MODELS FOR PREDICTING THE INFLECTIONAL PARADIGM OF CROATIAN WORDS

Jan ŠNAJDER
University of Zagreb, Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab

Šnajder, J. (2013): Models for predicting the inflectional paradigm of Croatian words. Slovenščina 2.0, 1 (2): 1–34.
URL: http://www.trojina.org/slovenscina2.0/arhiv/2013/2/Slo2.0_2013_2_02.pdf.

Morphological analysis is a prerequisite for many natural language processing tasks. For inflectionally rich languages such as Croatian, morphological analysis typically relies on a morphological lexicon, which lists the lemmas and their paradigms. However, a real-life morphological analyzer must also be able to handle properly the out-of-vocabulary words. We address the task of predicting the correct inflectional paradigm of unknown Croatian words. We frame this as a supervised machine learning problem: we train a classifier to predict whether a candidate lemma-paradigm pair is correct based on a number of string- and corpus-based features. The candidate lemma-paradigm pairs are generated using a handcrafted morphology grammar. Our aim is to examine the machine learning aspect of the problem: we test a comprehensive set of features and evaluate the classification accuracy using different feature subsets. We show that satisfactory classification accuracy (92%) can be achieved with SVM using a combination of string- and corpus-based features. On a per word basis, the F1-score is 53% and accuracy is 70%, which outperforms a frequency-based baseline by a wide margin. We discuss a number of possible directions for future research.

Keywords: computational morphology, paradigm prediction, machine learning, feature selection, Croatian language.

1 INTRODUCTION

Morphological analysis plays an important role in many natural language processing applications. Typical morphological analysis tasks include the recognition of morphologically related words, stemming and lemmatization, segmentation of words into morphemes, and the labeling of morphemes with grammatical features they express. Inflectionally rich languages, such as Slavic languages, are notoriously challenging for morphological analysis as they are highly fusional and abound with morphological syncretisms. For such languages, the word-and-paradigm approach to morphology (Hockett 1954) seems to be the only reasonable option. In traditional grammar, an *inflectional paradigm* is "a set of all the inflected forms that a lexeme assumes" (Aronoff and Fudeman 2011). A paradigm is typically represented as a table (in general, an *n*-dimensional array, where *n* is the number of features) in which each cell corresponds to a particular combination of grammatical features (cf. Table 1). Paradigms with identical patterns of inflection can be grouped together and for each such group a single paradigm can be chosen as an exemplary paradigm. Calder (1989) was among the first to use paradigms in a computational model of morphology. In his work, and most subsequent work related to paradigmatic morphology, the word "paradigm" is used in a more technical sense (which we adopt here) to denote a formal description of an inflectional pattern.

Morphological analysis of inflectionally rich languages typically relies on some sort of morphological lexicon, which lists the stems or lemmas (the canonical forms of lexemes) and their associated paradigms. However, 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 paradigmatic morphology, this means being able to predict the correct inflectional paradigm of a given word-form.

In this article we address the task of predicting the lemma and the correct inflectional paradigm (the description of an inflectional pattern) of unknown Croatian words. We frame this as a supervised machine learning problem: we

train a model that decides which lemma and paradigm are correct based on a number of string- and corpus-based features. The model is used to disambiguate the output of a morphology grammar. Given an unknown word-form as input, we first generate the candidate lemma-paradigm pairs using the morphology grammar, and then use the classifier to decide which pair is correct. This is in contrast to most earlier approaches, which use handcrafted scoring functions to decide on the correct paradigm. The aim of this article is to examine the machine learning aspect of the problem: what the relevant features are and how well we can do on this classification task. We carry out feature analysis and evaluate the classification accuracy using different feature subsets. We show that a satisfactory level of accuracy can be achieved with a combination of string- and corpus-based features. Although our focus is on Croatian language, we believe our results are applicable to other languages, especially Slavic languages.

The rest of the article is structured as follows. In the next section we give a brief overview of related work. In Section 3 we define the problem of paradigm prediction, while in Section 4 we describe the features used for building the models. In Section 5 we analyze the features, evaluate the classification accuracy, and discuss the results. Section 6 concludes the article and outlines future work.

2 RELATED WORK

Much work on paradigm prediction 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 (Esplá-Gomis et al. 2011). However, most work seems to address paradigm prediction 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; Lindén 2009; 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 a 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 in this case 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 handcrafted constraints, while Segalovich (2003) guesses the stems and the paradigms based on morphological similarity. Lindén (2009) uses both corpus-based features and lexicon-based information to learn analogical relations with which lemmas and paradigms of unknown words can be predicted. 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).

Another line of research that has addressed the problem of paradigm induction is unsupervised morphology learning (Hammarström and Borin 2011). Unsupervised morphology learning aims to discover morphology descriptions from unannotated data, for the purpose of, inter alia, deriving language descriptions, bootstrapping morphological analyzers, and modeling language acquisition. The seminal work is that of Goldsmith (2001), who extracts sets of stems and affixes (so-called signatures), the latter bearing resemblance to paradigms, based on minimum description length principle. In other work paradigms are typically induced by clustering the word-forms from corpus and an analysis of their endings (Nakov et al. 2004; Oliver 2003; Monson et al. 2008), possibly within a probabilistic framework (Chan 2006; Dreyer and Eisner 2011).

