PoliTa: A multitagger for Polish

Łukasz Kobyliński
Institute of Computer Science
Polish Academy of Sciences
ul. Jana Kazimierza 5, Warsaw, Poland
lkobylinski@ipipan.waw.pl

Abstract

Part-of-Speech (POS) tagging is a crucial task in Natural Language Processing (NLP). POS tags may be assigned to tokens in text manually, by trained linguists, or using algorithmic approaches. Particularly, in the case of annotated text corpora, the quantity of textual data makes it unfeasible to rely on manual tagging and automated methods are used extensively. The quality of such methods is of critical importance, as even 1% tagger error rate results in introducing millions of errors in a corpus consisting of a billion tokens. In case of Polish several POS taggers have been proposed to date, but even the best of the taggers achieves an accuracy of ca. 93%, as measured on the one million subcorpus of the National Corpus of Polish (NCP). As the task of tagging is an example of classification, in this article we introduce a new POS tagger for Polish, which is based on the idea of combining several classifiers to produce higher quality tagging results than using any of the taggers individually.

Keywords: POS tagging, combining taggers, voting ensembles

1. Introduction

The task of Part of Speech (POS) tagging consists in assigning POS tags to tokens (words) in text. It is a fundamental task in Natural Language Processing (NLP) and the accuracy of the morphosyntactic annotation layer produced by POS taggers gains much attention, as most other NLP tasks rely on it. The errors in the POS annotation layer directly influence the accuracy of methods used in such tasks as word sense disambiguation (WSD), sentiment analysis and many others.

For English, the task may be considered nearly solved, as taggers achieve an accuracy of over 97%. In the case of highly inflectional languages, such as Polish, there is still a large margin of tagger-made mistakes, as the authors of even the best taggers report accuracy not higher than 91% (Waszczuk, 2012). This is because the task is much harder for morphologically rich languages, as can be directly seen when comparing the sets of possible POS tags in both English and Polish. For example, in the case of Brown's tagset there are ca. 200 possible tags, while over 4000 in case of the tagset used in the National Corpus of Polish (NCP) (Przepiórkowski et al., 2011). The tagset used in the NCP project consists of tags comprised of many positional attributes, separated by a colon, identifying a particular part of speech (flexeme) and its grammatical categories. For example, the correct tags of the word "candidates" in both languages are:

- Brown's tagset: candidates [candidate:NNS],
- NCP tagset: kandydaci [kandydat:subst:pl:nom:m1].

The problem of producing an accurate morphosyntactic layer of annotation is of a crucial importance in the case of text corpora. Such corpora are either annotated manually by qualified linguists, or automatically, using taggers. For large corpora, such as the National Corpus of Polish, which contains more than 1 billion tokens, relying on manual tagging of the whole corpus is infeasible, because of time and cost constraints. Both manual and automated methods are thus often used, by annotating only a selected, representative part of a corpus by hand and using it as a gold-standard annotation to train automated taggers. Taggers are then used to generate annotations for the remaining part of the corpus.

Several taggers for Polish have been proposed to date, particularly in the last few years new algorithmic approaches to tagging Polish have been presented. The accuracy of the taggers is gradually improving: starting from around 89% in 2010 (Pantera tagger) to ca. 91% in 2012 (Concraft tagger). The results are still less than desired and well below the accuracy achieved for English, so the question arises whether new methods should be further explored, given the fact that the costs of creating a new tagger are usually high and several have already been developed.

To answer that question, we propose to firstly inspect more closely the characteristics of the existing taggers and their mutual relationships. The tagging accuracy statistic provided by the authors of the methods does not tell us how different the taggers in fact are and where does the improvement in performance come from. Are the mistakes made by the taggers similar or completely different? Is the best performing tagger actually better than the others in all possible contexts? Would it be beneficial to choose a different method in a particular context, to maximize the expected tagging accuracy?

