Learning to Lemmatise Slovene Words

Sašo Džeroski and Tomaž Erjavec

Department for Intelligent Systems,
Jožef Stefan Institute,
Jamova 39, SI-1000 Ljubljana, Slovenia
saso.dzeroski@ijs.si, tomaz.erjavec@ijs.si

Abstract. Automatic lemmatisation is a core application for many language processing tasks. In inflectionally rich languages, such as Slovene, assigning the correct lemma to each word in a running text is not trivial: nouns and adjectives, for instance, inflect for number and case, with a complex configuration of endings and stem modifications. The problem is especially difficult for unknown words, as word forms cannot be matched against a lexicon giving the correct lemma, its part-of-speech and paradigm class.

The paper discusses a machine learning approach to the automatic lemmatisation of unknown words, in particular nouns and adjectives, in Slovene texts. We decompose the problem of learning to perform lemmatisation into two subproblems: the first is to learn to perform morphosyntactic tagging, and the second is to learn to perform morphological analysis, which produces the lemma from the word form given the correct morphosyntactic tag. A statistics-based trigram tagger is used to learn to perform morphosyntactic tagging and a first-order decision list learning system is used to learn rules for morphological analysis.

The dataset used is the 90.000 word Slovene translation of Orwell's '1984', split into a training and validation set. The validation set is the Appendix of the novel, on which extensive testing of the two components, singly and in combination, is performed. The trained model is then used on an open-domain testing set, which has 25.000 words, pre-annotated with their word lemmas. Here 13.000 nouns or adjective tokens are previously unseen cases. Tested on these unknown words, our method achieves an accuracy of 81% on the lemmatisation task.

1 Introduction

Lemmatisation is a core functionality for various language processing tasks. It represents a normalisation step on the textual data, where all inflected forms of a lexical word are reduced to its common lemma. This normalisation step is needed in analysing the lexical content of texts, e.g. in information retrieval, term extraction, machine translation etc.

In English, lemmatisation is relatively easy, especially if we are not interested in the part-of-speech of a word. So called stemming can be performed with a

J. Cussens and S. Džeroski (Eds.): LLL'99, LNAI 1925, pp. 69–88, 2000.

lexicon which lists the irregular forms of inflecting words, e.g. ‘oxen’ or ‘took’, while the productive ones, e.g. ‘wolves’ or ‘walks’, can be covered by a small set of suffix stripping rules. The problem is more complex for inflectionally rich languages, such as Slovene.

Lemmatisation in inflectionally rich languages must pressupose correctly determining the part-of-speech together with various morphosyntactic features of the word form. Adjectives in Slovene, for example, inflect for gender (3), number (3) and case (6), and in some instances, also for definiteness and animacy. This, coupled with various morpho-phonologically induced stem and ending alternations gives rise to a multitude of possible relations between a word form and its lemma. A typical Slovene adjective has, for example, 14 different orthographic inflected forms, and a noun 8.

It should be noted that we take the term ‘lemma’ to mean a word form in its canonical form, e.g., infinitive for verbs, nominative singular for regular nouns, nominative plural for pluralia tantum nouns, etc. The orthography of what we call a ’lemma’ and of the ‘stem’ of a word form are, in general, different, but much less in English than in Slovene. For example, the feminine noun ‘postelja’/’bed’, has ‘postelja’ as its lemma and this will be also its headword in a dictionary. However, the stem is ‘postelj-’, as the ‘-a’ is already the inflectional morpheme for (some) feminine nouns. Performing lemmatisation thus in effect involves performing morphological analysis, to identify the ending and isolate the stem, and synthesis, to join the canonical ending to it.

Using a large lexicon with coded paradigmatic information, it is possible to reliably, but ambiguously lemmatise known words. Unambiguous lemmatisation of words in running text is only possible if the text has been tagged with morphosyntactic information, a task typically performed by a part-of-speech tagger. Much more challenging is the lemmatisation of unknown words. In this task, known as ‘unknown word guessing’ a morphological analyser can either try to determine the ambiguity class of the word, i.e. all its possible tags (and stems), which are then passed on to a POS tagger, or it can work in tandem with a tagger to directly determine the context dependent unambiguous lemma.

While results on open texts are quite good with hand-crafted rules (Chanod & Tapanainen, 1995), there has been less work done with automatic induction of unknown word guessers. Probably the best known system of this kind is described in (Mikheev, 1997). It learns ambiguity classes from a lexicon and a raw (untagged) corpus. It induces rules for prefixes, suffixes and endings: the paper gives detailed analysis of accuracies achieved by combining these rules with various taggers. The best results obtained for tagging unknown words are in the range of 88%. However, the tests are performed on English language corpora and it is unclear what the performance as applied to lemmatisation would be with inflectionally richer languages.

In this article, we discuss a machine learning approach to the automatic lemmatisation of unknown words in Slovene texts. We decompose the problem of learning to perform lemmatisation into two subproblems. The first is to learn rules for morphological analysis, which produce the lemma from the word form given the correct tag in the form of a morphosyntactic description (MSD). The second is to learn to perform tagging, where tags are MSDs.

We use an existing annotated/disambiguated corpus to learn and validate rules for morphological analysis and tagging. A first-order decision list learning system, Clog (Manandhar, Džeroski, & Erjavec, 1998) is used to learn rules for morphological analysis. These rules are limited to nouns and adjectives, as these are, of the inflectional words, by far the most common new (unknown) words of a language. A statistics-based trigram tagger, TnT (Brants, 2000) is used to learn to perform MSD tagging. Once we have trained the morphological analyser and the tagger, unknown word forms in a new text can be lemmatised by first tagging the text, then giving the word forms and corresponding MSDs to the morphological analyser.

