Predicting Inflectional Paradigms and Lemmata of Unknown Words for Semi-automatic Expansion of Morphological Lexicons

Nikola Ljubešić
University of Zagreb
nljubesi@ffzg.hr

Miquel Esplà-Gomis
Universitat d'Alacant
mespla@dlsi.ua.es

Filip Klubička
University of Zagreb
fklubick@ffzg.hr

Nives Mikelić Preradović
University of Zagreb
nmikelic@ffzg.hr

Abstract

In this paper we describe a semi-automated approach to extend morphological lexicons by defining the prediction of the correct inflectional paradigm and the lemma for an unknown word as a supervised ranking task trained on an already existing lexicon. While most ranking approaches rely only on heuristics based on a single information source, our predictor uses hundreds of features calculated on the candidate stem, corpus evidence and statistics calculated from the existing lexicon. On the example of the Croatian language we show that our approach significantly outperforms a heuristic-based baseline, yielding correct candidates in 77% of cases on the first position and in 95% of cases on the first five positions.

1 Introduction

Morphological lexicons are a vital resource in automatic processing of morphologically rich languages and their construction is a tedious and costly process.

The most reasonable approach to organizing inflectional morphological lexicons of morphologically rich languages is to define inflectional paradigms and assign them to corresponding lexemes. In this way, every entry in the morphological lexicon becomes a pair (l, p) of a lemma l and a paradigm p which allow to derive all the possible surface forms of a given word.

In this paper we frame the problem of assisting a process of extending an existing morphological lexicon as a supervised ranking problem. Namely, for each unknown word of a language we generate all possible pairs (l, p) and rank them with the goal of positioning existing pairs as high as possible. The result of the raking process is presented to a linguist through a graphical interface for labelling the correct pairs (l, p).drastically lower than would be the case if the lexicon was built manually.

2 Related Work

A significant amount of research has focused on the problem of enhancing the process of producing morphological resources. The most widely used approach is ranking pairs (l, p) of lemmas and paradigms by various scoring functions which rely on corpus evidence, the most popular being the coverage of all inflected forms derived from a pair (l, p) in a given monolingual corpus (Clément et al., 2004; Tadić and Oliver, 2004; Sagot, 2005; Šnajder et al., 2008; Esplà-Gomis et al., 2011). While our approach follows the same ranking paradigm, we argue that a significant amount of additional information can be gained from corpora and other information sources, supervised machine learning being the obvious solution for combining those.

The approach by Lindén (2009) does not rely on corpus evidence only, but uses the existing lexicon as well, showing that by combining corpus and lexicon evidence significant gains can be achieved.

The first approach to exploit machine learning over multiple sources of information for extending morphological lexicons is the work of Kaufmann and Pfister (2010) who use the information from a morphological lexicon, a morphological grammar and a corpus, and combine it via a machine-learning approach to guess the stem and morphosyntactic information for unknown words. Using a different approach, Ahlberg et al. (2014) learns paradigms from an initial collection of inflection tables, and new words are assigned to these paradigms by using a confidence score. This approach is later extended by Ahlberg et al. (2015) to use multi-class classification (using support vector machines) for choosing the best paradigm. In this work, all the possible suffixes and prefixes from a given surface form are used as binary features,

Proceedings of Recent Advances in Natural Language Processing, pages 379–387,
Hissar, Bulgaria, Sep 7–9 2015.

after applying feature selection in order to optimise the performance.

Regarding supervised approaches, it is worth noting the work by Durrett and DeNero (2013), in which patterns are built from morphologically analysed corpora to infer paradigms. For a given new surface form, they are applied in order to obtain all the inflections, and a hidden Markov model is used to choose the likeliest paradigm.

The work most similar to ours, on which we build upon, is the one by Šnajder (2012) who defines a set of string and corpus features and exploits them in a supervised learning setting, framing the problem as a binary classification task, i.e. predicting whether a candidate pair (l, p) is correct or not. This approach enables both a fully automatic lexicon construction process and the fact that a surface form can be a realisation of more than one (l, p) pair. However, results show that, although quite a high accuracy of 92% is reported (on an artificially balanced dataset), the approach is not sufficient for the positively labeled instances to be included in a morphological lexicon without human inspection, while exposing linguists to a collection of pairs (l, p) that are classified as correct is far from optimal as, in case of a false positive, alternatives are not given.