In this work we do not consider the problem of unsupervised paradigm induction, but instead address the task of paradigm prediction in a supervised setting. We

Case	Singular	Plural
Nominative	*vojnik-∅*	*vojni**c**-i*
Genitive	*vojnik-a*	*vojnik-a*
Dative	*vojnik-u*	*vojni**c**-ima*
Accusative	*vojnik-a*	*vojnik-e*
Vocative	*vojni**č**-e*	*vojni**c**-i*
Locative	*vojnik-u*	*vojni**c**-ima*
Instrumental	*vojnik-om*	*vojni**c**-ima*

Table 1: Inflectional paradigm of the Croatian noun *vojnik* (*soldier*). Stem-internal changes (due to sibilarization and palatalization) are shown in bold.

are interested in building good models for paradigm prediction, assuming that the training data is available. Our work focuses on the machine learning aspect of the problem: we test a comprehensive set of features and carry out a detailed evaluation of the models.

3 PROBLEM DEFINITION

The problem of predicting inflectional paradigms of (unknown) words can be formulated as follows: given a word-form *w*, determine its stem *s* and the corresponding inflectional paradigm *p*. For example, given word-form *vojnika* (genitive singular/accusative singular/genitive plural form of the noun *vojnik* (*soldier*)), we wish to determine that its stem is *vojnik* and that its paradigm *p* is the one shown in Table 1. The corresponding paradigm is the one which, when used with stem *s*, generates the valid word-forms of *s*, including word-form *w* itself. The stem *s* and the paradigm *p* are tied together in the sense that *s* functionally depends on *p*: in other words, given *w*, the inflectional paradigm (possibly ambiguously) determines the stem of *w*.[1] For example, if we know that *p* is the paradigm of *vojnika*, we also know that the stem of *vojnika* is

[1] Ambiguity arises in the presence of a non-bijective transformation from a stem to a word-form, which gives rise to a non-functional inverse transformation back from the word-form to the stem. A typical example in Croatian inflectional morphology are the morphologically conditioned stem-internal changes that replace two or more distinct phonemes with one identical phoneme. A case in point are the palatalization alternations *k/č* (*vojnik→vojniče*) and *c/č* (*stric→striče*). For details, please refer to (Šnajder 2010).

vojnik. Likewise, the stem and the inflectional paradigm (possibly ambiguously) determine the lemma *l*. Thus, the problem of paradigm prediction actually amounts to determining, for a given word-form *w*, its lemma *l* and the associated inflectional paradigm *l*. 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*.

The difficulty in determining the correct inflectional paradigm arises from the fact that for most word-forms there are many candidate LPPs – a large number of possible stems can be combined with many paradigms defined for a language. It should be emphasized that this will be the case even when using a handcrafted morphology grammar. A morphology grammar can, of course, narrow down the space of possibilities, but it cannot completely resolve the ambiguity because the question of which stems combine with what paradigms is ultimately a lexical one. Thus, in order not to discard a possibly valid hypothesis, a morphology grammar will have to overgenerate. In view of this, the problem of paradigm prediction is typically approached in two steps: (1) generation of LPP hypotheses admissible by the grammar and (2) the selection of the correct LPP based on grammar-external evidence. Note that, due to homography, some word-forms will have more than one correct LPP. The selection of correct LPPs is typically accomplished using some heuristic scoring mechanisms. Alternatively, as we do in this article, selection can be framed as a classification problem.

3.1 LPP candidate generation

The first step in paradigm prediction is the generation of LPP candidates using a morphology grammar (an inflectional morphology model). We assume that the grammar is generative (capable of generating word-forms given a lemma) and reductive (capable of reducing a word-form to a stem); consequently, by compositionality we assume that the grammar is capable of lemmatizing a given

word-form. We can abstract this functionality with two functions:

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

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

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

which lemmatizes a word-form to a set of candidate LPPs. Note again that, due to grammar ambiguity (and, in addition, due to homography), the result of lemmatization is a set of LPP candidates. Moreover, again due to grammar ambiguity, a single lemma l_i may be associated with more than one paradigm, while one paradigm p_i may be associated with more than one lemma.

As a concrete grammar implementation, 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 grammar is based on functional representation of word-form transformations and is implemented in the functional programming language Haskell (Jones 2003).[2] The current version of the grammar uses 93 paradigms: 48 for nouns, 13 for adjectives, and 32 for verbs. The HOFM formalism uses a succinct representation of string-based transformations, allowing for compact representation of more complex inflectional paradigms. For example, a paradigm that involves stem-internal changes, such as the one shown in Table 1, can be represented as a single paradigm, without the need to factor it into several paradigms that operate on different allomorphs of the stem. The reader is referred to (Šnajder and Dalbelo Bašić 2008; Šnajder 2010) for details.

At this point we should emphasize that our attention will be focused on paradigm prediction for open-class words: nouns, adjectives, and verbs. Words with other

[2] The Croatian HOFM grammar is available for download under the CC BY-NC-SA 3.0 license from http://takelab.fer.hr/data/hofm

part-of-speech (abbreviations, adpositions, conjunctions, interjections, particles, numerals, pronouns) are less interesting in this respect because they constitute a closed class and/or do not inflect. An exception are the ordinal numbers and adverbs derived from adjectives, which are open-class words and inflect like adjectives. These words could be covered by the adjective paradigms but would need to be additionally disambiguated; we leave this for future work.

