list and the number of generalisations that appear in it, e.g. 21/6. This is an estimate of the complexity of the rules induced by FOIDL. The induced rules for each MSD were applied to the testing cases of that MSD and the incorrectly predicted cases recorded. The third entry gives the percentage of correctly predicted cases in the testing set.

Given that FOIDL generates and analyses forms without the support of a lexicon and without any additional constraints, the results are relatively good. As can be seen, there is no systematic relationship between the synthesis and analysis results. The differences are in general due to the different informativeness of the base form compared to the oblique form. It should also be noted that the number of MSDs in a language does not seem connected to the morphological complexity of a particular MSD in the language.

The results significantly reflect the morphological make-up of the languages; the worst scores are obtained for the agglutinative Estonian, where noun synthesis bottoms out with 73%. Given the nature of the language, this is not very surprising. Namely, determining the correct paradigm (and thus the endings) for Estonian can seldomly be predicted on the basis of the form of the word itself; to illustrate, we give below a classical example* (the nouns are in singular, with the nominative being the lemma form):

	nominative	genitive	partitive
(*wolf*)	*susi*	*soe*	*sutt*
(*kiss*)	*musi*	*musi*	*musi*
(*piss*)	*kusi*	*kuse*	*kust*

Next comes Romanian, with approximately 1 in 10 words in the test set being analysed/synthesised incorrectly. For English, the accuracy is three percentage points higher. Especially for Romanian, the accuracy is high enough for the system to make an inexpensive practical alternative to (possibly unavailable) hand-crafted morphological systems.

Unintuitively, applying FOIDL to the Slavic languages yields even better results, with Czech showing the best overall (97%) performance. One reason for the good overall results is that a large part of the inflectional complexity comes from the large numbers of inflectional forms that these languages distinguish. In the set-up discussed here, with the MSD being a part of the predetermined input, this does not have any adverse effects on the results.

With Slovene, the low accuracy on Adjectives, especially their synthesis, comes as a surprise. Analysing the results illuminates the disadvantages of using FOIDL on purely orthographic representations. Namely, almost all errors in the adjective set are due to a productive morpho-phonological alternation in Slovene, the schwa elision. If a schwa (weak *-e-*) appears in the last syllable of the word when it has the *-0* ending, this schwa is dropped with non-null endings: `Ncmsn`: *zaimek-0*, but `Ncmpn`: *zaimk-i*. However, the stem-final *-e-* need not be

*Heiki-Jaan Kaalep, personal communication.

Language		Synthesis			Analysis		
	PoS	MSDs	Rules	Acc	MSDs	Rules	Acc
en	N*	1	21/6	94.1%	1	22/5	94.0%
	A*	0	0/0	/	0	0/0	/
	*	1	21/6	94.1%	1	22/5	94.0%
ro	N*	15	298/99	92%	15	422/144	86.9%
	A*	15	91/32	95.9%	15	101/27	94.9%
	*	30	389/131	93.0%	30	523/171	88.9%
cs	N*	45	865/326	93.8%	45	1054/375	92.5%
	A*	46	144/76	99.0%	46	178/96	98.9%
	*	91	1009/402	96.6%	91	1232/471	96.0%
sl	N*	54	819/323	95.9%	54	1014/404	95.0%
	A*	54	1675/877	77.5%	54	1530/822	90.2%
	*	108	2494/1200	85.5%	108	2544/1226	92.3%
et	N*	29	1463/396	73.2%	29	1235/376	76.3%
	A*	29	512/181	88.5%	29	520/223	88.0%
	*	58	1975/577	78.2%	58	1755/599	80.1%

Table 1. Accuracy and complexity of FOIDL rules

a schwa, in which case it is not deleted: `Ncmsn`: *primer-0*, but `Ncmpn`: *primer-i*.

In such cases FOIDL will, in effect, be guessing whether an *-e-* should be deleted or not, and will base its decisions on the vagaries of the training set. As it cannot simply delete the stem-final 'e', given that certain words retain it, for cases where *-e-* elisions happens, FOIDL extends the left context of the input string until it finds no more counterexamples in the training set, for example:

```
syn_aOpnpn(A,B) :-
        split(A,C,[e,l]),
        split(B,C,[l,a]),
        split(A,D,[s,e,l]).
syn_aOpnpn(A,B) :-
        split(A,C,[e,n]),
        split(B,C,[n,a]),
        split(A,D,[scaron,e,n]),
        split(A,[s],E).
```

Such an approach does not effectively help, as no amount of left context will be able to predict if the 'e' is a schwa and should be elided. While this alternation occurs in all inflections, the reason it severely degrades the performance with adjectives is that a large proportion of the (test set) adjectives are derived from nouns or verbs and exhibit the adjective forming suffixes *-en* and *-el*. Finally, the reason that the error is more severe in the synthesis direction is that while it is difficult to predict *-e-* elision, *-e-* epenthesis is more straightforward, as two consecutive consonants are very seldom allowed at the end of the word.