The remainder of the article is organised as follows. Section 2 describes the corpus used to inductively develop the morphological analyser and the tagger. This was the 90.000 word Slovene Multext-East annotated corpus, which was divided into a larger training set and a smaller validation set. Section 3 describes the process of learning rules for morphological analysis, including an evaluation of the learned rules on the validation set. Similarly, Section 4 describes the process of training the tagger, including an evaluation of the learned tagger on the validation set. Section 5 describes the evaluation of the lemmatisation performed by the combination of the learned tagger and morphological analyser on the validation set and the testing set. The validation set on which we perform a detailed analysis is the one from the Multext-East corpus; we also evaluate the results on a text of 25.000 words from a completely different domain, pre-annotated with the word lemmas. Finally, Section 5 concludes and discusses directions for further work.

2 The Training and Validation Data Sets

The EU Multext-East project (?; Erjavec, Lawson, & Romary, 1998) 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/*. The centrepiece of the corpus is the novel "1984" by George Orwell, in the English original and translations. For the experiment reported here, we used the annotated Slovene translation of "1984". This corpus has been further cleaned up and re-encoded within the scope of the EU project ELAN (Erjavec, 1999).

The novel is sentence segmented (6,689 sentences) and tokenised (112,790) into words (90,792) and punctuation symbols (21,998). Each word in the corpus is annotated for context disambiguated linguistic annotation. This annotation contains the lemma and morphosyntactic descriptions (MSD) of the word in question. The corpus is encoded according to the recommendation of the Text Encoding Initiative, TEI (Sperberg-McQueen & Burnard, 1994). To illustrate the information contained in the corpus, we give the encoding of an example sentence in Table 1.

Table 1. The TEI encoding of the sentence ‘Winston se je napotil proti stopnicam.’ (’Winston made for the stairs.’)

```
<s id="Osl.1.2.3.4">
<w lemma="Winston" msd="Npmsn">Winston</w>
<w lemma="se" msd="Px------y">se</w>
<w lemma="biti" msd="Vcip3s--n">je</w>
<w lemma="napotiti" msd="Vmps-sma">napotil</w>
<w lemma="proti" msd="Spsd">proti</w>
<w lemma="stopnica" msd="Ncfpd">stopnicam</w>
<c>.</c>
</s>
```

The MSDs are structured and more detailed than is commonly assumed for part-of-speech tags; they are compact string representations of a simplified kind of feature structures — the formalism and MSD grammar for the Multext-East languages is defined in (Erjavec & (eds.), 1997). The first letter of a MSD encodes the part of speech (`Noun`, `Adjective`); Slovene distinguishes 11 different parts-of-speech. The letters following the PoS give the values of the position determined attributes. Each part of speech defines its own appropriate attributes and their values, acting as a kind of feature-structure type or sort. So, for example, the MSD `Ncmpi` expands to `PoS:Noun, Type:common, Gender:masculine, Number:plural, Case:instrumental`. It should be noted that in case a certain attribute is not appropriate (1) for a language, (2) for the particular combination of features, or (3) for the word in question, this is marked by a hyphen in the attribute's position. Slovene verbs in the indicative, for example, are not marked for gender or voice, hence the two hyphens in `Vcip3s--n`.

For the experiment reported here, we first converted the TEI encoded novel into a simpler, tabular encoding. Here each sentence ends with an empty line, and all the words and lemmas are in lower-case. This simplifies the training and testing regime, and, arguably, also leads to better results as otherwise capitalised words are treated as distinct lexical entries. The example sentence from Table 1 converts to the representation in Table 2.

Table 2. A tabular encoding of the sentence ‘Winston se je napotil proti stopnicam.’ (‘Winston made for the stairs.’)

winston	winston	Npmsn
se	se	Px------y
je	biti	Vcip3s--n
napotil	napotiti	Vmps-sma
proti	proti	Spsd
stopnicam	stopnica	Ncfpd
.		.

To give an impression of the size and complexity of the dataset we give in Table 3 the distribution over part-of-speech for the disambiguated Slovene words in “1984”. The first column in the Table gives the number of word tokens, the second of word types, i.e. of different word forms appearing in the corpus. The third column gives the number of different lemmas in the corpus and the fourth the number of different MSDs. The last column is especially interesting for lemmatisation, as it gives the number of tokens that are identical to their lemmas; these represent the trivial cases for lemmatisation. As can be seen, approx. 38% of noun tokens and 16% of adjective tokens are already in their lemma form. This serves as a useful baseline against which to compare analysis results.

Table 3. Part-of-speech distribution of the words in the ‘1984’ corpus.

Category		Token	Type	Lemma	MSD	=
Verb	(V)	25163	4883	2003	93	1405
Noun	(N)	19398	6282	3199	74	7408
Pronoun	(P)	10861	373	64	581	4111
Conjunction	(C)	8554	32	32	2	8554
Preposition	(S)	7991	86	82	6	7987
Adjective	(A)	7717	4063	1943	167	1207
Adverb	(R)	6681	790	786	3	4479
Particle	(Q)	3237	41	41	1	3237
Numeral	(M)	1082	193	112	80	511
Abbreviation	(Y)	60	14	14	1	60
Interjection	(I)	47	7	7	1	47
Residual	(X)	1	1	1	1	1
Total	(*)	90792	16401	7902	1010	39007

The Slovene Orwell also exists in a format that contains all the possible interpretations (MSDs, lemmas) for each word form in the corpus. This version was also used in the experiment, to train the morphological analyser and to determine the unknown words in the validation set; we return to this issue below.