In this article we explore these questions and draw from the area of machine learning, where the idea of combining classifiers to achieve higher classification accuracy than using any of the individual methods is very well researched. Tagging is in fact classification: it is the process of assigning one of predefined classes (tags) to each of the given instances (word forms / tokens). We show that there is a possibility of improving Polish tagging accuracy, by utilizing the diversity of currently available taggers and propose an implementation in the form of a tagger based on the results produced by the existing methods: a multi- (poli-) tagger

"PoliTa".

The proposed approach has been devised taking into account the entire process of tagging, starting with a plain text input and ending with morphologically annotated tokens. This allows us to combine individual tagger decisions not only on the level of morphosyntactic disambiguation (the selection of one of possible POS tags for predetermined tokens), but also on the level of segmentation and division into sentences. Consequently, we may evaluate the performance of the proposed method in the most realistic use-case scenario: tagging plain text.

2. Previous work

There have been many attempts to create an ensemble of taggers for English and other languages, for which multiple tagging methods exist. In the case of English, (Brill and Wu, 1998) reported a considerable reduction (6.9%) of the number of errors produced in tagging by using a voted combination of three different taggers. (van Halteren et al., 2001) achieved a much higher, 19.1% reduction of tagging error rate in comparison to the best individual tagger. The authors have adopted a simple voting strategy, pairwise voting and stacking of four different taggers: a trigram tagger, a transformation based learning system, a tagger built around a memory-based learning method and a maximum entropy model.

In the case of Polish, an evaluation of an ensemble of taggers has been presented by (Śniatowski and Piasecki, 2012). The performance of the system has been estimated using a now outdated tagset and a smaller corpus consisting of ca. 880 000 tokens. The authors have also used a method of evaluation, which is now considered to give unfair advantage to some taggers, as it measures the POS tagging disambiguation accuracy and not the accuracy of tagging plain text, which is usually the real world scenario. In another evaluation (Radziszewski and Śniatowski, 2011) have provided results of experiments of combining three taggers using the currently used tagset and a larger corpus, but still employing the approach measuring the disambiguation accuracy, as opposed to the accuracy of tagging plain text.

The results of our own preliminary experiments concerning using voting ensembles to increase the accuracy of POS tagging have been described in (Kobyliński, 2013). In this contribution we describe the architecture of an actual tool developed to use the proposed techniques, elaborate on the subject of similarities and differences between state-of-the-art taggers and propose new strategy of combining component tagger results: per-class voting.

3. Evaluation method

As pointed by (Radziszewski and Acedański, 2012), many previous Polish tagger evaluations considered only the quality of morphosyntactic disambiguation, by reporting the accuracy of the taggers trained on a gold-standard data and tested on text, for which the correct segmentation and morphological analysis was already known. Such an approach artificially increased the accuracy results of the taggers and concealed the problem with imperfect approaches to automatic segmentation and morphological analysis used in real-world scenarios. Here, we have decided to adopt the evaluation framework proposed by the authors of the cited work and performed an end-to-end tagger evaluation, by measuring plain-text tagging accuracy. We also report accuracy results separately for known and unknown words to indicate the differences between the performance of any word-guessing mechanisms used by the taggers.

All the evaluations have been performed on the manually annotated part of the National Corpus of Polish, version 1.1, which consists of 1 215 513 tokens, manually annotated by trained linguists. We will refer to this dataset as NCP1M further on. The process of preparing the gold-standard data for training taggers has been presented in Figure 1. It follows the standard ten-fold cross-validation procedure, with the addition of converting the test folds to plain text (removing any annotation) and by morphologically reanalyzing the training folds. Morphological reanalysis is done by feeding training folds turned into plain text to a morphological analyzer and including the correct interpretation from the gold-standard data, if it has been missed by the analyzer. The aim of this procedure is to train taggers on data most similar to the real-world test data, which is expected to be produced by a morphological analysis tool.

