Guessing the Correct Inflectional Paradigm of Unknown Croatian Words

Jan Šnajder

University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
jan.snajder@fer.hr

Abstract

A real-life morphological analyzer must be able to handle properly the out-of-vocabulary words. We address the task of guessing the correct inflectional paradigm of unknown Croatian words. We frame this as a supervised machine learning problem: we train a model for deciding whether a candidate lemma-paradigm pair is correct based on a number of string- and corpus-based features. Our aim is to examine the machine learning aspect of the problem: we analyze the features and evaluate the classification accuracy using different feature subsets. We show that satisfactory level of accuracy (92%) can be achieved with SVM using a combination of string- and corpus-based features. We discuss a number of possible directions for future research.

Ugibanje pravilne pregibne paradigme za neznane hrvaške besede

Uporaben morfološki analizator mora znati pravilno obravnavati tudi besede, ki jih nima v leksikonu. Prispevek je posvečen ugibanju pravilne pregibne paradigme za neznane hrvaške besede z uporabo nadzorovanega strojnega učenja. Model se odloči, ali je kandidat oz. par lema-paradigma, pravilen glede na večje število lastnosti, ki temeljijo na nizih in korpusu. Namen prispevka je, da preuči razne vidike strojnega učenja tega problema: analiziramo uporabljene lastnosti in ovrednotimo natančnost klasifikacije glede na različne podmnožice lastnosti. Pokažemo, da lahko zadovoljivo raven natančnosti (92%) dosežemo s SVM in z uporabo kombinacije lastnosti nizov in korpusa. Obravnavamo tudi več smernic za nadaljnje delo.

1. Introduction

Morphological analysis plays a vital role in many natural language processing applications, especially for morphologically rich languages such as Croatian. Morphological analysis typically relies on some form of a morphological lexicon, which lists the stems (or lemmas) and their associated word-forms. In a word-and-paradigm setting (Hockett, 1954), the relation between the stem and its word-forms is defined by an inflectional paradigm (pattern). The unavoidable problem of lexicon-based morphological analysis is the limited lexicon coverage. A real-life morphological analyzer must be able to deal in a satisfactory manner with out-of-vocabulary words. In a word-and-paradigm setting, this means being able to guess the correct inflectional paradigm of an unknown word-form.

In this paper we address the task of guessing the correct inflectional paradigm of unknown Croatian words. We frame this as a supervised machine learning problem: we train a model that decides which paradigm is correct based on a number of string- and corpus-based features. To guess the paradigm of an unknown word, we first generate the candidate lemma-paradigm pairs using a morphology grammar, and then use the classifier to decide which pair is correct. This is in contrast to most earlier approaches, which use hand-crafted scoring functions to decide on the correct paradigm. The aim of this paper is to examine the machine learning aspect of the problem: what the relevant features are and how well can we do on this classification task. We carry out feature analysis and evaluate the classification accuracy using different feature subsets. We show that satisfactory level of accuracy can be achieved with a combination of string- and corpus-based features.

The rest of the paper is structured as follows. In the next section we give a brief overview of related work. In Section 3 we define the problem, while in Section 4 we describe the features used for building the models. In Section 5 we analyze the features and evaluate the classification accuracy. In Section 6 we discuss the results and outline directions for further research. Section 7 concludes the paper.

2. Related Work

Much work on paradigm guessing comes from research in part-of-speech (POS) tagging and the related task of POS guessing (Mikheev, 1997; Kupiec, 1992). The problem has also been addressed in the context of rule-based machine translation systems (Esplá-Gomis et al., 2011). However, most work seems to address paradigm guessing in relation to (semi-)automatic lexicon acquisition (Oliver, 2003; Tadić and Fulgosi, 2003; Oliver and Tadić, 2004; Clement et al., 2004; Sagot, 2005; Forsberg et al., 2006; Hana, 2008; Šnajder et al., 2008; Adolphs, 2008; Kaufmann and Pfister, 2010; Esplá-Gomis et al., 2011). The basic idea is to first use a lemmatizer to obtain the lemmas and paradigms for each word-form from corpus. Because of grammar ambiguity, this usually results in a number of possible candidates. Thus, the next step is to disambiguate the output of the morphology grammar by assessing the plausibility of each lemma-paradigm pair. This is most commonly done by generating the corresponding word-forms and analyzing their corpus frequencies. An incorrect lemma-paradigm pair is likely to produce linguistically invalid word-forms that will not be attested in the corpus, and a suitably designed corpus-based scoring function can be used to decide which paradigm is correct. Some approaches use the