We took Parts I – III of “1984” as the training set, and the Appendix of the novel, comprising approx. 15% of the text, as the validation set. It should be noted that the Appendix, entitled “The Principles of Newspeak” has quite a different structure and vocabulary than the body of the book; it therefore represents a rather difficult validation set, even though it comes from the same text as the training part.

The main emphasis of the experiments we performed is on the Slovene nouns and adjectives in the positive degree. The reason for this is that nouns and adjectives represent the majority of unknown words; the other parts of speech are either closed, i.e. can be exhaustively listed in the lexicon, or, bar verbs, do not inflect. The reason for limiting the adjectives degree to positive only is similar: adjectives that form the other two degrees (comparative and superlative) also represent a closed class of words.

To set the context, we give in Table 4 the distribution of nouns and positive adjectives in the dataset and in its training and validation parts, with the meaning of the columns being the same as in Table 3.

Table 4. The distribution of nouns and adjectives in the entire dataset, the training and the validation set.

Source	Category	Token	Type	Lemma	MSD	=
Entire dataset	Noun (N)	19398	6282	3199	74	7408
	Adjective (A)	7462	3932	1936	121	1207
	Both (*)	26860	10214	5135	195	8615
Training set	Noun (N)	18438	6043	3079	74	7049
	Adjective (A)	7019	3731	1858	120	1124
	Both (*)	25457	9774	4937	194	8173
Validation set	Noun (N)	960	533	379	51	359
	Adjective (A)	443	347	245	62	83
	Both (*)	1403	880	624	113	442

2.1 The Lexical Training Set

As was mentioned, the training set for morphological analysis was not the disambiguated body of the book, but rather its undisambiguated, lexical version, in which each word form is annotated with all its possible MSDs and lemmas. This represents a setting in which lexical look-up has been performed, but the text has not yet been tagged, i.e. disambiguated. The lexical training set thus contains more MSDs and lemmas per word form than does the disambiguated corpus. For a comparison with the disambiguated corpus data, we give in Table 5 the quantities for nouns and adjectives in the lexical training set.

Table 5. The distribution of nouns and adjectives in the lexical training set.

Category	Entry	WordF	Lemma	MSD
Noun (N)	15917	6596	3382	85
Adjective (A)	24346	4796	2356	157
Both (*)	40263	11392	5738	242

The first column in the Table gives the number of different triplets of word form, lemma and MSD; the second column represents the number of different word forms in the lexical training set, the third the number of different lemmas and the fourth the number of MSDs. We can see that, on the average, a lemma has two different word forms, that a noun word form is 2.4 times ambiguous, while adjectives are 5 times ambiguous.

2.2 Unknown Words

As our experiments centre around unknown words, this notion also has to be defined: we take as unknown those nouns and adjectives that appear in the validation corpus, but whose lemma does not appear in the lexical training set. It should be noted that this excludes 'half-unknown' words, which do share a lemma, but not a word form token. With this strict criterion, Table 6 gives the numbers for the unknown nouns and positive adjectives in the Appendix.

Table 6. The distribution of unknown nouns and adjectives in the validation set.

Category	Token	Type	Lemma	MSD	=
Noun (N)	187	144	127	37	85
Adjective (A)	92	82	72	31	26
Both (*)	279	226	199	68	111

3 Morphological Analysis

This section describes how the lexical training set was used to learn rules for morphological analysis of Slovene nouns and adjectives. For this purpose, we used an inductive logic programming (ILP) system that learns first-order decision lists, i.e. ordered sets of rules. We first explain the notion of first-order decision lists on the problem of synthesis of the past tense of English verbs, one of the first examples of learning morphology with ILP (Mooney & Califf, 1995). We then lay out the ILP formulation of the problem of learning rules for morphological analysis of Slovene nouns and adjectives and describe how it was addressed with the ILP system Clog. The induction results are illustrated for an example MSD. We finally discuss the evaluation of the learned rules on the evaluation set.

3.1 Learning Decision Lists

The ILP formulation of the problem of learning rules for the synthesis of past tense of English verbs considered in (Mooney & Califf, 1995) is as follows. A logic program has to be learned defining the relation `past(PresentVerb,PastVerb)`, where `PresentVerb` is an orthographic representation of the present tense form of a verb and `PastVerb` is an orthographic representation of its past tense form. `PresentVerb` is the input and `PastVerb` the output argument. Given are examples of input/output pairs, such as `past([b,a,r,k],[b,a,r,k,e,d]))]` and `past([g,o],[w,e,n,t])`. The program for the relation `past` uses the predicate split(A,B,C) as background knowledge: this predicate splits a list (of letters) A into two lists B and C.

Given examples and background knowledge, Foidl (Mooney & Califf, 1995) learns a decision list defining the predicate past. A decision list is an ordered set of rules: rules at the beginning of the list take precedence over rules below them and can be thought of as exceptions to the latter. An example decision list defining the predicate past is given in Table 7.