In HOFM , the morphological tags are encoded as MULTEXT-East descriptors (Erjavec et al. 2003).[3] MULTEXT-East encodes values of morphosyntactic attributes in a single string, using positional encoding. Each attribute is represented by a single letter at a predefined position, while non-applicable attributes are represented by hyphens. HOFM omits the values of those features that cannot be deduced solely at the morphological level, such as noun type (common/proper) or animacy of nouns and adjectives. For example, descriptor "N-msn" denotes a word-form that is a masculine noun in singular nominative case, but whose type and animacy feature are unknown.[4] As regards the verbs, the current version of HOFM encodes the complete paradigms of main verbs, except the aorist, imperfect, and passive forms. In HOFM, the passive forms are covered by adjectival paradigms, i.e., the passive participle (which is used for building both passive verb forms and adjective forms) is considered as part of the adjectival paradigm.[5] On the other hand, the aorist and imperfect forms were left out because they are rather uncommon in contemporary texts.[6] HOFM also accounts for doubletes (morphological variants with identical grammatical features), quite

[3] The current version of HOFM uses the now-outdated MULTEXT-East Version 3, described at http://nl.ijs.si/ME/V3. The changes introduced by the current MULTEXT-East Version 4 are not relevant for the work described in this article.

[4] Notice that, for predicting the inflectional paradigms, we will only be using morphosyntactic attributes whose values vary across the different word-forms of a word. We can therefore safely omit all attributes that are constant across a paradigm, such as noun type or animacy.

[5] Due to the existence of adjectival and adverbial participles in Croatian language, it is arguably difficult to draw a demarcation line between verbs and adjectives/adverbs. In the current version of HOFM, the verb paradigm includes the adverbial participles but excludes the adjectival participles. In MULTEXT-East Version 3 and Version 4 the verb paradigm includes both types of participles, whereas for the upcoming Version 5 (currently in a draft stage) it is proposed to exclude both types from the verb paradigm. Each of these decisions has theoretical and practical implications.

[6] This is also the case for our newspaper corpus. Notice, however, that extending HOFM to include the aorist and imperfect forms would be straightforward.

common for Croatian adjectives (e.g., the *-og/oga* and *-om/omu/ome* allomorphs in *crvenog/crvenoga* and *crvenom/crvenomu/crvenome*, respectively) and nouns with stem changes (e.g., *tvrtki/tvrtci*).

As an example of word-form generation and lemmatization using the Croatian HOFM, consider the following output of the interaction with the grammar module:[7]

```
> wfs "vojnik" N04
[("vojnik","N-msn"),("vojnika","N-msg"),("vojnika","N-msa"),
("vojnika","N-mpg"),("vojniku","N-msd"),("vojniku","N-msl"),
("vojniče","N-msv"),("vojnikom","N-msi"),("vojnici","N-mpn"),
("vojnici","N-mpv"),("vojnicima","N-mpd"),("vojnicima","N-mpl"),
("vojnicima","N-mpi"),("vojnike","N-mpa")]

> lm "vojnika"
[("vojnik",N01),("vojnikin",N03),("vojnik",N04),("vojniak",N05),
("vojniak",N06),("vojniko",N17),("vojniko",N19),("vojniko",N30),
("vojnik",N37),("vojnik",N41),("vojnika",N45),("vojniko",N20),
("vojnika",N28),("vojnika",N47),("vojnika",N29),("vojnike",N49),
("vojnikati",V17),("vojnik",A01),("vojnik",A06),("vojniak",A15),
("vojniki",A12),("vojniki",A13)]
```

The first example shows the generation of word-forms of lemma *vojnik* according to paradigm N04 (the correct paradigm for this lemma).[8] The second example shows the reduction of word-form *vojnika* to a set of LPP candidates. Due to grammar ambiguity, this particular word-form reduces to 22 LPP candidates, of which only the third one is correct. Despite the fact that HOFM defines applicability conditions for many paradigms (in the form of conditions imposed on the stem), the level of ambiguity is still quite large. On average, each word-form will be lemmatized to 17 candidate LPPs, among which there will be 7 distinct lemmas and 15 distinct paradigms.

[7] Output is from the interactive sessions of the GHC interpreter; http://www.haskell.org/ghc/

[8] In HOFM, paradigms are denoted as N01, N02, etc. for nouns, A01, A02, etc. for adjectives, and V01, V02, etc. for verbs. There is no special meaning associated with the number of the paradigm.

3.2 LPP classification

In the second step, given candidate LPPs generated by the grammar, we wish to decide which one is correct. In a supervised setting, the problem may be cast as (1) multiclass classification (choosing one LPP among many candidate LPPs), (2) multilabel classification (choosing a number of LPPs among many candidate LPPs),[9] or (3) binary classification (deciding for each LPP from candidate LPPs whether it is correct). The problem with (1) and (2) is that that the set of possible classes cannot be straightforwardly defined. More precisely, in these cases each class should correspond to a single LPP (not a single paradigm, as a single paradigm can occur in different LPPs), so one should come up with a way of representing these without actually encoding the lemma itself (e.g., by encoding the paradigm and the stem transformation). Another problem is that not all such classes would be admissible by the grammar, so one would need to find a way to include that information as well (at a cost of increased complexity, this information could be fed to the classifier as a binary feature). An additional, albeit less significant 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).[10] Approach (3), in which binary decisions are made for each LPP candidate, does not suffer from either of these problems and we shall adopt it here.

For classification, we use the support vector machine (SVM) (Vapnik 1999) with a radial basis function (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 instances is much larger than the number of features, which will be the case here. We use the LIBSVM

[9] We adopt standard machine learning terminology, which distinguishes *multiclass classification* (categorizing each instance into a single class from a set of more than two classes) from *multilabel classification* (categorizing each instance into multiple classes simultaneously).