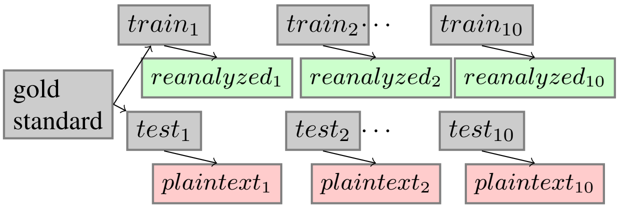

Figure 1: Processing of the gold standard data.

The actual tagger training and testing procedures are depicted in Figure 2. Each of the taggers is trained and evaluated ten times, each time selecting one of ten parts of the corpus for testing and the remaining parts for training. A data model is the result of training and then used in the evaluation procedure as an input data source. The final results are averages calculated over ten training and testing sequences. Each of the taggers and each tagger ensemble has been trained and tested on the same cross-validation folds, so the results are directly comparable.

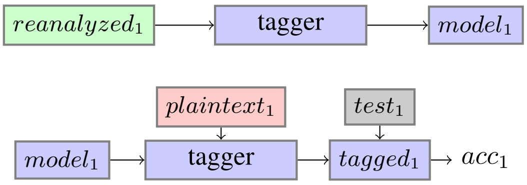

Figure 2: Training tagger model and tagger evaluation.

We use the *accuracy lower bound* (Acc_{lower}) metric to report and compare tagger accuracy. This metric penalizes all segmentation changes in regard to the gold standard and treats such tokens as misclassified. Furthermore, we report separate metric values for both known and unknown words to assess the performance of guesser modules built into the taggers. These are indicated as Acc^{K}_{lower} for known and Acc^{U}_{lower} for unknown words.

4. Tagger performance analysis

To enable a fair comparison of tagging results, we have firstly evaluated each of the individual taggers on the same data, which has been then used for further experiments concerning tagger ensembles.

Pantera (Acedański, 2010) is an adaptation of the Brill's algorithm to morphologically rich languages, such as Polish. Pantera includes several techniques of improving the tagging of inflectional languages, such as multi-pass tagging and transformation templates. In the experiments, we have used the learning threshold value of 6, as recommended by the author.

WMBT (Radziszewski and Śniatowski, 2011) is a memory based tagger, which disambiguates the set of possible tags in multiple tiers. The number of tiers is equal to the number of attributes in the tagset, including the grammatical class. Tokens in each of the individual tiers are classified using a k-Nearest Neighbour classifier. We have used the supplied `nkjp-guess.ini` configuration file for training the tagger.

WCRFT (Radziszewski, 2013) is a tiered tagger, based on Conditional Random Fields (CRF), a mathematical model similar to Hidden Markov Models. A separate CRF model is used to disambiguate distinct grammatical attributes. The `nkjp_s2.ini` configuration has been used during evaluation.

Concraft (Waszczuk, 2012) is another approach to adaptation of CRFs to the problem of POS tagging. In Concraft, the CRF layers are mutually dependent and the results of disambiguation from one of the layers may propagate to another. The first of the two layers used by the tagger includes tags related to POS, case and person, while the second contains all other grammatical categories.

Tagging accuracy of each of the individual taggers has been presented in Table 1.

n	Tagger	Acc_{lower}	Acc^{K}_{lower}	Acc^{U}_{lower}
1	Pantera	88.95%	91.22%	15.19%
2	WMBT	90.33%	91.26%	60.25%
3	WCRFT	90.76%	91.92%	53.18%
4	Concraft	91.07%	92.06%	58.81%

Table 1: The accuracy of individual state-of-the art POS taggers for Polish (evaluated on the NCP1M corpus, tenfold cross-validation).