5 Discussion

We have presented the results of learning rules for synthesizing and analyzing inflectional forms of nouns and adjectives for English, Romanian, Czech, Slovene, and Estonian, utilising the FOIDL system. Taking into account that FOIDL was given very limited background knowledge, the results obtained are, except for the agglutinative Estonian, quite satisfactory, esp. for the Slavic languages.

The errors are in part due to the insufficient information available to FOIDL. First, as has been discussed with the Slovene adjectives, the orthography of a word is sometimes not enough to predict whether a certain phonologically determined alternation should take place or not. To cover such cases, a phonological representation has to be substituted for, or added to the orthographic one. Furthermore, the background knowledge of FOIDL could in such a set-up be extended to take phonological regularities into account, by e.g., distinguishing vowels from consonants etc., thus leading to better generalisations.

Second, FOIDL does not have information on purely morphological features of a word, for example its paradigm class, i.e. declension. This to some extent lowers the results for the Slavic languages and significantly degrades performance with Estonian.

Several other directions for further work can be pointed out. As regards the induction methodology, at least two improvements of FOIDL seem to be needed. Efficiency seems to be a major problem, effectively limiting the size of training sets that can be considered to approximately 200 examples. Post-processing of the induced decision lists is also needed in order to remove irrelevant literals.

The induced rules act as an alternative to hand-crafted morphological systems, which are often hard to obtain and difficult to produce. The rules act as constraints on the relationship between the word-form, its lemma, and

its morphosyntactic description. The most obvious application is in combination with a tagger, as the analysis rule system can, without a lexicon, determine the ambiguity class of a word-form. Additionally, the analysis system can be used as a 'stemmer', or more accurately, lemmatiser, with a tagger first constraining the set of valid MSD interpretations of the word-form. If the MSD is uniquely determined, then the lemmatisation accuracy for the languages is identical to the analysis accuracies given in Table 1.

As an example, we tried analysing the Slovene word-form *golobu*, with the MSD being any of the 108 possible concepts of Slovene. This word was not a member of the training set, has one lemma (*golob/pigeon*) and its ambiguity class contains two MSDs, `Ncmsd` and `Ncmsl`. With the induced analysis rules FOIDL proposes only 10 concepts containing 3 different lemmatizations. Purely on induced morphological grounds, 90% of the possible concepts are thus eliminated. Introducing a simple phonological constraint of Slovene, which forbids a word ending in two vowels, the hypotheses are reduced to 8 concepts with two lemmatizations. If, finally, a lexicon of base forms constraints the lemma output of analysis, three concepts are left, two of which are correct.

In summary, we have successfully applied the ILP system FOIDL to learn rules for synthesis and analysis of nominal forms of five languages. Further work will focus on improving the induced rules by using additional linguistic background knowledge and using the improved rules to perform preliminary analysis of word forms appearing in corpora, producing input for further text processing, e.g., part-of-speech tagging.

Acknowledgements

This work was supported in part by the projects ESPRIT IV 20237 ILP2 and by Copernicus Cop 106 MULTEXT-East.

6 References

[1] J. Cussens. Part-of-speech tagging using Progol. In *Proceedings of the 6th International Workshop on Inductive Logic Programming*, pages 93–108, Berlin, 1997. Springer.

[2] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 133–140, Trento, Italy, 1992.

[3] W. Daelemans, T. Weijters, and A. van den Bosch, editors. *Empirical Learning of Natural Language Processing Tasks, ECML-97 MLnet Workshop Notes*, Prague, Czech Republic, 1997.

[4] S. Džeroski and T. Erjavec. Induction of slovene nominal paradigms. In N. Lavrač and S. Džeroski, editors, *Inductive Logic Programming; 7th International Workshop ILP-97, Proceedings*, pages 141–148. Springer, 1997.

[5] T. Erjavec and N. Ide. The MULTEXT-East corpus. In *Proceedings of the First International Conference on Language Resources and Evaluation, LREC'98*, pages 971–974, Granada, 1998. ELRA.

[6] N. Ide. Corpus Encoding Standard: SGML guidelines for encoding linguistic corpora. In *Proceedings of the First International Conference on Language Resources and Evaluation, LREC'98*, pages 463–470, Granada, 1998. ELRA.

[7] N. Ide, D. Tufiş, and T. Erjavec. Development and Assessment of Common Lexical Specifications for Six Central and Eastern European Languages. In *Proceedings of the First International Conference on Language Resources and Evaluation, LREC'98*, pages 233–240, Granada, 1998. ELRA.

[8] P. Kiparsky. "Elsewhere" in phonology. In Steven R. Anderson, editor, *Festschrift for Morris Halle*, pages 93–106. Holt, Rinehart and Winston, New York, 1973.

[9] N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, Chichester, 1994.

[10] R. J. Mooney. Inductive logic programming for natural language processing. In *Proceedings of the 6th International Workshop on Inductive Logic Programming*, pages 3–22, Berlin, 1997. Springer.