[10] This is not to say that a morphological analyzer can ignore homographs; on the contrary, a morphological analyzer should treat homographs properly by providing all possible analyses. Proper analysis of homographs, however, is not directly related to the task of paradigm prediction: even if we limit ourselves to predicting only one paradigm per word-form, the so-acquired morphological lexicon could still provide multiple analyzes for homographic word-forms.

implementation of the SVM algorithm (Chang and Lin 2011).

As a source of training data, we use the semi-automatically acquired inflectional lexicon from Šnajder et al. (2008).[11] The lexicon was acquired from articles comprising the newspaper section of the Croatian National Corpus totaling 20 million word form tokens (Tadić 2002). 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 candidate generated by the grammar, we compute a set of features based on which the LPP can be classified as either correct or incorrect. At this point we make no attempt to define a minimal set of features; instead, we use features that are easily computable and can be intuitively justified. 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.

[11] Alternatively, we could have used the Croatian Morphological Lexicon (Tadić and Fulgosi 2003), but this would have been less straightforward because this lexicon uses a different set of paradigms from HOFM.

With string-based features we aim to capture this information in an implicit and less strict way.

We use a set of eleven string-based features:

1. *EndsIn* – the ending character of the stem;
2. *EndsInCgr* – a binary feature indicating whether the word-form 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*, *đ*, *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 3-letter suffix s_l given inflectional paradigm p;
8. *StemSuffixProb* – the probability $P(s_s|p)$ of stem s having a 3-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;
11. *OneSyllable* – a binary feature indicating whether *NumSyllables* equals 1.

Notice that some features are overlapping. For example, the *OneSyllable* feature is a stripped down version of the *NumSyllables* feature. While in general a more expressive feature is preferred, the feature might just be too expressive and confuse the model by overfitting it to the training data. To account for this, the standard approach is to first consider all plausible features, some of which might overlap, and then perform feature analysis to filter out the redundant features. We turn to feature analysis in Section 5.2.

The features *StemSuffixProb* and *LemmaSuffixProb* can be seen as soft condi-

Paradigm N01		Paradigm N04		Paradigm A06	
Suffix (s_s)	**P(s_s|N01)**	**Suffix (s_s)**	**P(s_s|N04)**	**Suffix (s_s)**	**P(s_s|A06)**
-ist	0.0196	*-nik*	0.5139	*-ran*	0.1694
-tor	0.0163	*-jak*	0.1389	*-jen*	0.1156
-ing	0.0143	*-jek*	0.0416	*-van*	0.0618
-ter	0.0196	*-log*	0.0416	*-jiv*	0.0565
-nov	0.0108	*-lik*	0.0278	*-đen*	0.0510

Table 2: Five most frequent 3-letter stem suffixes for noun paradigms N01 and N04 and adjective paradigm A06 (estimates from a sample of morphological lexicon from Šnajder et al. (2008)).

tions on stem and lemma endings, respectively. The intuition is that, if a stem or lemma end in a suffix that is highly probable for a particular paradigm, then this paradigm is likely to be the correct one. Conversely, if a stem or a lemma end in a suffix that has rarely or never been observed for a stem or a lemma of a given paradigm, it is very likely that the stem or the lemma are ill-formed and do not belong to that particular paradigm. We obtain these probabilities as maximum likelihood estimates from the morphological lexicon used for training. As an example, consider Table 2, which shows five most frequent stem suffixes for noun paradigms N01 and N04 and adjective paradigm A06.[12] The probability distributions are quite different for the three paradigms. Incidentally, for paradigm N04 suffix *-nik* accounts for more than 51% of suffixes. Returning to our earlier example from Section 3.1, we can use this information as a strong evidence that LPP (*vojnik,* N04) is correct and that LPPs (*vojnik,* N01) and (*vojnik,* A06) are both incorrect.

4.2 Corpus-based features

The second group of LPP features, the corpus-based features, are calculated based on the frequencies of word-forms attested in the corpus. The general idea

[12] Paradigm N01 describes the so-called "type a" declension of masculine nouns, such as *izvor* (*source*) and *ekran* (*screen*). Paradigm N04 is similar to N01, except that it applies to stems ending in *k/g/h*, which undergo a stem change, as exemplified by Table 1. Paradigm A06 describes the inflection of qualificative adjectives with comparative suffix *-iji*, such as *star* (*old*) and *loš* (*bad*).

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, as proposed by Šnajder et al. (2008), 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. Let $T(p)$ denote the set of morphological tags of inflectional paradigm p. Furthermore, 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:

$$P(t|p) = \frac{\sum_{(l,p')\in L;\ p'=p;\ (w,t')\in wfs(l,p);\ t'=t} \#(w, C)}{\sum_{(l,p')\in L;\ p'=p;\ (w,t)\in wfs(l,p)} \#(w, C)} \tag{3}$$

$$P(t|l,p) = \frac{\sum_{(w,t')\in wfs(l,p);\ t'=t} \#(w, C)}{\sum_{(w,t)\in wfs(l,p)} \#(w, C)} \tag{4}$$

Due to syncretism, an identical word-form may be associated with different morphological tags. Because we do not perform POS tagging of the corpus, we do not disambiguate such cases. Instead, we treat an ambiguous group of tags as a single probabilistic outcome.