4.1. Tagger agreement

As a simple means of evaluating the similarities between all of the four taggers and verifying the complexity of the problem, we have compared the outcomes of the evaluated methods and identified the cases in which one, all, or none of the taggers provided the correct tag for a particular test instance. As the results presented in Table 2 show, only 4.18% cases are not handled correctly by any of the taggers. This suggests that there is a great potential in combining several approaches to tagging, as even the best of the individual taggers makes almost 9% mistakes on this data. It is also worth noting that in 90.73% cases all or majority of the taggers are correct, so simply selecting the answer provided by most of the taggers gives comparable results to the individual taggers, but is not enough to beat the best of them.

Tagger outcome	Cases	Examples
All taggers correct	82.78%	
Majority correct	7.95%	3-1, 2-1-1
Correct present, no majority	2.71%	2-2, 1-1-1-1
Minority correct	2.38%	1-3, 1-2-1
All taggers wrong	4.18%	

Table 2: Tagger agreement.

4.2. Tagger complementarity

Following the approach proposed by (Brill and Wu, 1998), we have evaluated the relative differences between the sets of errors made by the individual taggers. The tagger complementarity $Comp(A, B)$ measures how different the mistakes made by the two taggers A and B are. The calculated value is the percentage of time when tagger A is wrong that tagger B is correct:

$$Comp(A, B) = (1 - \frac{e_{AB}}{e_A}) * 100,$$

where e_{AB} is the number of common errors, both in A and B, while e_A is the number of errors made by tagger A. The results for the used taggers, as evaluated on the NCP1M corpus, are presented in Table 3.

A \ B	Pantera	WMBT	WCRFT	Concraft
Pantera	0.00%	42.33%	42.16%	45.22%
WMBT	34.09%	0.00%	35.30%	39.52%
WCRFT	30.78%	32.25%	0.00%	33.97%
Concraft	32.21%	34.52%	31.72%	0.00%

Table 3: Tagger complementarity.

As the results in Table 3 suggest, there is a large overlap in the sets of errors made by the taggers (all values are below 50%, while for completely independent taggers the value would be 100%). There is however still hope of achieving a lower rate of mistakes, especially in the case of Pantera and Concraft taggers combination, for which the complementarity value is the highest.

4.3. Theoretical bounds of combined accuracy

A theoretical upper bound of the expected accuracy of an ensemble of classifiers may be calculated as the number of times all taggers make a mistake, while tagging the test dataset. Even if only one of the classifiers provides the correct answer, there is a possibility of developing a tagger selection method, which is able to accurately distinguish between correct and wrong tagger decisions. The accuracy of such an "Oracle" for each of the possible tagger ensembles has been presented in Table 5 and Figure 3. It is worth noting that such a theoretical upper bound rises to 95.82% in the case of an ensemble constructed on the basis of all the evaluated taggers.

The accuracy of the best individual tagger, which is 91.07% (Concraft), is the natural lower bound, below which creating an ensemble is pointless. With regard to tagger selection strategy, we may think of another lower bound, such as random selection strategy, over which a more elaborate approach should have a significant advantage. Both of these bounds are also presented both in Table 5 and Figure 3.

5. PoliTa: combining component taggers

Here we present the evaluated approaches to combining the results of the individual taggers – the modes of operation of PoliTa tagger. The experiments have been conducted using an ensemble of the four component taggers in several voting scenarios. As has been the case in the experiments described earlier, we have performed an end-to-end tagger evaluation, using plain text as input. The general workflow of the system has been presented in Figure 4. Each of the component taggers has been used in exactly the same setup and using the same parameters, as during the individual evaluation. Taggers have been previously trained on the same set of 10 training folds. Tagging has been performed on test folds, which have been created by turning the original gold-standard data into plain text and performing morphological analysis using the Maca framework (Radziszewski and Śniatowski, 2011) and Morfeusz SGJP analyzer (Woliński, 2006).