web as additional source of information (Oliver and Tadić, 2004; Cholakov and Van Noord, 2009). Moreover, some approaches use word-form properties to decide on the correct paradigm: Forsberg et al. (2006) use hand-crafted constraints, while Segalovich (2003) guesses the stems and the paradigms based on morphological similarity. It is also possible to use context-based information when analyzing the word-forms from corpus (Kaufmann and Pfister, 2010). More recent approaches use machine learning to predict the stem and the morphosyntactic features (Kaufmann and Pfister, 2010). In many cases the problem of paradigm guessing is also addressed in an unsupervised setting, in which paradigms are induced by clustering the word-forms from corpus and an analysis of their endings (Nakov et al., 2004; Oliver, 2003; Esplá-Gomis et al., 2011) – an instance of the more general task of morphology induction (Goldsmith, 2001).

3. Problem Definition

The problem of guessing inflectional paradigms of (unknown) words can be formulated as follows: given a word-form w, determine its correct stem s and its correct inflectional paradigm p. The correct paradigm is the one which, when used with stem s, generates the valid word-forms, including word-form w. The stem and the paradigm are tied together: given w, the inflectional paradigm (possibly ambiguously) determines the stem of w. Moreover, the stem and the inflectional paradigm (possibly ambiguously) determine the lemma l. Thus, the problem actually amounts to determining, for a given word-form, its lemma and the associated inflectional paradigm. In what follows, we call a pair (l, p), consisting of lemma l and inflectional paradigm p, a *lemma-paradigm pair*, or an LPP for short. We call an LPP (l, p) *correct* if (1) the lemma l is valid (it is an existing word of the language and it is indeed a lemma) and (2) the paradigm p is the correct paradigm for l; otherwise we call the LPP *incorrect*. To difficulty in determining the correct inflectional paradigm arises from the fact that for most word-forms there are many candidate LPPs. This problem is typically approached in two steps: generation of LPP candidates and selection of LPP candidates. Selection can be accomplished using scoring or, as we do, using classification.

3.1. LPP generation

The candidate LPPs of a given word-form are generated using a morphology grammar (an inflectional morphology model). Te concrete implementation of the grammar does not concern us here. We assume that the grammar is generative (capable of generating word-forms given a lemma) and reductive (capable of lemmatizing a word-form). We can abstract this with two functions:

$$wfs(l, p) \mapsto \{(w_1, t_1), (w_2, t_2), \ldots, (w_n, t_n)\} \quad (1)$$

which, given a LPP, generates a set of word-forms $w_1, \ldots, w_n$ paired up with the corresponding morphological tags $t_1, \ldots, t_n$, and

$$lm(w) \mapsto \{(l_1, p_1), (l_2, p_2), \ldots, (l_m, p_m)\} \quad (2)$$

which lemmatizes a word-form to a set of candidate LPPs. In general, one lemma may be associated with more than one paradigm, and one paradigm may be associated with more than one lemma. We also assume that the grammar can reduce each lemma to its stem.

In this work we use the Croatian higher-order functional morphology (HOFM) grammar described by Šnajder and Dalbelo Bašić (2008) and refined by Šnajder (2010). The current version of the grammar uses 93 paradigms: 48 for nouns, 13 for adjectives, and 32 for verbs. The morphological tags are encoded as MULTEXT-East descriptors (Erjavec et al., 2003). Following are examples of word-form generation and lemmatization using the grammar:

```
> wfs "vojnik" N04
[("vojnik","N-msn"),("vojnika","N-msg"),
("vojnika","N-msa"),("vojnika","N-mpg"),
("vojniku","N-msl"),("vojniče","N-msv"),...]

> lm "vojnika"
[("vojnik",N01),("vojnikin",N03),
("vojnik",N04),("vojniak",N05),
("vojniak",N06),("vojniko",N17),...]
```

The second example illustrates the ambiguity of the grammar: many LPPs have been generated (22 in total), of which only the third one is correct. Despite the fact that HOFM defines applicability conditions for certain paradigms, the level of ambiguity is still quite large. On average, each word-form is lemmatized to 17 candidate LPPs, among which there are 7 distinct lemmas and 15 distinct paradigms.