As an example, consider Table 3, where we show the probability distributions for two paradigms, N01 and N04. Paradigm N01 has six groups of syncretic forms (all forms are syncretic except singular instrumental case), while paradigm N04 has four such groups (see also Table 1). Notice, however, that the syncretism is not shared among the two paradigms. Thus, the distribution of tags for the two

Paradigm N01			Paradigm N04		
Tags (t)	**P(t|N01)**	**P(t|l, N01)**	**Tags (t)**	**P(t|N04)**	**P(t|l, N04)**
N-ms[n|a]	0.37	0.09	N-msn	0.29	0.06
N-m[sg|pg]	0.27	0.77	N-m[sg|sa|pg]	0.32	0.54
N-ms[d|l]	0.11	0.01	N-ms[d|l]	0.06	0.01
N-m[sv|pa]	0.05	0.11	N-msv	0.06	0
N-msi	0.07	0.02	N-msi	0.03	0.01
N-m[pn|pv]	0.11	0	N-mp[n|v]	0.14	0.23
N-mp[d|l|i]	0.02	0	N-mp[d|l|i]	0.05	0.07
			N-mpa	0.05	0.08
JS divergence	3.68			0.79	

Table 3: Distribution of morphological tags for noun paradigms N01 and N04 in the corpus and the distributions of tags generated by two LPPs for lemma l = *vojnik*. Bottom row shows the Jensen-Shannon divergence between the two pairs of paradigm and LPP distributions.

paradigms is rather different, although all tags have non-negative probabilities. The third and sixth column show the tag distribution conditioned on both the paradigm and the lemma l = *vojnik*. From this we can see that, for example, if *vojnik* is paired with the paradigm N01, the genitive case forms would have a high probability of 0.77, while the probability of the same forms at the paradigm level is only 0.27. Overall, tag distribution of N04 seems to provide a better fit to lemma *vojnik* than tag distribution of N01. The similarity of distributions $P(t|p)$ and $P(t|l,p)$ can be measured in a number of ways, one of them being the Jensen-Shannon divergence. In this particular case, the Jensen-Shannon divergence is much larger for paradigm N01 than for N04, providing supporting evidence that N04 is the correct paradigm.

We use the following nine corpus-based features:

1. *LemmaAttested* – a binary feature indicating whether the lemma is attested in the corpus, i.e., $\#(l, C) > 0$;
2. *Scoreo* – 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, $p_2(t) = P(t|l,p)$:

$$score_5(l,p) = -\sum_{t \in T(p)} P(t|p) \times \ln \frac{P(t|l,p)}{P(t|p)} = \mathrm{KL}(p_1||p_2)$$

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

$$score_6(l,p) = \frac{1}{2}\mathrm{KL}(p_1||p_2) + \frac{1}{2}\mathrm{KL}(p_2||p_1)$$

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, spanning years 1999 through 2009 and totaling about 400K word-form types and about 55M word-form tokens. Stop words (function words, including all closed-class words) and words occurring less than tree times in the corpus were filtered out to reduce the noise.

4.3 Other features

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

1. *ParadigmId* – a categorical (multinomial) feature denoting the LPP's inflectional paradigm;
2. *POS* – the part-of-speech of the LPP's inflectional paradigm (noun, adjective, or verb).

The intuition behind *ParadigmId* feature is that we expect a functional dependence to exist between the paradigm and the values of other features, and having *ParadigmId* as a feature allows the model to exploit this dependence. For example, it is reasonable to expect that *endsInCons* feature is relevant only for a subset of paradigms that are applicable to stems ending in a consonant. Similarly, we can expect the *Score2* feature to be less indicative for adjectival paradigms, because the proportion of corpus-attested word-form types will generally be lower for adjectives than for other parts-of-speech because comparative and superlative word-forms are less frequent in the corpus.[13] The same line of reasoning holds for the *POS* feature.

[13] Note that corpus-based features based on conditional probability $P(t|p)$ do encode this dependence. Nonetheless, the relevance and reliability of such features might vary across paradigms, thus encoding *ParadigmId* as a separate feature might still be helpful.

5 EVALUATION

In this section we turn to the evaluation of the paradigm prediction models. The purpose of evaluation is twofold: apart from determining how accurately we can predict the inflectional paradigms, we also wish to analyze what features are most useful for this task. We continue by first describing the data set, followed by feature analysis and evaluation of classification accuracy.

5.1 Data set

We compiled the data set for training and testing from the aforementioned inflectional lexicon from Šnajder et al. (2008). We sampled from the lexicon 5,000 LPPs for training and 5,000 LPPs for testing, with at least one attested word-form in the corpus. Because the distribution of paradigms is very uneven, we used stratified sampling with respect to the inflectional paradigms. Furthermore, we ensured that there is no LPP that appears in the test set, but does not appear in the training set, as otherwise the probability distributions would be undefined.

Table 4 shows the distributions and coverage of noun, adjective, and verb paradigms in the test set. The distributions follow a power-law distribution; the five most-frequent paradigms for each part-of-speech account for over 77% of types in the data set and cover over 75% word-form tokens in the corpus. Nouns make up the majority of the lexicon (67%), followed by adjectives (22.6%), and verbs (10.4%). In the corpus, however, the proportion of verbs (25.8%) is larger than that of adjectives (19.2%), again with a clear prevalence of nouns (55%).