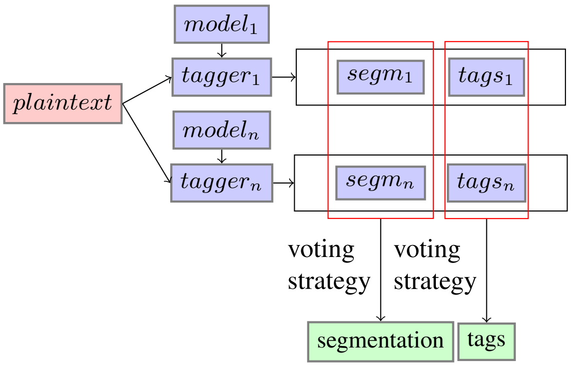

Figure 4: PoliTa architecture.

5.1. Combining segmentation

The consequence of the fact that we are using plain text as input to individual taggers is that there is a possibility of differences in segmentation of the annotated text produced by each of these methods. As such, we have to perform an additional segmentation disambiguation step to deal with these variations and determine the final segmentation, on which the resulting annotation will be based. The differences in segmentation are common, e.g. in the case of tokens containing periods, as shown in Figure 5.

In each case of a segmentation ambiguity, token sequences are lined up using their orthographic word forms. We then use voting to select the most widely represented segmentation variant. In cases in which two different segmentations get the same number of votes, we select one of them randomly. The processing is thus performed token by token and if a difference in segmentation appears, the results of taggers using any of the rejected segmentations are excluded from further processing.

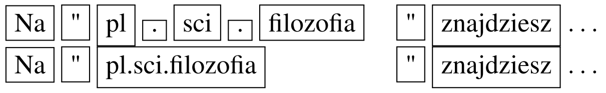

Figure 5: Segmentation differences between taggers (In “pl.sci.filozofia” you will find . . .).

5.2. Combining annotation

We have evaluated the accuracy of each possible configuration of a tagger ensemble, consisting of 2, 3 and 4 taggers. We have evaluated several approaches to combining the decisions of individual taggers, as described below.

Simple voting In the simplest approach, we have used voting to decide which of the proposed segmentations and POS tags is the most probable. Each of the taggers voted for their decision and the POS tag with the greatest number of votes has been selected as the output of the meta-tagger. In the cases in which no majority could be established, a random tagger has been selected as the winning one.

Weighted voting Next, we have evaluated an approach, in which greater influence is given to taggers, which achieve higher accuracy results in individual evaluations. We have thus used the accuracy lower bound metric value, calculated individually for each of the taggers, as the weight of the vote of a particular tagger. In other words, in the case of a tie in the number of votes between alternative annotations, the annotation provided by the group of taggers that perform better (on average) than the other groups will be selected as the winning one.

Per-class voting Finally, we have incorporated the information about the performance of each of the taggers with regard to the grammatical class of the word form under consideration in the voting scheme. In this approach, the votes of the taggers are weighted according to their accuracy in tagging a particular class of tokens. The idea behind this strategy is the assumption that some of the component taggers are better than the others in tagging a particular class of words. The accuracies of tagging tokens of the most common grammatical classes for each of the individual taggers have been presented in Table 4. In cases in which the token has been identified as an unknown word form, the average accuracy of tagging unknown words of that particular tagger has been used as the weighting ratio.

The results of experiments using the PoliTa tagger and the approaches described above are presented in Table 5 and Figure 3.

6. Conclusions and future work

The proposed approach, while relying on the work previously done in the field of Polish POS tagging, allows to achieve higher tagging accuracy than any of the state-of-the-art taggers for Polish. As the experiments conducted on the largest available hand-annotated language resource

