Definition Extraction with Balanced Random Forests

Łukasz Kobyliński[1] and Adam Przepiórkowski[2,3]

[1] Institute of Computer Science, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
L.Kobylinski@elka.pw.edu.pl
[2] Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa, Poland
adamp@ipipan.waw.pl
[3] Institute of Informatics, University of Warsaw,
ul. Banacha 2, 02-097 Warszawa, Poland

Abstract. We propose a novel machine learning approach to the task of identifying definitions in Polish documents. Specifics of the problem domain and characteristics of the available dataset have been taken into consideration, by carefully choosing and adapting a classification method to highly imbalanced and noisy data. We evaluate the performance of a Random Forest-based classifier in extracting definitional sentences from natural language text and give a comparison with previous work.

1 Introduction

Natural Language Processing (NLP) tasks often involve heavily imbalanced data, with a dominating "uninteresting" class and a minority "interesting" class. One such task is that of definition extraction, where a set of sentences is to be classified into definitional and non-definitional sentences. There may be as many as 20 non-definition sentences for any single definition sentence in an instructive text, but it is the latter class that a definition extraction system is interested in.

The usual Machine Learning (ML) classifiers, ranging from naïve bayesian methods, through decision trees, perceptrons and various lazy learners, to the currently very popular classifiers based on Support Vector Machines (SVMs) and on Adaboost, do not work well in such cases, even when trained with subsampling (of uninteresting examples) or oversampling (of the interesting examples). The problem is that such classifiers attempt to minimise the overall error rate, rather than concentrating on the interesting class. In case of a dataset with a 1:20 ratio of interesting to uninteresting cases, it is difficult to beat a classifier uniformly assigning each new item to the uninteresting class: such a classifier reaches the overall accuracy higher than 95%, but at the cost of misclassifying all the interesting cases!

The problem of heavily imbalanced data has already been addressed in the ML community, where most solutions consist either in the assignment of a high

A. Ranta (Ed.): GoTAL 2008, LNAI 5221, pp. 237–247, 2008.

cost to the misclassification of the minority class or in subsampling and/or oversampling. A novel approach to the problem has been proposed in Chen *et al.* 2004 and it consists in a modification of the Random Forest (Breiman, 2001) classifier.

Random Forest (RF) is a homogeneous ensemble of unpruned decision trees (e.g., CART, C4.5), where — at each node of the tree — a subset of all attributes is randomly selected and the best attribute on which to further grow the tree is taken from that random set. Additionally, Random Forest is an example of the bagging (bootstrap aggregating) method, i.e., each tree is trained on a set bootstrapped[1] from the original training set. Decisions are reached by simple voting.

Balanced Random Forest (BRF; Chen *et al.* 2004) is a modification of RF, where for each tree two bootstrapped sets of the same size, equal to the size of the minority class, are constructed: one for the minority class, the other for the majority class. Jointly, these two sets constitute the training set.

The aim of this paper is to demonstrate that BRF is a technique well-suited to the difficult problem of definition extraction and, by extension, other NLP tasks. When trained on the dataset of Polish instructive texts introduced in Przepiórkowski *et al.* (2007b,a), BRF-based classifiers give better results than manual definition extraction grammars (Przepiórkowski *et al.*, 2007a), the usual ML classifiers, even when sequentially combined with some *a priori* linguistic knowledge (Degórski *et al.*, 2008), or a linear combination of such ML classifiers and complete manual grammars (Przepiórkowski *et al.*, 2008).

In what follows we first introduce the attribute space assumed here (§2), then describe the used classification approach (§3) and present the results of our experiments (§4). Finally, we outline work conducted previously in the field (§5) and conclude with possibilities of further research (§6).

2 Feature Selection

Employing any machine learning approach to unstructured data requires that data is represented in the form of feature values, either binary, numeric or nominal. We use a relatively straightforward approach of n-gram representation of the sentences in the available document set. Each sentence is represented by a vector of binary values, where each value indicates whether a particular n-gram is present in the corresponding sentence. The n-grams consist of base forms of words, their parts of speech and grammatical cases that appear in the greatest number of sentences in all documents. We individually count the occurrences of each of the n-grams in sentences marked as definitions and non-definitions. Both lists are then combined and a number of most common entries is selected to form a dictionary of features used for sentence description.

[1] That is, examples in such a bootstrapped training set are uniformly and randomly drawn *with replacement* from the original training set. As a result, some examples will be repeated while other will not make it to the bootstrapped set.

Even by limiting the length of generated n-grams to $n \leq 3$ and having a choice of three distinct n-gram types: base word form (further denoted as *base*), part of speech of the word (*ctag*) and its grammatical case (*case*), we face a problem of many possible dictionary configurations, selecting from the set of $3^1 + 3^2 + 3^3 = 39$ possibilities. Including too many n-gram types would result in a an extremely large attribute space, while including too few in reducing the potential classification accuracy. We approached the problem by measuring the average value of the χ^2 statistic of each of the possible n-gram types with respect to the class attribute. This was performed on a training set consisting of all the available documents, on the basis of 100 n-grams for each of the 39 types. Table 1 presents a list of the 20 n-gram types with the highest average χ^2 value.