Our approach tries to facilitate the best of the two worlds – producing a ranked output for every unknown word as this is the optimal representation for the necessary human inspection, and combining all available information sources and the supervised learning paradigm to produce an output with the highest quality possible.

The remainder of the paper is structured as follows: in the following section we describe the components of our method. Section 4 describes the experimental setting while Section 5 gives the discussion of the results of the experiments. The paper ends with the conclusions in Section 6.

3 The Method

Our approach for producing a ranked list of candidate pairs (l, p) for each unknown word consists of three steps: 1) generating candidates; 2) extracting features from each candidate; and 3) ranking the candidates by supervised learning. We describe those in detail in the remainder of this section.

3.1 Candidate Generation

When we want to add an unknown surface form to a morphological lexicon we first need to know which pairs (l, p) are compatible with it. In this work, we focus on languages using suffixing for morphological inflection. This strategy is the most frequent for languages all around the world (Dryer, 2013), and it is the one specifically used by Croatian, which is our case of study. For suffixing languages, a paradigm in a morphological lexicon adds suffixes to a given stem in order to produce surface forms. Therefore, a good hint to find out the candidate paradigms from an unknown word is the inflection suffix. Unfortunately, finding out which is the suffix of a surface form without knowing its paradigm is not straightforward. Our strategy consists in checking which suffixes in the whole collection of suffixes generated by all the paradigms in a morphological lexicon match the unknown word, so we can obtain a collection of $(\mathrm{stem}, \mathrm{suffix})$ candidate pairs (l, p). Having these candidates it is possible to identify which paradigms produce the suffixes and consequently to obtain a collection of candidate pairs (l, p).

To simplify the search of candidate suffixes for a given unknown word, we use a *generalised suffix tree* (McCreight, 1976) containing all the possible suffixes from the paradigms in our lexicon.[1] Each of these suffixes is labeled with the index of the corresponding paradigms that can produce it. The generalised suffix tree data structure allows to retrieve the paradigms compatible with an unknown word by efficiently searching for all the compatible suffixes; when a suffix is found, the collection of paradigms generating it is retrieved and the list of candidates is enlarged with the new pairs (l, p).

3.2 Ranking the Candidates

Our approach is aimed at producing a ranked list of candidate pairs (l, p) for a given unknown surface form to be added to the lexicon. To do so, we use a binary classification approach which classifies each candidate pair (l, p) as either correct or incorrect, as well as a certainty measure for the candidate pair to belong to the positive class. We finally use that certainty measure to rank our candidates from the most suitable to the least suitable one.

To train our prediction models we define several features by which each candidate in our dataset is

[1] Note that this method could be easily adapted to prefixing languages by using a prefix tree instead a suffix tree.

represented. A significant part of the features we use are those proven to be informative by Šnajder (2012). We extended that list of features with those using probabilities of paradigm-conditioned suffixes of different length, probabilities of paradigm-conditioned prefixes, coverage of morphosyntactic classes, and coverage of surface forms tagged in the corpus with the corresponding morphosyntactic description (MSD).

The rest of this section describes the specific groups of features.

3.2.1 Stem Features

Stem features capture information about the stem obtained from the surface form after removing the suffix according to the pair (l, p) to be evaluated. These features are the following:

EndsIn – categorical feature containing the last character of the stem
EndsInCons – binary feature whether the stem ends with a consonant
EndsInPals – binary feature whether the stem ends with a palatal voice
EndsInVelars – binary feature whether the stem ends with a velar voice
NumSyllables – number of the syllables of the stem
OneSyllable – binary feature whether the stem contains one syllable only
StemLength – length of the stem

3.2.2 Lexicon Features

The lexicon features represent the information from the existing lexicon about the relation between a paradigm and suffixes and prefixes of stems and lemmata that belong to that paradigm. This information is encoded as paradigm-conditioned probabilities of affixes of length n, i.e. $P(\text{affix}_n|\text{paradigm})$. The features are the following:

LemmaSuffixProb_n – probability of a lemma suffix of length n given the paradigm
StemSuffixProb_n – probability of a stem suffix of length n given the paradigm
StemPrefixProb_n – probability of a stem prefix of length n given the paradigm

For each of these features $n \in \{1, 2, 3\}$, meaning that there are all together 9 different lexicon features.