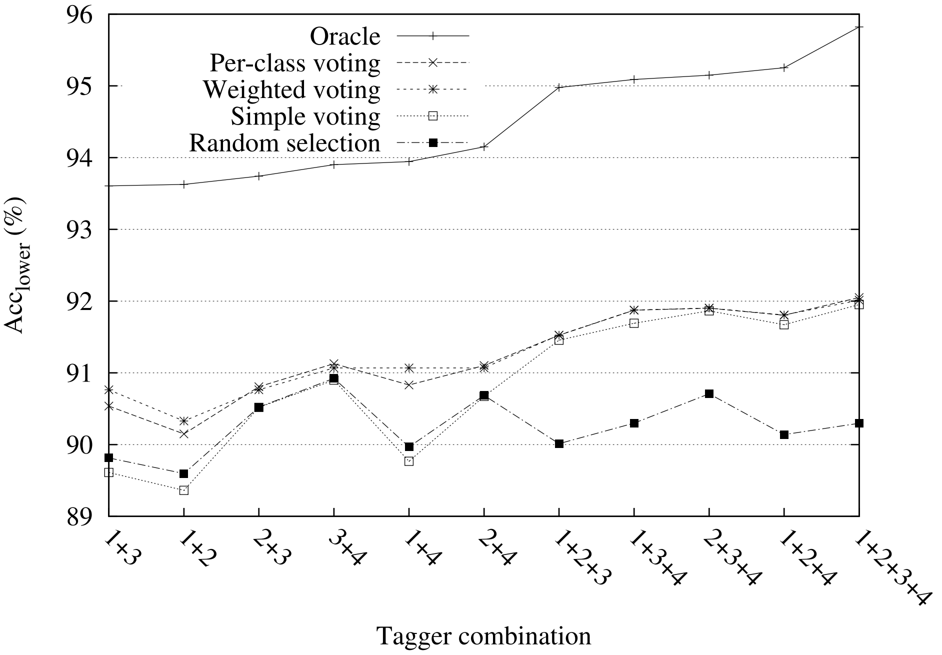

Figure 3: The accuracy of tagger combinations. Oracle: theoretical upper bound. Random selection: a lower bound for evaluating tagger combinations. Taggers are identified by numbers, as given in Table 1.

Strategy	Metric	Taggers				
		1+2+3	1+2+4	1+3+4	2+3+4	1+2+3+4
Random	Acc_{lower}	90.01%	90.14%	90.30%	90.71%	90.30%
Simple Voting	Acc_{lower}	91.46%	91.67%	91.69%	91.86%	91.95%
	Acc^{K}_{lower}	92.57%	92.74%	92.85%	92.78%	92.87%
	Acc^{U}_{lower}	55.19%	56.99%	54.02%	62.07%	62.18%
Weighted Voting	Acc_{lower}	91.53%	91.81%	91.88%	91.90%	92.01%
	Acc^{K}_{lower}	92.60%	92.81%	92.89%	92.82%	92.91%
	Acc^{U}_{lower}	56.55%	59.45%	59.08%	62.31%	62.81%
Per-Class-Voting	Acc_{lower}	91.52%	91.80%	91.87%	91.90%	92.05%
	Acc^{K}_{lower}	92.61%	92.78%	92.89%	92.82%	92.96%
	Acc^{U}_{lower}	56.47%	60.02%	59.03%	62.13%	62.65%
Oracle	Acc_{lower}	94.98%	95.25%	95.09%	95.15%	95.82%

Table 5: The accuracy of tagger combinations. Taggers are identified by numbers, as given in Table 1.

show, the accuracy of PoliTa tagger is one percentage point above the best-performing individual tagger.

PoliTa tagger will be made available as a web-service and included in an already existing framework for publishing language-related resources called Multiservice (Ogrodniczuk and Lenart, 2013). This allows us to avoid the problems concerned with packaging and distributing the tagger with its component taggers, trained models and additional libraries, often working in heterogeneous environments. It also allows us to update it with new component taggers in the future.

7. Acknowledgements

This work has been funded by the National Science Centre project number DEC-2011/01/N/ST6/01107.

8. References

Acedański, S. (2010). A morphosyntactic Brill tagger for inflectional languages. In *Advances in Natural Language Processing*, pages 3–14.