Table 7. A first-order decision list for the synthesis of past tense of English verbs.

```
past([g,o],[w,e,n,t]) :- !.
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]).
```

The general rule for forming past tense is to add the suffix '-ed' to the present tense form, as specified by the default rule (last rule in the list). Exceptions to these are verbs ending on '-e', such as 'skate', where '-d' is appended, and verbs ending in '-ep', such as 'sleep', where the ending '-ep' is replaced with '-pt'. These rules for past tense formation are specified as exceptions to the general rule, appearing before it in the decision list. The first rule in the decision list specifies the most specific exception: the past tense form of the irregular verb 'go' is 'went'.

Our approach is to induce rules for morphological analysis in the form of decision lists. To this end, we use the ILP system Clog (Manandhar et al., 1998). Clog shares a fair amount of similarity with Foidl (Mooney & Califf, 1995): both can learn first-order decision lists from positive examples only — an important consideration in NLP applications. Clog inherits the notion of *output completeness* from Foidl to generate implicit negative examples (see (Mooney & Califf, 1995)). Output completeness is a form of closed world assumption which assumes that all correct outputs are given for each given combination of input arguments' values present in the training set. Experiments show that Clog is significantly more efficient than Foidl in the induction process. This enables Clog to be trained on more realistic datasets, and therefore to attain higher accuracy.

3.2 Learning Rules for Morphological Analysis

We formulate the problem of learning rules for morphological analysis of Slovene nouns and adjectives in a similar fashion to the problem of learning the synthesis of past tense of English verbs.

We have used Clog earlier to generate rules for synthesis and analysis of nouns and adjectives for English, Romanian, Czech, Slovene, and Estonian (Manandhar et al., 1998). In the current experiment, we re-use the rules learned for the analysis of Slovene nouns and adjectives.

Triplets are extracted from the training corpus, consisting of the word form itself, and the lexical, undisambiguated lemmas with their accompanying MSDs, thus using a setting similar to the one prior to tagging. The lexical training set is used to obtain the word forms and their undisambiguated lemmas and MSDs. Each triplet is an example of analysis of the form `msd(orth,lemma)`. Within the learning setting of inductive logic programming, `msd(Orth,Lemma)` is a relation or predicate, that consist of all pairs (word form, lemma) that have the same morphosyntactic description. `Orth` is the input and `Lemma` the output argument. A set of rules has to be learned for each of the `msd` predicates.

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 *članki*/*članek*/`Ncmpn` gives rise to the following example:

```
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, as seen above for the collapsing of proper and common nouns (`Nc`, `Np`) to `n0`. This gives rise to generalised MSDs, such as `n0mpn` above. For the complete noun and adjective paradigms, where we have all together 242 MSDs, we find that Slovene needs 108 generalised MSDs, 54 for nouns (85 MSDs) and 54 for adjectives (157). Each generalised MSD is a target predicate to be learned. Examples for these 108 predicates are generated from the training lexicon as described above.

Instead of Foidl's predicate `split/3`, the predicate `mate/6` is used as background knowledge in Clog. `mate` generalises `split` to deal also with prefixes, and allows the simultaneous specification of the affixes for both input arguments. As Slovene inflection only concern the endings of words, the prefix arguments will be empty lists, and the form `mate` that will be used corresponds to the following definition:

```
mate(W1,W2,[],[],Y1,Y2) :- split(W1,X,Y1), split(W2,X,Y2).
```

As an example, consider the set of rules induced by Clog for the particular task of analysing the genitive singular of Slovene feminine nouns. The training set for this concept contained 608 examples, from which Clog learned 13 rules of analysis. Nine of these were lexical exceptions, and are not interesting in the context of unknown word lemmatisation. We list the four generalisations in Table 8.

From the bottom up, the first rule describes the formation of genitive for feminine nouns of the canonical first declension, where the lemma ending -*a* is replaced by -*e* to obtain the genitive. The second rule deals with the canonical second declension where *i* is added to the nominative singular (lemma) to obtain

Table 8. A first-order decision list for the analysis of Slovene feminine nouns in the singular genitive declination.