3.2.3 Corpus Features

The corpus features are extracted from an external monolingual corpus. If such a corpus is available, it can be used to confirm the existence of the word forms derived from the pair (l, p) and to measure whether the observed frequency distribution of different forms is close to the expected one as calculated on existing lexicon entries. Additionally, we propose here to use a morphosyntactically annotated corpus which allows us to indirectly introduce the contextual information used by the tagger in its decision process. The corpus features are the following:

Freq – corpus frequency of the unknown word
LemmaAttested – binary feature whether the candidate lemma was attested in the corpus
NumAttForms – number of attested word forms from the expanded candidate paradigm
NumAttTags – number of morphosyntactic tags with at least one attested word form
PropAttForms – proportion of attested word forms
PropAttTags – proportion of morphosyntactic tags with at least one attested word form
PropAttFormsPoS – proportion of attested words forms tagged with the corresponding PoS
PropAttFormsMSD – proportion of attested words forms tagged with the corresponding morphosyntactic description
SumAttForms – sumation of corpus frequencies of word forms generated
SimTagDistrJS – Jensen-Shannon divergence between the expected paradigm-conditioned probability distribution of morphosyntactic categories (measured on the training portion of the existing lexicon and the corpus) and the observed probability distribution of morphosyntactic categories of the candidate (measured on the candidate and the corpus)
SimTagDistrCos – cosine distance of distributions used to obtain SimTagDistrJS

3.2.4 Other Features

Two categorical features are included in this category: the paradigm and the part-of-speech (PoS) of a given candidate. These features enable the model to capture the a-priori probability of each paradigm and PoS and possible dependences of other features on the paradigm or PoS. To clarify the latter with an example, the number of syllables of a stem is hardly a good predictor of the correctness of a

candidate if it is not joined with the information on the paradigm of the candidate. Namely, there are paradigms that prefer stems with a specific number of syllables.

4 Experimental Setting

4.1 The Datasets

The two main sources of information we use in building our system are an existing morphological lexicon of Croatian and a corpus of Croatian. While we use both for extracting features (see Sections 3.2.2 and 3.2.3), we use the lexicon for producing the annotated dataset we train our predictor on.

4.1.1 The Lexicon

The morphological lexicon of Croatian we use in our experiments is part of the Apertium rule-based machine translation system (Forcada et al., 2011). It is the only freely available morphological lexicon of Croatian which contains both definitions of paradigms and lexemes attached to these paradigms.[2]

At the time we ran our experiments, the lexicon consisted of 413 paradigms from open-word classes, out of which 204 were noun paradigms, 167 were verbal and 42 adjectival. There were 10,183 lexemes in the lexicon annotated with one of the 413 paradigms. The whole lexicon was, up to that point, produced manually by the members of the Apertium community.

These lexemes produce almost 70 thousand different surface forms. Once those surface forms are used to generate all candidate pairs (l, c), which are the instances we perform classification and ranking on, we end up with around 7 million instances, with a ratio of positive and negative examples of 1:100.

Given that the amount of the available training data is huge, we randomly split the existing lexicon in two parts: 80% of the lexical entries were used for development, while the remaining 20% of the entries were put aside for testing.

The final development set contains 55,458 surface forms, while the test set consists of 12,089 surface forms. Each of these surface form has at least one pair (l, p) from which it could be derived. While 90% surface forms can be only derived from one pair (l, p), 9% can be derived from two of them and the remaining 1% can be derived from up to 7 pairs (l, p).[3] Generating candidate pairs (l, p) for the surface forms produced 6.1 million development and 1.3 million testing instances.