To generate the negative training and testing instances, we proceeded as follows. For each LPP, we generate all word-forms using the function wfs (cf. Section 3.1). 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 both the training and test set. Given

Nouns			Adjectives			Verbs		
Id	**Count**	**Cov.%**	**Id**	**Count**	**Cov.%**	**Id**	**Count**	**Cov.%**
N01	920	12.3	A06	372	6.3	V17	194	11.5
N28	729	14.6	A08	329	5.0	V16	151	7.4
N22	309	4.0	A04	265	4.0	V27	32	2.2
N37	242	1.5	A10	105	0.2	V32	29	1.5
N33	142	1.5	A12	33	1.6	V13	26	1.5
N02	104	2.2	A14	8	0.3	V14	11	0.1
N10	78	1.1	A13	7	0.9	V25	9	0.1
N41	76	4.3	A11	4	0.2	V23	9	0.3
N15	75	1.0	A01	4	0.5	V21	7	0.2
N04	72	3.4	A09	2	∼0	V20	6	0.1
N39	68	0.6	A15	1	∼0	V11	6	0.1
N16	67	0.2	A07	1	0.1	V08	6	∼0
N07	62	0.9	A03	1	0.1	V02	6	∼0
N21	58	0.4				V01	6	0.1
N17	56	0.3				V24	4	0.2
N30	55	1.1				V15	4	0.2
N29	45	2.6				V09	3	0.1
N23	25	0.1				V30	2	0.1
⋮	⋮	⋮				⋮	⋮	⋮
Total	3350	55.0		1132	19.2		518	25.8

Table 4: Frequency-sorted lists of noun, adjective, and verb paradigms from the test set. *Cov.%* denotes the proportion of word-form tokens in the corpus covered by each of the paradigm.

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 sampled word-form, while the test set contains LPPs obtained from word-forms that did not appear in the training set. Also notice that the number of positive and negative instances is artificially balanced; a realistic data set would contain about 17 incorrect LPPs for each correct LPP. We chose to balance the data set because SVM tends to perform poorly on imbalanced data sets (Wu and Chang 2003).

5.2 Feature analysis

Some of the features we defined are redundant or perhaps irrelevant for LPP prediction. 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 prediction.

For feature analysis we used the open source tool Weka (Hall et al. 2009). We used three univariate filtering methods: information gain (IG), gain ratio (GR), and the RELIEF method. The univariate filtering methods determine the relevance of features based on the intrinsic properties of the data; a statistical test is applied to each individual feature in order to determine its importance, features are ranked accordingly, and a desired number of top-ranked features is then chosen. Among the three considered methods, RELIEF (Kira and Rendell 1992; Kononenko 1994) is probably the most efficient. RELIEF works by iteratively estimating the feature weights based on their ability to discriminate between neighboring instances in the input space.[14]

Table 5 summarizes the feature analysis results. 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, and 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*. Individually, these features seem to be less relevant for paradigm prediction, according to the methods we used.

The univariate methods do not measure the dependencies between the features, thus they cannot detect feature redundancy. We therefore also analyzed

[14] More recently, Sun and Li (2006) have shown that RELIEF is less heuristic than initially thought, and in fact solves a margin optimization problem based on the nearest neighbor classifier.

	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 5: Feature selection analysis with univariate filtering (Ranking) and multivariate feature subset selection (FSS).

the features using two multivariate feature subset selection (FSS) methods: correlation-based feature selection (CFS) (Hall 1998) and consistency subset selection (CSS) (Liu and Setiono 1996), both with greedy forward search as the optimization method. Table 5 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

We conducted two experiments to evaluate the classification accuracy of our models. In the first experiment, we evaluate the binary classification accuracy, which is in line with how we formulated the problem of paradigm prediction. In the second experiment, we consider a more realistic setting and evaluate classification accuracy on a per word basis.

5.3.1 BINARY CLASSIFICATION ACCURACY

In the first experiment, 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. Table 6 shows classification accuracy on the test set. 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. To analyze how models would perform in such cases, we evaluated on three frequency bands: all LPPs, 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.

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. Furthermore, 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 only slightly worse than the model using the full set of 22 features.

Features	Count	Word-forms attested		
		≥ 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

Table 6: Paradigm classification accuracy (%) for models with different feature subsets, for three different frequency bins of the word-forms

5.3.2 PER WORD CLASSIFICATION PERFORMANCE

Binary classification accuracy gives us an insight into how models with different feature sets compare against each other. In practice, however, we are not interested in binary classification per se, but choosing the correct LPP among the set of LPP candidates (choosing one among about 17). Thus, a more realistic evaluation would consider model performance on a per word basis. To this end, we built another test set comprised of 1,000 LPPs sampled from 5,000 correct LPPs that we used for testing in the first experiment (cf. Section 5.1). The set contains 654 noun LPPs, 233 adjective LPPs, and 113 verb LPPs. For each LPP from this set, we generated the incorrect LPPs by choosing at random one word-form from the set $wfs(l, p)$ and applying on it the $lm(w)$ function to obtain all its LPP candidates. In this way we obtain for each word-form its correct LPP and all its incorrect LPPs.[15] The final data set contains 17,111 LPPs. On this set we compute the model performance in terms of standard information retrieval measures of precision (P), recall (R), and (micro-averaged) F1-score (van Rijsbergen 1979). Precision score of 100% would mean that no incorrect LPP has been classified

[15] Here we ignore the fact that homographs have more than one correct LPP. This only marginally affects the precision scores.

	All			Nouns			Adjectives			Verbs		
Features	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
All	37.4	91.6	53.1	37.0	91.9	52.3	40.7	94.8	57.0	33.0	83.2	47.2
RELIEF	32.3	90.3	47.6	31.6	89.3	46.6	37.1	96.1	53.5	28.0	84.1	42.0
Baseline	15.9	82.8	26.7	13.7	79.7	23.4	18.1	90.6	30.2	41.9	85.0	56.1

Table 7: Per word classification performance of all-features model and 5-features model across different parts-of-speech.

as correct, while a recall of 100% would mean that all correct LPPs have been identified as such.