3.2. LPP classification

Given candidate LPPs generated for an unknown word, we wish to decide which one is correct. In a supervised machine learning setting, the problem may in principle be cast as (1) multiclass classification (choosing one LPP among candidate LPPs), (2) multilabel classification (choosing a number of LPPs among candidate LPPs), or (3) binary classification (deciding for each LPP from candidate LPPs whether it is correct). The problem with (1) is that it does not account for homographs (the cases in which a single word-form has more than one correct LPP). The problem with (2) is that it is difficult to define the possible classes (they should encode both the stem transformation and the paradigm). Moreover, both (1) and (2) are difficult to combine with the output of a morphology grammar. Approach (3) is the most straightforward and we shall follow it here.

For classification, we use the SVM with an RBF kernel. The SVM algorithm tends to outperform other machine learning algorithms on a variety of learning problems. The RBF kernel implicitly defines an infinite-dimensional feature space, and is thus a good choice for problems for which the number of examples is much larger than the number of features, which will be the case here.

As source of training data, we use the semi-automatically acquired inflectional lexicon from (Šnajder et al., 2008). The lexicon contains 68,465 manually verified LPPs for Croatian nouns, adjectives, and verbs. We will use a fraction of this data for training and testing. It should be

noted that the distribution of LPPs in the lexicon with respect to the paradigms is very uneven; the ten least frequent paradigms appear only 40 times in the lexicon, whereas the ten most frequent paradigms appear over 50,000 times.

4. Features

Given an LPP, we compute a set of features based on which the LPP can be classified as either correct or incorrect. We distinguish between two groups of features: string-based and corpus-based.

4.1. String-based features

The string-based features are based on the orthographic properties of the lemma or the stem. The intuition behind this is that incorrect LPPs tend to generate ill-formed (or somewhat odd-formed) stems and lemmas. For example, there is no adjective in Croatian language that ends in *-kč*; an LPP that would generate such a stem could be discarded immediately. In fact, many paradigms defined in traditional grammar books are conditioned on the stem ending, requiring that it belongs to a certain group of phonemes or that it forms a consonant group. Similarly, there are paradigms that are applicable only to one-syllable stems. We use the following string-based features:

1. *EndsIn* – the ending character of the stem;
2. *EndsInCgr* – a binary feature indicating whether the word-forms ends in a consonant group (two consecutive consonants);
3. *EndsInCons* – a binary feature indicating whether the word-form ends in a consonant;
4. *EndsInNonPals* – a binary feature indicating whether the word-form ends in a non-palatal (*v*, *r*, *l*, *m*, *n*, *p*, *b*, *f*, *t*, *d*, *s*, *z*, *c*, *k*, *g*, or *h*);
5. *EndsInPals* – a binary feature indicating whether the word-form ends in a palatal (*lj*, *nj*, *ć*, *d*, *č*, *dž*, *š*, *ž*, or *j*);
6. *EndsInVelars* – a binary feature indicating whether the word-form ends in a velar (*k*, *g*, or *h*);
7. *LemmaSuffixProb* – the probability $P(s_l|p)$ of lemma l having a three-letter suffix s_l given inflectional paradigm p;
8. *StemSuffixProb* – the probability $P(s_s|p)$ of stem s having a three-letter suffix s_s given inflectional paradigm p;
9. *StemLength* – the number of characters in the stem;
10. *NumSyllables* – the number of syllables in the stem (determined heuristically);
11. *OneSyllable* – a binary feature indicating whether *NumSyllables* equals 1.