4.1.2 The Corpus

For gathering corpus evidence we used the largest available corpus of Croatian: the second version of the Croatian web corpus *hrWaC* (Ljubešić and Klubička, 2014), consisting of 2 billion words. The corpus is morphosyntactically tagged and lemmatised (Agić et al., 2013) with tools trained on a 90k-token training corpus (Agić and Ljubešić, 2014).

4.2 The Classifiers

We consider two classifiers for our task: support vector machines (SVM) and Random Forests (RF). While SVM has proven to be the best performing classifier on many different problems, the strengths of RF are comparable prediction strength and much higher speed. We use the Scikit-learn implementations of the two classifiers (Pedregosa et al., 2011).

Given that the RF classifier is a stochastic process, each experiment on that classifier is run 10 times and we report the mean and standard deviation of the scoring function.

For optimising our binary classifiers, we use randomised search for RF as the number of hyperparameters is quite high, while we perform grid search on SVM with the RBF kernel.

During classifier optimisation we use the F1 of the positive class as our scoring function since the dataset is highly unbalanced, having for each positive instance 100 negative ones.

4.3 Ranking

For producing ranked results we opt for the simple pointwise ranking approach in which we use certainty of the positive class on the binary classification problem as our ranking function.

We do not take pairwise ranking under consideration as we expect 1.1 correct answers among 100 candidates, making the computational cost of a drastically higher number of necessary classifications for pairwise ranking hard to argue for.

We perform ranking with both of our classifiers. In case of RF we rank the candidates by the descending probability of the positive class, while

[2] `http://sourceforge.net/p/apertium/svn/HEAD/tree/languages/apertium-hbs/`

[3] The high amount of surface forms which can be derived from two paradigms can be explained by the fact that different verbal paradigms exist regarding verb's aspect, transitivity and reflexivity. Consequently, if a verb is biaspectual, it is given two paradigms.

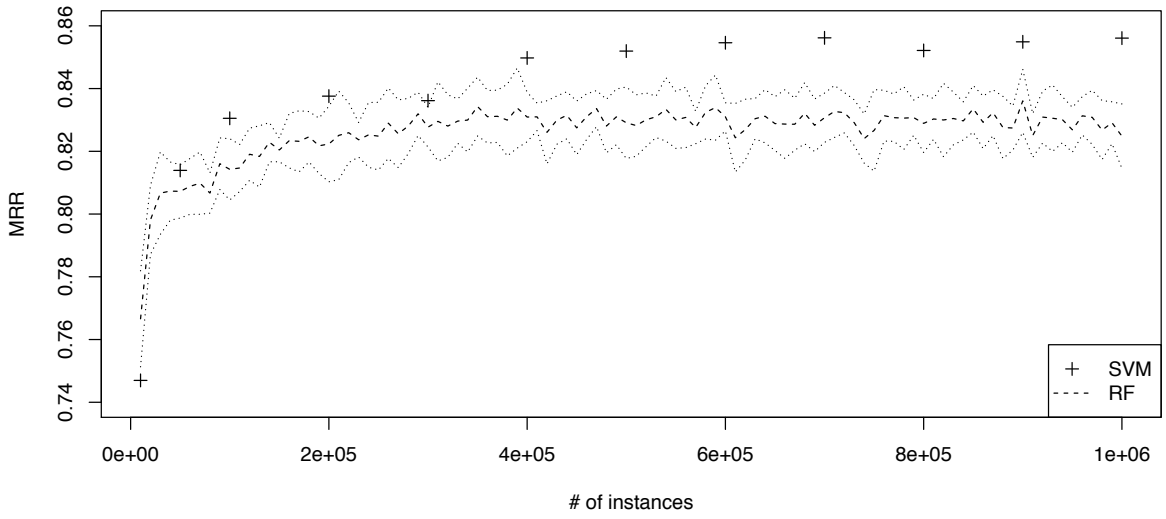

Figure 1: Ranking performance of both classifiers as a function of training data size

with SVMs we use the descending distance of each instance to the separating hyperplane.

We evaluate the ranking results via mean reciprocal rank (MRR) (Craswell, 2009) as for most of the surface forms there exists only one correct candidate pair. While reciprocal rank of a ranked result is the multiplicative inverse of the position of the (first) correct pair (l, p) in the ranking, the MRR is the average of reciprocal ranks of all the ranked results.