Table 7 shows the results for the two best-performing models from the first experiment: the all-features model and the model that uses five features selected by RELIEF. As the baseline, we use the $score_1$ function (cf. Section 4.2), which predicts as correct the LPPs with the largest sum of word-form frequencies from corpus. Overall, the all-features model performs best with an F-score of above 53%. Both models have a relatively high recall and a comparatively low precision across all parts-of-speech. Using only five features leads to a 5 point drop in precision. Both models perform best on adjectives and worst on verbs. Moreover, both models outperform the baseline, except on the verbs, for which the simple frequency-based scoring seems to be a competitive baseline. This suggests that incorrect LPPs of verbs tend to generate word-forms that have less evidence in the corpus. In contrast, incorrect LPPs of nouns and adjectives often generate valid word-forms. Notice, however, that in practice we do not know a word-form's true part-of-speech, thus we cannot make use of this information.

Overall per word precision is 37%, which means that for each word-form one or two LPPs will be wrongly classified as correct. The reason for this is that the problem is formulated as binary classification. Binary LPP classification makes locally optimal decisions, without considering the set of LPP candidates as a whole and the constraint that only one LPP may be correct.[16] This also

[16] Again, this is the case if we ignore the homographs.

	All		Nouns		Adjectives		Verbs	
Features	**Acc**	**Ties**	**Acc**	**Ties**	**Acc**	**Ties**	**Acc**	**Ties**
All	70.2	2	67.7	2	79.4	0	65.5	0
RELIEF	66.5	1	63.3	1	72.5	0	72.6	0
Baseline	34.0	655	23.9	489	40.5	155	78.9	11

Table 8: Per word classification accuracy (top-ranked LPP) and the number of ties of all-features model and 5-features model across different parts-of-speech.

explains why for verbs the baseline model, which chooses only the top-scored LPP, achieves a larger precision than binary classifiers, which consider each LPP in isolation.

By considering the confidence scores of LPP candidates we could in principle make more informed classification decisions, which could improve the precision. A straightforward approach is to choose, for each word-form, only the top-ranked LPP as the correct one. Table 8 shows the results for this setting. Because we choose only one LPP per word, computing precision and recall would make no sense here, thus we compute the accuracy as the number of correctly predicted paradigms averaged over the number of word-forms in the sample. In case of ties (cases in which two or more LPPs are predicted the same score), we compute partial credits: 1/2 score for two-way ties, 1/3 score for three-way ties, etc. In Table 8 we also show the total number of words for which there are ties. The overall accuracy is 70.2% and 66.5% for all-features model and 5-features model, respectively. The accuracy is again the best on adjectives and the worst on verbs. Both models outperform the $score_1$ baseline by a wide margin on nouns and adjective, but not on verbs. Because the models output probability estimates, there are not many ties. In contrast, the baseline output values are more coarse-grained and therefore there are many more ties.

5.4 Remarks

The above results have raised several issues that deserve further comments.

Choice of corpus. In this work we used a newspaper corpus, and it is possible that this choice has an effect on the overall prediction accuracy. As noted by one reviewer, a newspaper corpus is not likely to contain many aorist and imperfect verb forms, which would add significantly to homography. Although the grammar we used does not model the aorist and imperfect verb forms, the argument still applies to vocative noun forms, which would also add to homography but are likewise underrepresented in newspaper corpora. Although this issue deserves further examination, from a practical viewpoint it is indeed reasonable to use a corpus that minimizes homography, as this will improve the overall prediction accuracy.

Choice of grammar. Perhaps a more interesting question is the choice of the grammar. What might be of particular importance for paradigm prediction in Croatian is the modeling of verbs, more concretely the question of how to treat adjectival and adverbial participles (cf. footnote 5). In the current implementation of the HOFM grammar, the adjectival participles are not included in the verb paradigm. However, including them into the verb paradigm might allow for better prediction of verb paradigms. We leave this issue for future work. Another issue is the level of grammar ambiguity. HOFM defines applicability conditions for many paradigms; a grammar that does not define such conditions would overgenerate more, leading to a decrease in precision. This suggest that paradigm prediction performance is dependent on the specific grammar used and perhaps does not readily generalize across different grammars.

Training set selection. 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 prediction to acquire such a lexicon. Related to this is the question of how many instances per paradigm we need to train a good classifier. The active learning framework provides a way to minimize the number of training instances and hence reduce the manual labeling efforts. Active learning may also be combined with ranking-based classification to speed up the annotation process.

Semi-automatic lexicon acquisition. Probably the most interesting application of paradigm prediction is semi-automatic lexicon acquisition. In this setting, confidence-ranked lists of LPP candidates are presented to an expert, who then identifies the correct LPP, which ideally should be ranked first. In this setting it would make sense to evaluate paradigm prediction as a ranking task. There are also a number of other factors that should be considered, such as the presence of noise in the corpus (i.e., words for which no correct LPP exists and which should be rejected), treatment of proper names, and the workflow parameters (e.g., in what order the word-forms should be processed, is the model being updated based on the input from the expert, etc.).

Other evaluation scenarios. There are a couple of other 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.

6 CONCLUSION AND PERSPECTIVES

Being able to determine the inflectional paradigm of an unknown word is important for morphological analysis of highly inflectional languages. We have addressed the problem of paradigm prediction for 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 SVM 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 (paradigm label, two string-based features and two corpus-based features). Degradation in classification performance on infrequent words is minimal. When evaluated on a per word basis, the all-features model achieves 53% of F1-score and 70% of accuracy, outperforming the frequency-based baseline by a wide margin. The models perform best on adjectives and worst on verbs.