Brill, E. and Wu, J. (1998). Classifier combination for improved lexical disambiguation. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pages 191–195, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kobyliński, Ł. (2013). Improving the accuracy of Polish POS tagging by using voting ensembles. In Vetulani, Z., editor, *Proceedings of the 6th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 453–456, Poznań, Poland. Wydawnictwo Poznańskie, Fundacja Uniwersytetu im. Adama Mickiewicza.

Śniatowski, T. and Piasecki, M. (2012). Combining polish

Class	Count	Acc_{lower} (%) per tagger			
		1	2	3	4
subst	331570	85.21	86.25	87.36	**88.29**
interp	223542	99.63	99.97	99.97	99.97
adj	128703	76.53	81.10	81.56	**82.52**
prep	115818	97.04	97.28	97.54	**98.05**
qub	68079	92.98	**93.82**	92.91	92.92
fin	59458	98.64	98.70	98.81	**98.94**
praet	53326	**90.90**	88.96	89.80	89.69
conj	44840	95.17	**95.41**	94.61	93.96
adv	42750	95.31	**95.59**	95.29	94.77
inf	19213	98.91	**99.20**	99.09	99.14
comp	17842	97.26	**97.29**	96.84	96.88
num	16160	33.40	56.40	**60.32**	55.99

Table 4: Accuracy of individual taggers per grammatical class. Count is the number of tokens, which have been assigned to the particular class in the gold-standard data. Taggers are identified by numbers, as given in Table 1.

morphosyntactic taggers. In Bouvry, P., Kłopotek, M., Leprévost, F., Marciniak, M., Mykowiecka, A., and Rybiński, H., editors, *Security and Intelligent Information Systems*, volume 7053 of *Lecture Notes in Computer Science*, pages 359–369. Springer Berlin Heidelberg.

Ogrodniczuk, M. and Lenart, M. (2013). A multi-purpose online toolset for NLP applications. In Métais, E., Meziane, F., Saraee, M., Sugumaran, V., and Vadera, S., editors, *Proceedings of the 18th International Conference on Applications of Natural Language to Information Systems*, volume 7934 of *Lecture Notes in Computer Science*, pages 392–395. Springer-Verlag, Berlin, Heidelberg.

Przepiórkowski, A., Bańko, M., Górski, R. L., and Lewandowska-Tomaszczyk, B., editors. (2011). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw. Forthcoming.

Radziszewski, A. and Acedański, S. (2012). Taggers gonna tag: an argument against evaluating disambiguation capacities of morphosyntactic taggers. In *Proceedings of TSD 2012*, LNCS. Springer-Verlag.

Radziszewski, A. and Śniatowski, T. (2011). A Memory-Based Tagger for Polish. In *Proceedings of the LTC 2011*. Tagger available at http://nlp.pwr.wroc.pl/redmine/projects/wmbt/wiki/.

Radziszewski, A. and Śniatowski, T. (2011). Maca — a configurable tool to integrate Polish morphological data. In *Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation*.

Radziszewski, A. (2013). A tiered CRF tagger for Polish. In R. Bembenik, Ł. Skonieczny, H. R. M. K. M. N., editor, *Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions*, page to appear. Springer Verlag.

van Halteren, H., Daelemans, W., and Zavrel, J. (2001). Improving accuracy in word class tagging through the combination of machine learning systems. *Comput. Linguist.*, 27(2):199–229, June.

Waszczuk, J. (2012). Harnessing the CRF complexity with domain-specific constraints. The case of morphosyntactic tagging of a highly inflected language. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 2789–2804, Mumbai, India.

Woliński, M. (2006). Morfeusz — a practical tool for the morphological analysis of Polish. In Kłopotek, M. A., Wierzchoń, S. T., and Trojanowski, K., editors, *Intelligent Information Processing and Web Mining*, Advances in Soft Computing, pages 503–512. Springer-Verlag, Berlin.