```
n0fsg(A,B):-mate(A,B,[],[],[t,v,e],[t,e,v]),!.
n0fsg(A,B):-mate(A,B,[],[],[e,z,n,i],[e,z,e,n]),!.
n0fsg(A,B):-mate(A,B,[],[],[i],[]),!.
n0fsg(A,B):-mate(A,B,[],[],[e],[a]),!.
```

the genitive. The third rule attempts to cover nouns of the second declension that exhibit a common morpho-phonological alteration in Slovene, the schwa elision. Namely, if a schwa (weak *-e-*) appears in the last syllable of the word when it has the null ending, this schwa is dropped with non-null endings: *bolezen-0*, but *bolezn-i*. Finally, the topmost rule models a similar case with schwa elision, coupled with an ending alternation, which affects only nouns ending in *-ev*.

3.3 Evaluating the Morphological Rules

The rules for morphological analysis learned by Clog were first tested independently of the tagger on the Appendix of the novel '1984'. For each token in the Appendix, the correct (disambiguated) MSD tag is used and the appropriate `msd` predicate is called with the token as an input argument. An error is reported unless the returned output argument is equal to the correct lemma as specified by the '1984' lexicon (of which the training lexicon is as a subset). Table 9 summarises the results.

Table 9. Validation results for the morphological analyser on all words, known and unknown words.

	All		Known		Unknown	
	Acc.	Correct/Err	Acc.	Correct/Err	Acc.	Correct/Err
Nouns	97.5%	936/24	99.1%	766/ 7	90.9%	170/17
Adjectives	97.3%	431/12	96.6%	339/12	100%	92/0
Both	97.4%	1367/36	98.3%	1105/19	93.9%	262/17

It might come as a surprise that the accuracy on known words is not 100%. However, the errors on known words are on word forms that do not appear in the training corpus. Only word forms that appear in the training corpus are used to learn the rules for morphological analysis together with the corresponding undisambiguated sets of MSDs. The training lexicon is used to provide the latter, and not all word forms of the lemmas that appear in the training corpus.

4 Tagging for Morphosyntax

Syntactic wordclass tagging (van Halteren, 1999), often referred to as part-of-speech tagging has been an extremely active research topic in the last decade. Most taggers take a training set, where previously each token (word) had been correctly annotated with its part-of-speech, and learn a model of the language This model enables them to predict the parts-of-speech for words in new texts to a greater or lesser degree.

Some taggers learn the complete necessary model from the training set, while others must make use of background knowledge, in particular a morphological lexicon. The lexicon contains all the possible morphological interpretations of the word forms, i.e. their ambiguity classes. The task of the tagger is to assign the correct interpretation to the word form, taking context into account.

For our experiments, we needed an accurate, fast, flexible and robust tagger that would accommodate the large Slovene morphosyntactic tagset. Importantly, it also had to be able to deal with unknown words, i.e. word forms not encountered in the training set or background lexicon.

In an evaluation exercise (Džeroski, Erjavec, & Zavrel, 1999) we tested several different taggers on the Slovene Orwell corpus. They were: the Hidden Markov Model (HMM) tagger (Cutting, Kupiec, Pedersen, & Sibun, 1992; Steetskamp, 1995), the Rule Based Tagger (RBT) (Brill, 1995), the Maximum Entropy Tagger (MET) (Ratnaparkhi, 1996), and the Memory-Based Tagger (MBT) (Daelemans, Zavrel, Berck, & Gillis, 1996). After this experiment was performed, a new tagger became available, called TnT (Brants, 2000). It works similarly to our original HMM tagger (Steetskamp, 1995) but is a more mature implementation. We therefore substituted TnT for HMM in the evaluation.

We also trained a tagger using the ILP system Progol. On English, this approach attains accuracies comparable to other state-of-the-art taggers (Cussens, 1997). Unambiguous tagging of Slovene data was less satisfactory (Cussens, Džeroski, & Erjavec, 1999) (although the tagger turned out to be a very good validation aid, as it can identify errors of manual tagging). We have thus omitted this tagger from the experimental comparison.

The comparative evaluation of RBT, MET, MBT and TnT was performed by taking the body of '1984' and using 90% of randomly chosen sentences as the training set, and 10% as the validation set. The evaluation took into account all tokens, words as well as punctuation. While (Džeroski et al., 1999) considered several different tagsets, here we use the 'maximal' tagset, where tags are full MSDs.

The results indicate that accuracy is relatively even over all four taggers, at least for known words: the best result was obtained by MBT (93.6%), followed by RBT (92.9%), TnT (92.2%) and MET (91.6%). The differences in tagging accuracies over unknown words are more marked: here TnT leads (67.55%), followed by MET (55.92%), RBT (45.37%), and MBT (44.46%). Apart from accuracy, the question of training and testing speed is also paramount; here RBT was by far the slowest (3 days for training), followed by MET, with MBT and TnT being very fast (both less than 1 minute).

Table 10. Excerpts from the a) n-gram and b) lexicon files generated by the TnT tagger.

a) An excerpt from the n-gram file generated by TnT.

```
Vcps-sma        544
 Vcip3s--n       82
         Afpmsnn 17
         Aopmsn   2
         Ncmsn   12
         Npmsn    1
         Css      2
         Afpnpa   1
         Q        3
...
```

b) An excerpt from the lexicon file generated by TnT.