[11] R. J. Mooney and M. E. Califf. Induction of first-order decision lists: Results on learning the past tense of English verbs. *Journal of Artificial Intelligence Research*, (3):1–24, 1995.

[12] J.R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266, 1990.

[13] S. Wermter, E. Riloff, and G. Scheller, editors. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*. Springer, Berlin, 1996.

Appendix I. MULTEXT-East Morphosyntactic Tables for Nouns & Adjectives

Noun (N)

=	=========	==============	=	EN	RO	SL	CS	ET
P	ATT	VAL	C	x	x	x	x	x
=	=========	==============	=					
1	Type	common	c	x	x	x	x	x
		proper	p	x	x	x	x	x
-	---------	--------------	-					
2	Gender	masculine	m	x	x	x	x	
		feminine	f	x	x	x	x	
		neuter	n	x	x	x	x	
-	---------	--------------	-					
3	Number	singular	s	x	x	x	x	x
		plural	p	x	x	x	x	x
		dual	d			x	x	
-	---------	--------------	-					
4	Case	nominative	n			x	x	x
		genitive	g			x	x	x
		dative	d			x	x	
		accusative	a			x	x	
		vocative	v		x		x	
		locative	l			x	x	
		instrumental	i			x	x	
		direct	r		x			
		oblique	o		x			
		partitive	1					x
		illative	x					x
		inessive	2					x
		elative	e					x
		allative	t					x
		adessive	3					x
		ablative	b					x
		translative	4					x
		terminative	9					x
		essive	w					x
		abessive	5					x
		komitative	k					x
		aditive	7					x
*	**********	**********	*					
5	Definitne	no	n		x			
		yes	y		x			

6	Clitic	no	n		x		
		yes	y		x		
7	Animate	no	n			x	
		yes	y			x	
				EN	RO	SL	CS	ET

Adjective (A)

P	ATT	VAL	C	EN	RO	SL	CS	ET
1	Type	qualificative	f	x	x	x	x	
		possessive	s			x	x	
		ordinal	o			x		
2	Degree	positive	p	x	x	x	x	x
		comparative	c	x	x	x	x	x
		superlative	s	x	x	x	x	x
3	Gender	masculine	m		x	x	x	
		feminine	f		x	x	x	
		neuter	n		x	x	x	
4	Number	singular	s		x	x	x	x
		plural	p		x	x	x	x
		dual	d			x	x	
5	Case	nominative	n			x	x	x
		genitive	g			x	x	x
		dative	d			x	x	
		accusative	a			x	x	
		vocative	v		x		x	
		locative	l			x	x	
		instrumental	i			x	x	
		direct	r		x			
		oblique	o		x			
		partitive	1					x
		illative	x					x
		inessive	2					x
		elative	e					x
		allative	t					x
		adessive	3					x
		ablative	b					x
		translative	4					x
		terminative	9					x
		essive	w					x
		abessive	5					x
		komitative	k					x
		aditive	7					x
6	Definite	no	n		x			
		yes	y		x			
7	Clitic	no	n		x			
		yes	y		x			
8	Animate	no	n				x	
		yes	y				x	
9	Format.	nominal	n				x	
		compound	c				x	
				EN	RO	SL	CS	ET

Sašo Džeroski is a research associate at the Department of Intelligent Systems, J. Stefan Institute, Ljubljana, Slovenia (since 1989). He has held visiting researcher positions at the Turing Institute, Glasgow (UK), Katholieke Universiteit Leuven (Belgium), German National Research Center for Computer Science (GMD), Sankt Augustin (Germany) and the Foundation for Research and Technology-Hellas (FORTH), Heraklion (Greece). His research interest is in machine learning and knowledge discovery in databases, in particular inductive logic programming and its applications and knowledge discovery in environmental databases. He is co-author of Inductive Logic Programming: Techniques and Applications, Ellis Horwood 1994. He is the scientific coordinator of ILPnet2, The Network of Excellence in Inductive Logic Programming. He was program co-chair of the 7th International Workshop on Inductive Logic Programming ILP'97 and will be program co-chair of the 16th International Conference on Machine Learning ICML'99.

Tomaž Erjavec received his B.Sc., M.Sc. and Ph.D. degrees in 1984, 1990, and 1997 from the Faculty of (Electrical Engineering and) Computer Science, University of Ljubljana, and an M.Sc. from the Dept. for Cognitive Science, University of Edinburgh in 1992. He works at the Institute Jožef Stefan since 1984, now at the Dept. for Intelligent Systems. His research interest are in the field of computational linguistics and language technologies, especially computational morphology and the development of Slovene language resources. He is currently involved in the projects FIDA: a Slovene Reference Corpus and IMI (Development of Digital Publishing with Distance Learning Support) and in EU projects CONCEDE (Consortium for Central European Dictionary Encoding) and TELRI-II (Trans European Lamguage Resources Infrastructure II) and was the organiser of the conference "Language Technologies for the Slovene Language", Ljubljana, October '98.