4.2. Corpus-based features

The corpus-based features are calculated based on the frequencies of word-forms attested in the corpus. The general idea is that a correct LPP should have more of its word-forms attested in the corpus than an incorrect LPP. Instead of only looking at total counts of attested word-forms, one can also look at the distributions of attested word-forms across the morphological tags. The intuition behind this is that every inflectional paradigm has its own distribution of morphological tags, and that a correct LPP will generate word-forms that obey such a distribution. For instance, in case of a noun paradigm, we can expect a genitive word-form to be far more frequent than a vocative word-form. Hence, an LPP that generates more vocative word-forms than genitive word-forms is unlikely to be correct.

In what follows, we use $\#(w, C)$ to denote the number of occurrences of word-form w in corpus C. Set $T(p)$ denotes the set of morphological tags of inflectional paradigm p. Let $P(t|p)$ denote the probability distribution of morphological tag t conditioned on the inflectional paradigm p, and let $P(t|l, p)$ denote the probability of morphological tag t generated by LPP (l, p). We obtain these distributions as maximum likelihood estimates using the LPPs from the inflectional lexicon L and word-form frequencies from corpus C:

$$P(t|p) = \frac{\sum_{(l,p')\in L;p'=p;(w,t')\in wfs(l,p);t'=t} \#(w, C)}{\sum_{(l,p')\in L;p'=p;w\in wfs'(l,p)} \#(w, C)}$$

$$P(t|l, p) = \frac{\sum_{(w,t')\in wfs(l,p);t'=t} \#(w, C)}{\sum_{w\in wfs'(l,p)} \#(w, C)}$$

where wfs' is a simpler version of the wfs function that only returns the word-forms. Notice that, because we do not perform POS tagging of the corpus, we count the ambiguous word-forms (inner and outer homographs) multiple times. We use the following corpus-based features:

1. *LemmaAttested* – a binary feature indicating whether the lemma is attested in the corpus, i.e., $\#(l, C) > 0$;
2. *Score0* – the number of corpus-attested word-form types generated by the LPP:
$$score_0(l, p) = |wfs'(l, p) \cap C|$$
3. *Score1* – the sum of corpus frequencies of word-forms generated by the LPP:
$$score_1(l, p) = \sum_{w\in wfs'(l,p)} \#(w, C)$$
4. *Score2* – the proportion of corpus-attested word-form types generated by the LPP:
$$score_2(l, p) = \frac{|wfs'(l, p) \cap C|}{|wfs'(l, p)|}$$
5. *Score3* – the product of paradigm-conditioned distribution of morphological tags and the distribution of tags generated by the LPP:
$$score_3(l, p) = \sum_{t\in T(p)} P(t|p) \times P(t|l, p)$$

6. *Score4* – the expected number of corpus-attested word-form types generated by the LPP:

$$score_4(l,p) = \sum_{t \in T(p)} P(t|p) \times \min\left(1, \#(w,C)\right)$$

7. *Score5* – the Kullback-Leibler divergence between the paradigm-conditioned distribution of morphological tags, $p_1(t) = P(t|p)$, and the distribution of tags generated by the LPP, $p_2(t) = P(t|l,p)$:

$$score_5(l,p) \quad = \quad \mathrm{KL}(p_1||p_2)$$

8. *Score6* – the Jensen-Shannon divergence between the aforementioned distributions:

$$score_6(l,p) = \mathrm{KL}(p_1||p_2) + \mathrm{KL}(p_2||p_1)$$

9. *Score7* – the cosine similarity between the aforementioned distributions:

$$score_7(l,p) = \frac{\sum_{t \in T(p)} p_1(t) \times p_2(t)}{\sqrt{\sum_{t \in T(p)} p_1(t)^2 \times \sum_{t \in T(p)} p_2(t)^2}}$$

We computed the above features on the *Vjesnik* newspaper corpus totaling 23 million word-form tokens and 330,298 word-form types (the same corpus was that used for lexicon acquisition in (Šnajder et al., 2008)).

4.3. Other features

Besides the string- and corpus-based features, we also use the following two features:

1. *ParadigmId* – a nominal feature denoting the LPP's inflectional paradigm;

2. *POS* – the part-of-speech of the LPP's inflectional paradigm (noun, adjective, or verb).