```
...
juhe       2      Ncfsg       2
julij      1      Npmsn       1
julija    59      Npfsn      58    Npmsa--y  1
julije     4      Npfsg       4
juliji    10      Npfsd      10
julijin    4      Aspmsa--n   2    Aspmsn    2
...
```

Given the above assessment, we chose for our experiment the TnT tagger: it exhibits good accuracy on known words, excellent accuracy on unknown words, is robust and efficient. In addition, it is easy to install and run, and incorporates several methods of smoothing and of handling unknown words.

4.1 Learning the Tagging Model

The disambiguated body of the novel was first converted to TnT training format, identical to our tabular file, but without the lemma; each line contains just the token and the correct tag. For word tags we used their MSDs, while punctuation marks were tagged as themselves. This gives us a tagset of 1024, comprising the sentence boundary, 13 punctuation tags, and the 1010 MSDs.

Training the TnT tagger produces a table of MSD n-grams (n=1,2,3) and a lexicon of word forms together with their frequency annotated ambiguity classes. The n-gram file for our training set contains 1024 uni-, 12293 bi-, and 40802 trigrams, while the lexicon contains 15786 entries. Example stretches from the n-gram and lexicon file are given in Table 10.

The excerpt from the n-gram file can be interpreted as follows. The tag `Vcps-sma` appeared 544 times in the training corpus. It was followed by the tag

Vcip3s--n 82 times. The triplet Vcps-sma, Vcip3s--n, Afpmsnn appeared 17 times.

The excerpt from the lexicon file can be interpreted as follows. The word form juhe appeared in the corpus twice and was tagged Ncfsg in both cases. The word form julijin appeared 4 times and was tagged twice as Aspmsa--n and twice as Aspmsn. The ambiguity class of the word form julijin is thus the tagset {Aspmsa--n,Aspmsn}.

We did not make use of any background lexicon. We left the smoothing parameters of TnT at their default values. Experiments along these lines could well improve the tagging model.

4.2 Evaluating the Tagger

We then tested the performance of the TnT tagger on the Appendix validation set. The results are summarised in Table 11.

Table 11. Validation results for the TnT tagger.

	Accuracy	Correct/Err
All tokens	83.7%	4065/789
All words	82.5%	3260/692
Known words	84.3%	3032/565
Unknown words	64.2%	228/127

We can see that the overall tagging accuracy is 83.7%, which is less than in the randomly partitioned training/testing sets and underlines the intuition that the Appendix is quite different from the rest of the book. This is somewhat reflected also in the accuracies on unknown words, which are here 64.2%, but were 67.55% on the random fold.

In Table 12 we concentrate only on nouns and adjectives. Here the accuracy is even somewhat lower, bottoming out at 58.3% for unknown nouns.

Table 12. Validation results for the TnT tagger on nouns and adjectives.

		All		Known		Unknown
	Accuracy	Correct/Err	Accuracy	Correct/Err	Accuracy	Correct/Err
Nouns	73.8%	708/252	77.5%	599/174	58.3%	109/ 78
Adjectives	62.3%	276/167	60.7%	213/138	68.4%	63/ 29
Both	70.1%	984/419	72.2%	812/312	61.6%	172/107

The above results raise fears that cascading the tagger and the analyser might not give much better results than simply assigning each word form as the lemma, but as the following section will show, this is not quite the case.

5 Experiment and Results

The previous sections explained the '1984' dataset, and the training and separate testing of the learned analyser and tagger on the validation set. This section gives the results where the two are combined to predict the correct lemma of (unknown) words in the validation and testing sets. We describe two experiments; one is on the Appendix of the '1984' novel, the other on a Slovenian/EU legal document.

5.1 Lemmatisation of the Validation Set

The first experiment concerns the validation set, i.e. the Appendix of the novel. The Appendix was first tagged with TnT, following which the predicted tags were used for morphological analysis. For convenience, we first summarise the relevant data in the validation set in Table 13.

Table 13. Distribution of words in the validation set (Appendix of the novel).

	All			Known			Unknown				
Category	Token	Type	Lemma	Token	Type	Lemma	Token	Type	Lemma	MSD	=
All words	3952	1557	1073	3597	1276	828	355	281	245		
Nouns	960	533	379	773	389	252	187	144	127	37	85
Adjectives	443	347	245	351	265	173	92	82	72	31	26
Both	1403	880	624	1124	654	425	279	226	199	68	111

The Table gives for each of all, known and unknown words, nouns and adjectives, the number of all tokens in the Appendix, the number of different word forms and the number of lemmas. 'Unknown'-ness was computed against the lexically tagged body of the novel; the words whose lemma is not in the training corpus are unknown. The Table shows that 58% of all lemmas, and 81% of unknown lemmas are nouns or adjectives. Word Forms of an unknown noun/adjective lemma, on average, appear 1.2 times in the text, and the word form and lemma are different in 60% of the cases.

We then tested the combination tagger/analyser on the unknown nouns and adjectives. Because we take the part of speech of the unknown words as given, our assesment does not take into account errors where the tagger classifies an unknown word as a noun or adjective, even though the word in fact belongs to a different part of speech. If the analyser then attempts to lemmatise these words, the results are wrong, except for isolated lucky guesses. In the validation set, there were 59 words misstaged as a noun or adjective, which is 1.5% of all the words or 4% of the total number of true nouns and adjectives in the Appendix.

As was explained in the preceding sections, tagging is 87.5% correct on known and 61.2% on unknown noun and adjective tokens, while lemmatisation is correct 98.3% and 93.9% respectively. When the two methods are combined, the accuracy is as given in Table 14.

Table 14. Lemmatisation results on the validation set.

	All		Known		Unknown	
	Accuracy	Correct/Err	Accuracy	Correct/Err	Accuracy	Correct/Err
Nouns	91.7%	880/ 80	95.4%	738/ 35	75.9%	142/ 45
Adjectives	87.6%	388/ 55	88.0%	309/ 42	85.9%	79/ 13
Both	90.4%	1268/135	93.1%	1047/ 77	79.2%	221/ 58