As our heuristic-based baseline, we take the scoring function from Esplà-Gomis et al. (2011). Given a pair (l, p), this approach produces the collection of surface forms that can be derived from it and calculates a confidence score based on the number of these surface forms attested in the corpus.

5 Results

We perform two sets of experiments. In the first set, we optimise both classifiers on the binary classification task, using F1 score on the positive class as our scoring function. In the second set of experiments we use the optimised classifiers for pointwise ranking, using MRR as our scoring function.

5.1 Classification

Given that optimising classifiers on multiple millions of instances would be extremely time-consuming, we limit our development data on 500 thousand instances, as it showed to produce stable results during our early experiments. On both classifiers we perform optimisation via 10-fold cross-validation on the development data. We perform a final evaluation of the optimised classifiers on our test data.

The result on the binary classification task obtained on the test data for SVM is 70.4% and for RF 59.8$\pm$2.4%. Regarding the time necessary for training and annotating the test set, SVM takes 215.86 and 99.04 seconds, while RF takes 3.28 and 0.46 seconds.

These results show quite clearly that, while the RF classifier is magnitudes faster on both training and testing, SVM outperforms RF with a wide margin.

5.2 Ranking

In the first ranking experiment we compare the two optimised classifiers while taking into account the amount of data used for training. We plot the results in form of learning curves in Figure 1. For RF we vary the training data size from 10 thousand instances to 1 million instances in 10k-size steps. Given that the training time for SVM is much higher than for RF, we evaluate SVMs by increasing the amount of training data by 100k instances.

The results show that on the pointwise ranking task SVM still outperforms RF, but not as drastically as on the classification task.

Regarding the impact of the amount of training data on the ranking output, we observe a steep learning curve up to 100k learning instances (climb-

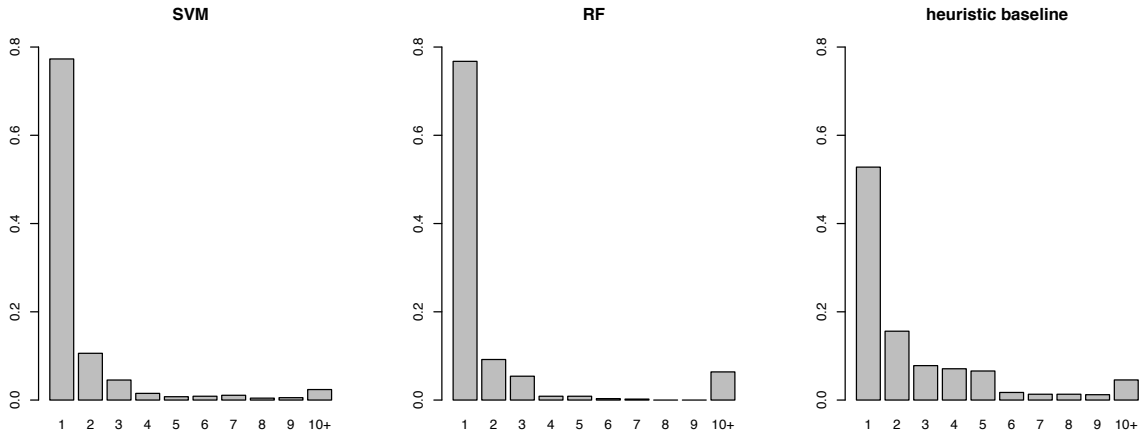

Figure 2: Distribution of positions of first correct candidates with both classifiers and the heuristic baseline

ing up to 0.831 MRR with SVM), with a moderate increase in MRR up to 500k (0.852 with SVM). The improvement obtained when doubling the amount of training data to one million instances is just 0.004 MRR (0.856 with SVM). Therefore we will perform the remainder of our experiments by using 500k training instances.

The heuristic-based baseline (Esplà-Gomis et al., 2011) does not depend on the amount of training data and produces an MRR of 0.674. Therefore we can conclude that our machine learning approaches significantly outperform our heuristic baseline.

5.3 Analysis of the Results

In this section we perform a deeper analysis of the ranking results obtained by using 500k training instances.