5. Evaluation

The purpose of evaluation is twofold: apart from determining how accurately we can guess the inflectional paradigms, we also wish to analyze what features are most useful for this task.

5.1. Data set

We compiled the data set for training and testing from the aforementioned inflectional lexicon (Šnajder et al., 2008). We sampled from the lexicon 5,000 LPPs for training and 5,000 LPPs for testing. Because the distribution of paradigms is very uneven, we used stratified sampling with respect to the inflectional paradigms. Moreover, we ensured that there is no LPP that appears in the test set, but does not appear in the training set (otherwise the probability distributions would be undefined). To generate the negative training and testing examples, we proceeded as follows. For each LPP, we generate all word-forms using the function wfs. Then, for all corpus-attested obtained word-forms, we generate the candidate LPPs using the function lm, and filter out those LPPs that exist in the lexicon. This generates a large number of incorrect LPPs, from which we again sample 5,000 for training and 5,000 for testing. Thus we end up with 10,000 LPPs (5,000 correct and 5,000 incorrect) in each the training and the test set. Given the number of classes and features (a total of 146 binary-encoded features), the amount of training data ought to be sufficient; a larger training set would unnecessary increase the time required for training. Notice that the training set contains correct and some incorrect LPPs for each selected word-form, while the test set contains LPPs obtained from word-forms that did not appear in the training set.

5.2. Feature analysis

Some of the features we defined are redundant or perhaps irrelevant for LPP classification. Because in absolute terms the number of features is not large, we need not perform feature analysis in order to reduce this number. Instead, the purpose of our feature analysis is to gain insight into what features are useful for paradigm guessing.

For feature analysis we used the open source tool Weka (Hall et al., 2009). Table 1 summarizes the results. We used three univariate filtering methods: information gain (IG), gain ratio (GR), and RELIEF method (Kononenko, 1994). We lists feature rankings obtained on the training set, with first five ranks shown in bold. The first two methods produced similar rankings: among string-based features, suffix probabilities are ranked the highest, while among corpus-based features, feature *Score5* is often ranked high, while ranks of other features vary. There are a number of features that are low-ranked (rank > 10) by each of the three methods: the five *EndsIn** features, *NumSyllables*, *OneSyllable*, *StemLength*, *Score1*, *Score3*, and *POS*.

The univariate methods do not measure the dependencies between the features, thus they cannot detect feature redundancy. We therefore also analyzed the features using two multivariate feature subset selection (FSS) methods: correlation-based feature selection (CFS) (Hall, 1998) and consistency subset selection (CSS) (Liu and R., 1996), both with greedy forward search as the optimization method. Table 1 shows the optimal subset selection obtained with each of these methods. Notice that both selected subsets contain both string- and corpus-based features.

5.3. Classification accuracy

For training and testing of models, we used the LIBSVM implementation of the SVM algorithm (Chang and Lin, 2011). We trained eight models using different feature subsets. We optimized the parameters of each model separately using 5-fold cross-validation on the training set. Classification accuracy on the test set is shown in Table 2. The reliability of probability estimates used for some of the corpus-based features depends on the frequencies of word-forms in the corpus. In a realistic setting, the unknown words tend to be less frequent in corpus. The last two columns of Table 2 show the classification accuracy for LPPs for which the frequency of word-forms in the corpus is less than or equal to 100 (rare words, accounting for 66% of the test set) and less than or equal to 10 (very rare words, accounting for 22% of the test set). The performance baseline is the majority class in each test set.

Table 1: Feature selection analysis

	Ranking			FSS	
Feature	IG	GR	RELIEF	CFS	CSS
String-based features:					
EndsIn	12	13	**2**		×
EndsInCgr	21	21	11		×
EndsInCons	17	15	20		
EndsInNonpals	22	22	19		
EndsInPals	19	18	21		
EndsInVelars	20	19	18		
LemmaSuffixProb	**2**	**2**	**3**		×
NumSyllables	14	14	12		×
OneSyllable	16	17	17		×
StemLength	15	16	15		×
StemSuffixProb	**1**	**1**	6	×	×
Corpus-based features:					
LemmaAttested	11	**3**	8	×	
Score0	8	**4**	16	×	
Score1	13	12	22		×
Score2	6	8	**5**		×
Score3	10	11	13		×
Score4	9	10	14		
Score5	**4**	**5**	**4**		
Score6	**3**	6	9		×
Score7	**5**	7	7		×
Other features:					
ParadigmId	7	9	**1**		×
POS	18	20	10		