The accuracy of lemmatisation is thus 79.2%. A closer look at the errors reveals that the majority is due to the fact that the TnT tagger tags a noun or an adjective with the wrong part of speech. This happens in 78 cases (58% of the errors); in 60 of them, the assigned PoS in not a noun or adjective, and in 18 a noun is misstagged as an adjective or vice-versa.

Obviously, tagger performance is the limiting factor in the achieved accuracy although the lemmatisation often manages to recover from errors of tagging. That is, in a large number of cases (245 known / 53 unknown), the predicted lemma of the word was correct, even though the assigned MSD was wrong. In fact, this is not surprising, as the unknown word guesser in TnT builds a suffix tree that helps it in determining the ambiguity classes of unknown words. Thus, TnT will often make an error when tagging a form that is syncretic to other forms, i.e. is identical in orthography, but has different inflectional features in its MSDs. For lemmatisation, it does not matter which of the syncretic MSDs is given, as they resolve to the same lemma.

While the errors are usually caused by the tagger/analyser tandem returning the wrong lemma, there are some cases (11, 8 known / 3 unknown) where the analyser simply fails, i.e. does not return a result. Even though in two cases TnT correctly tagged the word in question, the others, all of them unknown words, are examples of misstagged words. This means that the analyser can also function as a validation component, rejecting misstagged words.

5.2 Lemmatisation of a Slovenian/EU Legal Document

While the Appendix of the ‘1984’ novel, used for validation, is quite different from the body of the book, which was used for training, we nevertheless wanted to assess the results on a truly different text type, and thus gauge the robustness and practical applicability of the method. For this, we took the Slovene version of the text fully titled the “Europe Agreement Establishing an Association Between the European Communities and their Member States, Acting within the Framework of the European Union, of the One Part, and the Republic of Slovenia, of the Other Part June 10. 1996 Luxembourg”.

The text was collected and encoded as one of the 15 components of the one million word ELAN Slovene-English parallel corpus (Erjavec, 1999). This text is encoded in a similar manner as the '1984', and consists of 1,191 translation segments, which roughly correspond to sentences. It is tokenised into 12,049 words and 2,470 punctuation marks. However, the text had, in the ELAN release, not yet been tagged or lemmatised.

In order to be used as a testing set, the corpus had to be at least reliably lemmatised. This was achieved in two steps: first, the company Amebis d.o.o. kindly lemmatised the text with words known to their morphological analyser BesAna, which includes a comprehensive lexicon of the Slovene language. Here each known word was ambiguously lemmatised; we then semi-interactively, via a series of filters and manual edits, disambiguated the lemmas. This produced the text in which words that are known to BesAna are unambiguously and, for the most part, correctly lemmatised, while those unknown do not have a lemma. The latter do contain interesting terms, but they are mostly abbreviations, foreign words, dates, typos, and similar. The identification of such entities is interesting in its own right, and is usually referred to as 'named entity extraction'. However, this task is not directly connected to lemmatisation. We therefore chose to test the system on those words (nouns and adjectives) which were lemmatised but, again, did not appear in the training set. This gives us a fair approximation of the distribution of new inflected words in texts. With these remarks, Table 15 gives the main characteristics on this testing set. The Table shows that the number of unknown noun and adjective lemmas is about three times greater than in the Appendix.

Table 15. Distribution of words in the Slovenian/EU legal document.

	Token	Type	Lemma
Known	12049	3407	1672
Unknown	1458	863	644
Unknown nouns and adjectives	1322	796	595

For testing on this corpus, we used the same tagging and analysis models as before; we first tagged the complete text, then lemmatised the unknown nouns and adjectives. Here we, of course, do not have an evaluation on the correctness of the tagging procedure or of the morphological analysis in isolation, as we are lacking the correct MSDs. Table 16 summarizes the results of testing the tagger/analyser tandem. It contains the accuracy results for unknown noun and adjective tokens, as well as for their word types and lemmas.

For each of the data classes the Table contains the number of all items, and the number of correctly lemmatised ones, also as a percentage of the total. The mislematised cases are further subdivided into those that were simply wrong, those that, for types and lemmas sometimes returned the correct lemmatisation, and sometimes the incorrect one, and those where the analyser failed to analyse the word.

Table 16. Lemmatisation results on the Slovenian/EU legal document.

	Token	Type	Lemma
Accuracy	81.3%	79.8%	75.6%
All	1322	796	595
Correct	1075	635	450
Error	247	161	145
Wrong	195	105	73
Mixed	-	38	62
Fail	52	18	10

The Table shows that the per-token accuracy of 81.3% is in fact slightly higher than on the Appendix (79.2%), and shows the method to be robust. The analysis of the errors per word form type and lemma shows lower accuracy, but also points the way to improving the results. Instead of lemmatising tokens, i.e. each word in the text separately, the text can be preprocessed first, to extract the lexicon of unknown words. This would give us the equivalent of the Types column, where we can see that a significant portion of the errors are either 'mixed' cases or failures of the analyser. A voting regime on the correct lemmatisation can be applied to the mixed cases, while the failures, as was discussed above, usually point to errors in tagging.

6 Summary and Discussion

We have addressed the problem of lemmatisation of unknown words, in particular nouns and adjectives, in Slovene texts. This is a core normalisation step for many language processing tasks expected to deal with unrestricted texts. We approached this problem by combining a morphological analyser and a morphosyntactic tagger. The language models for both components were inductively learned from a previously tagged corpus, in particular, from the Slovene translation of the '1984' novel.