In Figure 2 we plot the distribution of the position of the first correct candidate for both our classifiers and compare it to our heuristic-based baseline. With both classifiers we position the correct candidate on the first position in 77% of cases. A slight difference in the classifier performance can be seen when we compare the percentage of surface forms for which a correct candidate can be found on the first three positions, where SVM reports 92.4% and RF 91.4%. If we assume that a human annotator can easily inspect the first 5 positions, correct candidates can be found with SVM in 94.7% and with RF in 93.1% of cases.

The heuristic-based approach shows significantly worse results, actually quite worse than reported in (Esplà-Gomis et al., 2011), which is due to the much higher morphological complexity of the language used in these experiments.While for only 52.8% of surface forms correct candidates are found on the first position, the first three and five positions contain correct candidates for 77.1% and 90.9% of surface forms respectively.

part of speech	RF	SVM
all	0.833 ± 0.004	0.852
noun	0.816 ± 0.010	0.827
verb	0.778 ± 0.009	0.838
adjective	0.935 ± 0.007	0.903

Table 1: Ranking performance by part of speech

In Table 1 we show the MRR score obtained on each part-of-speech of the surface form. The noticeably best results are obtained on adjectives, which is to be expected as their inflection is quite regular in Croatian. Worst results are obtained on verbs although nouns have the highest number of candidate paradigms. This can be explained by a much more complex inflectional system of verbs, part of which is used very infrequently.

Interestingly, the more successful SVM classifier performs slightly better on ”hard” parts-of-speech, especially verbs, while RF outperforms SVM on the ”easy” adjectival class.

5.4 Feature Analysis

In this section we inspect the impact of the defined features on our task by measuring the loss in MRR as they are either removed or used exclusively. Given the large number of features and the consequently large number of necessary experiments, we perform these with the faster RF classifier only.

	except	only
stem	0.708 ± 0.012	0.452 ± 0.001
corpus	$\mathbf{0.651 \pm 0.009}$	$\mathbf{0.737 \pm 0.006}$
lexicon	0.865 ± 0.007	0.398 ± 0.003
other	0.818 ± 0.010	0.569 ± 0.000

Table 2: Ranking performance of RF as features of a specific group are either removed (except) or used exclusively (only)

In the first experiment we either remove a feature group, as defined in section 3.2, or remove all but that feature group. The results of this experiment are shown in Table 2.

While removing feature groups, a significant loss in performance can be observed when removing corpus features. When removing lexicon features, a slight increase in MRR can be observed pointing to the conclusion that performing feature reduction on this feature group should be performed. First insights point to the conclusion that the features deteriorating our predictions are StemPrefix_2 and StemPrefix_3 while all the remaining features of this group improve our predictions. We leave this task for future work.

On the other hand, when using feature groups exclusively, i.e. using each of them separately, the best performance is obtained with corpus features. Lexicon features show to be of least help when used alone.

A reasonable performance is obtained when using the "other" feature group only. This group models the a priori probability of a paradigm given its part-of-speech and can be considered the most-frequent-paradigm baseline.

In the final experiment we focus on the most informative group of features – the corpus group. Again, we run experiments when removing a specific feature, or when training our predictor on that feature only.

The first thing we observe is that proportions of attested entities alone, as one would expect, are more informative than the numbers of the same. The most informative type of corpus information when used alone is the proportion of attested forms, outperforming the proportion of attested tags. Using annotated corpora, i.e. constraining attested forms only to those annotated with the expected morphosyntactic description (MRR 0.646) or part of speech (MRR 0.619), does outperform using raw text only (MRR 0.593). Distribution distances alone are not very informative (MRR 0.185), but generate the biggest loss in MRR once they are removed, proving the uniqueness of the information they provide.

5.5 Linguist Speed Improvements

A final inquiry was made in the speed improvements obtained by using the presented tool. A linguist very well acquainted with the paradigms at his disposal required on average 66 seconds for an entry when not using the tool, 76 seconds when using the candidate generator without the ranker, and 42 seconds when using both. From this we conclude that the tool brings a productivity increase by a factor of 1.6 while presenting unranked candidates does not bring any productivity gains.