Table 2: Classification accuracy (%)

	Word-forms attested		
Features (count)	≥ 1	≤ 100	≤ 10
All (22)	**91.97**	**91.94**	**90.65**
String-based (13)	87.01	87.69	87.98
Corpus-based (11)	87.78	86.59	82.04
IG (5)	81.14	79.05	76.46
GR (5)	59.76	80.90	77.29
RELIEF (5)	90.62	90.60	89.27
CFS (3)	81.69	79.51	78.67
CSS (13)	27.41	91.56	90.37
Baseline	50.00	56.51	69.92

As expected, the maximum accuracy of about 92% was achieved when using all features. Interestingly, in this case the classification accuracy does not decrease much on rare or very rare word-forms. Using only string- or corpus-based features gives worse performance than when using both kinds of features. Moreover, as expected, using only corpus-based features decreases the performance on rare words. As regards the models with feature selected subsets, all perform above the baseline except the one obtained with CSS. The RELIEF method seems to have selected a very good subset of features; a model with only five features (*ParadigmID*, *EndsIn*, *LemmaSuffixProb*, *Score5*, and *Score2*) performs just slightly worse than the model using the full set of 22 features.

6. Discussion

As the work described in this paper is preliminary, there are a number of issues that should be pointed out, especially as regards the evaluation.

Considering that on average there are 17 candidate LPPs per word-form, accuracy of 92% means that for each unknown word we would on average wrongly classify at least one candidate LPP. However, the problem with the above evaluation is that the test set is balanced in the number of positive and negative examples. In reality, there are more negative examples (incorrect LPPs) than positive examples, of which many can probably be classified as such with high confidence. For future work, we need to evaluate the classifier in terms of precision and recall on a per word basis.

In this work we ignored the classifier confidence scores, which may be used to produce rankings. Paradigm guessing is often addressed as a ranking task, and it would make sense to evaluate it as such. It would also be possible to build a metaclassifier that uses the confidence scores assigned to candidate LPPs to decide which LPP to choose. Moreover, ranking-based classification enables the interactive use of a paradigm guesser, which is very convenient for semi-automatic lexicon enlargement.

Another issue that we did not address is the size and diversity of the training set. Often a large morphological lexicon is not available, and one wishes to use paradigm guessing to acquire such a lexicon. Related to this is the question of how many examples per paradigm we need to learn a good classifier. The active learning framework provides a way to minimize the number of training examples and hence reduce the manual labeling efforts. Active learning may also be combined with ranking-based classification to speed up the annotation process.

Furthermore, there are three additional evaluation scenarios that may be considered. First is the evaluation in the context of rule-based tagging (e.g., constraint grammar based tagging, as described by Peradin and Šnajder (2012)), in which the goal is to disambiguate ambiguous morphosyntactic tags, rather than ambiguous paradigms (the former is probably an easier task in most cases). Related to this is a setting in which corpus-based information is not available (e.g., on-the-fly tagging), and one must choose the correct paradigm using only string-based and possibly context-based features. Yet another interesting evaluation scenario is the acquisition of inflectional lexicons from a list of lemmas, which is obviously an easier task than the one we addressed here because the level of grammar ambiguity is lower.

7. Conclusion

We have addressed the problem of paradigm guessing for unknown Croatian words as a binary classification task over the output of a morphology grammar. We defined a

number of string- and corpus-based features and trained different models on selected subsets of these features. The highest accuracy (about 92%) was achieved using the complete set of 22 features. Just slightly worse performance can be obtained with a subset of only five features (a combination of string- and corpus-based features). Degradation in classification performance on rare words is minimal.