We tested the combination of the learned analyser and tagger on the Appendix of the '1984' novel, as well as on a completely different text type, namely a Slovenian/EU legal document. In both cases, the overall accuracy of lemmatisation of unknown nouns and adjectives is about 80%.

Even at this level of accuracy, the lemmatisation approach proposed can be useful as an aid to the creation and updating of language resources (lexica) from language corpora. To our knowledge, there are no published results for lemmatisation of unknown words in Slovene or even other Slavic languages, so it is difficult to give a comparable evaluation of the results.

The combination of the morphological analyser and the tagger is performed in a novel way. Typically, the results of morphological analysis would be given as input to a tagger. Here, we give the results of tagging to the morphological analyser: an unknown word form appearing in a text is passed on to the analyser together with its morphosyntactic tag produced by the tagger.

Our method relies heavily on the unknown word guessing module of the tagger. While the TnT tagger has superior performance on unknown words as compared to other taggers, it, in the Appendix, still reaches only 64%, while the accuracy of the analyser is 94%. Given the combined accuracy of 79%, it is obvious that some of the errors committed by the tagger are not fatal: if the morphosyntactic tag produced by the tagger is within the inflectional ambiguity class of the word form, then the analyser should get the lemma right.

There are some obvious directions by which to improve the currently achieved accuracy. In our experiments here we used only the '1984' corpus for learning the language model. While enlarging the rather small training corpus is the obvious route, annotating corpora is a very time consuming task. However, plugging a larger lexicon into the system, which would at least cover all the closed word classes would be feasible and should improve the accuracy of tagging. Another extension to be considered is the addition of verbs, as the next largest open class of words.

Another avenue of research would be to combine the morphological analyser and the tagger in a more standard fashion, which to an extent is already done in the TnT tagger. Here we use morphological analyser first to help the tagger postulate the ambiguity classes for unknown words. While this proposal might sound circular, one can also say that the lemmatiser and the tagger each impose constraints on the context dependent triplet of word-form, lemma and MSD. It is up to further research to discover in which way such constraints are best combined.

It would also be interesting to compare our approach to morphological analysis, where synthesis rules are learned separately for each morphosyntactic description (MSD) to an approach where rules are learned for all MSDs of a word class together.

Acknowledgements

Thanks are due to Amebis, d.o.o. for ambigously lemmatising the ELAN corpus. Thanks also to Suresh Manadhar for earlier cooperation on learning morphology and for providing us with CLOG and to Thorsten Brants for providing TnT.

References

1. Brants, T. (2000). TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000* Seattle, WA. http://www.coli.uni-sb.de/~thorsten/tnt/.
2. Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, *21*(4), 543–565.

3. Chanod, J., & Tapanainen, P. (1995). Creating a tagset, lexicon and guesser for a French tagger. In *Proceedings of the ACL SIGDAT workshop From Text to Tags: Issues in Multilingual Language Analysis* Dublin.
4. Cussens, J. (1997). Part-of-speech tagging using Progol. In *Proceedings of the 6th International Workshop on Inductive Logic Programming*, pp. 93–108 Berlin. Springer.
5. Cussens, J., Džeroski, S., & Erjavec, T. (1999). Morphosyntactic tagging of Slovene using Progol. In Džeroski, S., & Flach, P. (Eds.), *Inductive Logic Programming; 9th International Workshop ILP-99, Proceedings*, No. 1634 in Lecture Notes in Artificial Intelligence, pp. 68–79 Berlin. Springer.
6. Cutting, D., Kupiec, J., Pedersen, J., & Sibun, P. (1992). A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 133–140 Trento, Italy.
7. Daelemans, W., Zavrel, J., Berck, P., & Gillis, S. (1996). MBT: A memory-based part of speech tagger-generator. In Ejerhed, E., & Dagan, I. (Eds.), *Proceedings of the Fourth Workshop on Very Large Corpora*, pp. 14–27 Copenhagen.
8. Dimitrova, L., Erjavec, T., Ide, N., Kaalep, H.-J., Petkevič, V., & Tufiş, D. (1998). Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages. In *COLING-ACL '98*, pp. 315–319 Montréal, Québec, Canada.
9. Džeroski, S., Erjavec, T., & Zavrel, J. (1999). Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. Research report IJS-DP 8018, Jožef Stefan Institute, Ljubljana. http://nl.ijs.si/lll/bib/dzerza-report/.
10. Erjavec, T. (1999). The ELAN Slovene-English Aligned Corpus. In *Proceedings of the Machine Translation Summit VII*, pp. 349–357 Singapore. http://nl.ijs.si/elan/.
11. Erjavec, T., & (eds.), M. M. (1997). Specifications and notation for lexicon encoding. MULTEXT-East final report D1.1F, Jožef Stefan Institute, Ljubljana. http://nl.ijs.si/ME/CD/docs/mte-d11f/.
12. Erjavec, T., Lawson, A., & Romary, L. (1998). East meets West: A Compendium of Multilingual Resources. CD-ROM. ISBN: 3-922641-46-6.
13. Manandhar, S., Džeroski, S., & Erjavec, T. (1998). Learning multilingual morphology with CLOG. In Page, D. (Ed.), *Inductive Logic Programming; 8th International Workshop ILP-98, Proceedings*, No. 1446 in Lecture Notes in Artificial Intelligence, pp. 135–144. Springer.
14. Mikheev, A. (1997). Automatic rule induction for unknown-word guessing. *Computational Linguistics*, *23*(3), 405–424.
15. Mooney, R. J., & Califf, M. E. (1995). Induction of first-order decision lists: Results on learning the past tense of English verbs. *Journal of Artificial Intelligence Research*, pp. 1–24.
16. Ratnaparkhi, A. (1996). A maximum entropy part of speech tagger. In *Proc. ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, pp. 491–497 Philadelphia.

17. Sperberg-McQueen, C. M., & Burnard, L. (Eds.). (1994). *Guidelines for Electronic Text Encoding and Interchange.* Chicago and Oxford.
18. Steetskamp, R. (1995). An implementation os a probabilistic tagger. Master's thesis, TOSCA Research Group, University of Nijmegen, Nijmegen. 48 p.
19. van Halteren, H. (Ed.). (1999). *Syntactic Wordclass Tagging.* Kluwer.