6 Conclusion

In this paper we have presented a supervised ranking approach to assisting the expansion of an existing morphological lexicon. We have shown that such approach outperforms the traditional heuristic-based scoring approach by a wide margin.

We have used two classifiers during our experiments, one more accurate, the other much faster. While SVM does perform better than RF, in a production scenario the difference is not crucial and if computational capacity is limited, one should opt for RF.

An inspection of specific types of features showed the corpus type to be the most informative. Inside that feature type the proportion of attested word forms that are tagged in the corpus with the expected morphosyntactic description is proven to be the overall most informative feature.

An initial inquiry in speed gains when using the predictor showed to increase the linguists productivity by a factor of 1.6. A potential increase in accuracy has to be verified with future experiments.

Future work should also include a feature selection process. Namely, we have noticed that, regardless of using classifiers that perform implicit feature selection, there are some features among the lexicon-based ones that do deteriorate our results.

Finally, as the features using annotations from the corpora have shown to be more informative than those using raw text only, additional features using that information source should be added to the feature space.

Acknowledgments

The research leading to these results has received funding from the European Fund for Regional Development 2007-2013 under grant agreement no. RC.2.2.08-0050 (project RAPUT) and from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no. PIAP-GA-2012-324414 (project Abu-MaTran).

References

Željko Agić and Nikola Ljubešić. 2014. The SETimes.HR linguistically annotated corpus of Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, Reykjavik, Iceland.

Željko Agić, Nikola Ljubešić, and Danijela Merkler. 2013. Lemmatization and morphosyntactic tagging of croatian and serbian. In *Proceedings of the Fourth Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 48–57, Sofia, Bulgaria, August.

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, Gothenburg, Sweden.

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, USA.

Lionel Clément, Bernard Lang, and Benoît Sagot. 2004. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, LREC'04, pages 1841–1844, Lisbon, Portugal, May.

Nick Craswell. 2009. Mean reciprocal rank. In Ling Liu and M.Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1703–1703. Springer US.

Matthew S. Dryer, 2013. *The World Atlas of Language Structures Online*, chapter "Prefixing vs. Suffixing in Inflectional Morphology". Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 1185–1195, Atlanta, USA.

Miquel Esplà-Gomis, Víctor M. Sánchez-Cartagena, and Juan Antonio 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'11, pages 339–346, Hissar, Bulgaria, September.

Mikel L. Forcada, Mireia Ginest-Rosell, Jacob Nordfalk, Jim ORegan, Sergio Ortiz-Rojas, Juan Antonio Prez-Ortiz, Felipe Snchez-Martnez, Gema Ramrez-Snchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Tobias Kaufmann and Beat Pfister. 2010. Semi-automatic extension of morphological lexica. In *Proceedings of the 2010 International Multiconference on Computer Science and Information Technology*, IMCSIT'10, pages 403–409, Wisla, Poland, October.

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

Nikola Ljubešić and Filip Klubička. 2014. {bs,hr,sr}WaC – web corpora of Bosnian, Croatian and Serbian. In *Proceedings of the 9th Web as Corpus Workshop*, WaC-9, pages 29–35, Gothenburg, Sweden.

Edward M. McCreight. 1976. A space-economical suffix tree construction algorithm. *Journal of the Association for Computing Machinery*, 23(2):262–272, April.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, November.

Benoît Sagot. 2005. Automatic acquisition of a slovak lexicon from a raw corpus. In Václav Matoušek, Pavel Mautner, and Tomáš Pavelka, editors, *Text, Speech and Dialogue*, volume 3658 of *Lecture Notes in Computer Science*, pages 156–163. Springer Berlin Heidelberg.

Jan Šnajder. 2012. Guessing the correct inflectional paradigm of unknown Croatian words. In *Proceedings of the Eighth Language Technologies Conference*, pages 173–178, Ljubljana, Slovenia, October.

Marko Tadić and Antoni Oliver. 2004. Enlarging the Croatian morphological lexicon by automatic lexical acquisition from raw corpora. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, LREC'04, pages 1259–1262, Lisbon, Portugal, May.

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