We have outlined several directions for further research. We plan to evaluate paradigm guessing as a ranking task on a per word basis, in the context of semi-automatic lexicon acquisition. We also intend to apply paradigm guessing for rule-based POS tagging of Croatian. From a machine learning perspective, we intend to experiment with additional features (including context-based features).

8. Acknowledgments

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under Grant 036-1300646-1986.

9. References

P. Adolphs. 2008. Acquiring a poor man's inflectional lexicon for German. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco*.

C.-C. Chang and C.-J. Lin. 2011. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pages 2:27:1–27:27.

K. Cholakov and G. Van Noord. 2009. Combining finite state and corpus-based techniques for unknown word prediction. In *Proceedings of the 7th Recent Advances in Natural Language Processing (RANLP) conference*.

L. Clement, B. Sagot, and B. Lang. 2004. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of LREC'04*, pages 1841–1844, May.

T. Erjavec, C. Krstev, V. Petkevič, K. Simov, M. Tadić, and D. Vitas. 2003. The MULTEXT-East morphosyntactic specifications for Slavic languages. In *Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages*, pages 25–32.

M. Esplá-Gomis, V.M. Sánchez-Cartagena, and J.A. Pérez-Ortiz. 2011. Enlarging monolingual dictionaries for machine translation with active learning and non-expert users. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2011)*, pages 411–415.

M. Forsberg, H. Hammarström, and A. Ranta. 2006. Morphological lexicon extraction from raw text data. In *FinTAL*, pages 488–499.

J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The Weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

M. A. Hall. 1998. Correlation-based feature subset selection for machine learning. Technical report.

J. Hana. 2008. Knowledge- and labor-light morphological analysis. *Ohio State University Working Papers in Linguistics*, 58:52–84.

C. F. Hockett. 1954. Two models of grammatical description. *Word*, 10:210–234.

T. Kaufmann and B. Pfister. 2010. Semi-automatic extension of morphological lexica. In *Computer Science and Information Technology (IMCSIT), Proc. of the 2010 International Multiconference on*, pages 403–409. IEEE.

I. Kononenko. 1994. Estimating attributes: Analysis and extensions of relief. In *European Conference on Machine Learning*, pages 171–182.

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6(3):225–242.

H. Liu and Setiono R. 1996. A probabilistic approach to feature selection - a filter solution. In *13th International Conference on Machine Learning*, pages 319–327.

A. Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

P. Nakov, Y. Bonev, G. Angelova, E. Cius, and W. Von Hahn. 2004. Guessing morphological classes of unknown German nouns. *Recent Advances in Natural Language Processing III (RANLP'03), Nicolov, Nicolas, Kalina Bontcheva, Galia Angelova and Ruslan Mitkov (eds.)*, pages 347–356.

A. Oliver and M. Tadić. 2004. Enlarging the Croatian morphological lexicon by automatic lexical acquisition from raw corpora. In *Proceedings of LREC'04*, pages 1259–1262.

A. Oliver. 2003. Use of internet for augmenting coverage in a lexical acquisition system from raw corpora. In *Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL 2003), RANLP*.

H. Peradin and J. Šnajder. 2012. Towards a constraint grammar based morphological tagger for Croatian. In *Text, Speech and Dialogue*, pages 174–182. Springer.

B. Sagot. 2005. Automatic acquisition of a Slovak lexicon from a raw corpus. *Lecture Notes in Computer Science*, 3658:156–163.

I. Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. *Proceedings of MLMTA*.

J. Šnajder and B. Dalbelo Bašić. 2008. Higher-order functional representation of Croatian inflectional morphology. In *Proceedings of the 6th International Conference on Formal Approaches to South Slavic and Balkan Languages, FASSBL6*, pages 121–130, Dubrovnik, Croatia. Croatian Language Technologies Society.

J. Šnajder, B. Dalbelo Bašić, and Tadić M. 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing and Management*, 44(5):1720–1731.

J. Šnajder. 2010. *Morfološka normalizacija tekstova na hrvatskome jeziku za dubinsku analizu i pretraživanje informacija*. Ph.D. thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb.

M. Tadić and S. Fulgosi. 2003. Building the Croatian morphological lexicon. In *Proceedings of EACL'2003*, pages 41–46.