This work provides a basis for further research. Our first priority will be to apply paradigm prediction to semi-automatic lexicon acquisition and carry out a comprehensive task-based evaluation in this setting. From a machine learning perspective, we will consider using additional features, such as part-of-speech tags and capitalization features.

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under Grant 036-1300646-1986. We thank the two anonymous reviewers for their helpful comments.

BIBLIOGRAPHY

Adolphs, P. (2008): Acquiring a poor man's inflectional lexicon for German. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*: 3414–3419.

Aronoff, M. and Fudeman, K. (2011): *What is morphology*: volume 8. Wiley-Blackwell.

Calder, J. (1989): Paradigmatic morphology. In *Proceedings of the Fourth Conference on European Chapter of the Association for Computational Linguistics*: 58–65. Association for Computational Linguistics.

Chan, E. (2006): Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special*

Interest Group on Computational Phonology and Morphology: 69–78. Association for Computational Linguistics.

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

Cholakov, K. and Van Noord, G. (2009): Combining finite state and corpus-based techniques for unknown word prediction. In *Proceedings of the 7th Recent Advances in Natural Language Processing (RANLP-09) conference*.

Clement, L., Sagot, B., and Lang, B. (2004): Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*: 1841–1844.

Dreyer, M. and Eisner, J. (2011): Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*: 616–627. Association for Computational Linguistics.

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

Esplá-Gomis, M., Sánchez-Cartagena, V., and Pérez-Ortiz, J. (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)*: 411–415.

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

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

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

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

Hammarström, H. and Borin, L. (2011): Unsupervised learning of morphology. *Computational Linguistics*, 37(2): 309–350.

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

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

Jones, S. P. (2003): Haskell 98 language and libraries: The revised report. Technical report.

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

Kira, K. and Rendell, L. A. (1992): A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*: 249–256. Morgan Kaufmann Publishers Inc.

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

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

Lindén, K. (2009): Entry generation by analogy—encoding new words for morphological lexicons. *Northern European Journal of Language Technology*, 1(1): 1–25.

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

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

Monson, C., Carbonell, J., Lavie, A., and Levin, L. (2008): ParaMor: Finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval*: 900–907. Springer.

Nakov, P., Bonev, Y., Angelova, G., Cius, E., and Von Hahn, W. (2004): Guessing morphological classes of unknown German nouns. *Recent Advances in Natural Language Processing III (RANLP-03)*, 347–356.

Oliver, A. (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*.

Oliver, A. and Tadić, M. (2004): Enlarging the Croatian morphological lexicon by automatic lexical acquisition from raw corpora. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*: 1259–1262.

Peradin, H. and Šnajder, J. (2012): Towards a constraint grammar based morphological tagger for Croatian. *Lecture Notes in Computer Science*, 7499: 174–182.

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

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

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

Šnajder, J. and Dalbelo Bašić, B. (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*: 121–130: Dubrovnik, Croatia. Croatian Language Technologies Society.

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

Sun, Y. and Li, J. (2006): Iterative relief for feature weighting. In *Proceedings of the 23rd international conference on Machine learning*: 913–920. ACM.

Tadić, M. (2002): Building the Croatian national corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02)*: 441–446.

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

van Rijsbergen, C. J. (1979): *Informaton Retrieval*. Butterworths, London.

Vapnik, V. (1999): *The nature of statistical learning theory*. Springer.

Wu, G. and Chang, E. Y. (2003): Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*: 49–56.

MODELI ZA PREDIKCIJO OBLIKOSLOVNIH PARADIGEM ZA HRVAŠKE BESEDE

Morfološka analiza je predpogoj za številne naloge pri računalniški obdelavi jezika. Pri oblikoslovno bogatih jezikih, kot je hrvaščina, temelji morfološka analiza navadno na morfološkem leksikonu, ki vsebuje seznam lem in njihove oblikoslovne paradigme. Vendar pa mora uporaben morfološki analizator znati ustrezno razčleniti tudi besede, ki jih ni v leksikonu. V prispevku se lotevamo avtomatskega prepoznavanja ustrezne oblikoslovne paradigme pri še neznanih hrvaških besedah. Problem obravnavamo z nadzorovanim strojnim učenjem, kjer na osnovi vrste besednih in korpusnih značilk klasifikator naučimo predvidevati, ali je določen par lema–paradigma ustrezen. Pare lema-paradigma smo generirali s pomočjo ročno izdelane oblikoslovne gramatike. Namen prispevka je analizirati postopke strojnega učenja pri obravnavi tega problema: testirali smo bogat nabor značilk in ocenili natančnost klasifikacije z uporabo različnih podmnožic značilk. Pokažemo, da je zadovoljivo natančnost klasifikacije (92 %) mogoče doseči z metodo SVM in z uporabo kombinacije besednih in korpusnih značilk. Dosežena natančnost za posamezno besedo v našem modelu je 70 %, vrednost F1 je 53 %, kar je bistveno boljše kot rezultat, ki upošteva samo pogostost pojavitev. Članek zaključimo s smernicami za nadaljnje delo.

Ključne besede: računalniška morfologija, predikcija oblikoslovnih paradigem, strojno učenje, izbor značilk, hrvaški jezik.

To delo je ponujeno pod licenco Creative Commons: Priznanje avtorstva – Deljenje pod enakimi pogoji 2.5 Slovenija.

This work is licensed under the Creative Commons Attribution ShareAlike 2.5 License Slovenia.

http://creativecommons.org/licenses/by-sa/2.5/si/