Table 1. Top 20 values of the χ^2 statistic of possible n-gram permutations

rank	n-gram	average χ^2	rank	n-gram	average χ^2
1	*base*	21.04	11	*base base ctag*	16.70
2	*ctag ctag case*	18.91	12	*ctag base ctag*	16.29
3	*ctag base*	18.53	13	*ctag ctag base*	14.77
4	*base case*	18.45	14	*ctag case*	14.69
5	*base ctag*	17.92	15	*ctag ctag ctag*	14.63
6	*base base*	17.81	16	*base ctag case*	14.52
7	*base base case*	17.73	17	*base base base*	14.33
8	*ctag base case*	17.43	18	*ctag*	13.88
9	*ctag ctag*	17.11	19	*ctag case ctag*	13.65
10	*ctag base base*	16.73	20	*base ctag ctag*	13.59

Unfortunately, just taking a number of attributes from the top of this list does not guarantee the best possible selection of n-gram types. This is because certain attribute pairs may be statistically dependent and introducing both of them into the dictionary would result in noise, instead of meaningful data for the classifier. Having experimented with different attribute configurations, we have chosen the following heuristic procedure of attribute selection: we take one attribute at a time from the sorted list, starting from the top, and reject these n-grams of length $n = 3$, for which another trigram with one of the same feature types has already been selected. The resulting set of 10 selected n-gram types is presented in Table 2.

Table 2. The selected set of n-gram types

no.	n-gram	no.	n-gram
1	*base*	6	*base base*
2	*ctag ctag case*	7	*ctag ctag*
3	*ctag base*	8	*ctag case*
4	*base case*	9	*base base base*
5	*base ctag*	10	*ctag*

For comparison purposes, we also present here the results of experiments on a dataset from the work of Degórski *et al.* 2008, where the set of n-grams has been selected in a different manner and using a different set of attributes. Specifically, this dataset, referenced later as the "baseline dataset", has been created by using the 100 most common uniform unigrams, bigrams and trigrams of base forms, parts of speech and cases (i.e., *base*, *base-base*, ..., *ctag-ctag-ctag*).

3 Classifying Imbalanced Data

As noted earlier, the available dataset of definitional and non-definitional sentences is highly imbalanced and consists of 10830 sentences, 546 of which contain — or are a part of — definitions. Consequently, any successful classification-based approach to extraction of definitions from this data must take into consideration — either explicitly or implicitly — the difference in training samples from both categories.

The most common way of dealing with imbalanced data is introducing appropriately weighted costs for specific classes or sampling the available training set. Balanced Random Forest is an approach where equalizing the influences of classes is not performed externally to classification algorithm by evaluating weights, but is integrated in the very process. Here, for the task of extracting definitions from a set of documents by sentence classification, we use the following algorithm, based on Chen *et al.* 2004:

- split the training corpus into definitions and non-definitions; let us assume that there are n_d definitions and n_{nd} non-definitions, where $n_d < n_{nd}$;
- construct k trees, each in the following way:
  - draw a bootstrap sample of size n_d of definitions, and a bootstrap sample of the same size n_d of non-definitions;
  - learn the tree (without pruning) using the CART algorithm, on the basis of the sum of the two bootstrap samples as the training corpus, but:
  - at each node, first select at random m features (variables) from the set of all M features ($m < M$; selection without replacement), and only then select the best feature (out of these m features) for this node; this random selection of m features is repeated for each node;
- the final classifier is the ensemble of the k trees and decisions are reached by simple voting.

We have chosen the value of m to be equal to $\sqrt{M}$ in all the experiments.

As Random Forest is a well known classifier and widely covered in the literature, it also allows having a greater insight into the results produced by the BRF approach. RFs have been verified to be suitable both for large and highly dimensional data, as is the case in natural language processing. They also provide means of estimating the classification error rate without performing a full cross-validation procedure and for estimating variable importance and variable interactions. In our current experiments we have not performed such estimations, as we are more interested in selecting the optimal set of n-gram types, than comparing the importance of particular features.

4 Experimental Results

We use several statistical parameters to describe and compare the results of the proposed classification approach: recall and precision are the most commonly calculated information retrieval performance measures. We assume the sentences marked as definitions to be the set of relevant documents in the retrieval task:

$$precision = \frac{|\{\text{definitions}\} \cap \{\text{retrieved sentences}\}|}{|\{\text{retrieved sentences}\}|} \tag{1}$$

$$recall = \frac{|\{\text{definitions}\} \cap \{\text{retrieved sentences}\}|}{|\{\text{definitions}\}|} \tag{2}$$

For a single-valued performance indicator, we use the F-measure, both in the form used in the previous papers on Polish definition extraction (marked as F_α) and in the more common sense (marked as F_β). For F_1 we just use F_1 (as $F_{\alpha=1} = F_{\beta=1}$):

$$F_\alpha = \frac{(1+\alpha) \cdot precision \cdot recall}{\alpha \cdot precision + recall} \tag{3}$$

$$F_\beta = \frac{(1+\beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{4}$$

Finally, we also calculate the area under the ROC curve (AUC), which is another single-valued measure of retrieval accuracy, but not tied to a single probability threshold value, like the F-measure. Still, because in the task of definition extraction we are more interested in maximizing the recall value (in other words: minimizing the false negative rate), we compare all further experiment results on the basis of $F_{\alpha=2}$ and $F_{\beta=2}$ values.

Our initial experiments aimed at verifying whether any additional preprocessing of the available data, commonly applied to text classification problems, would result in improving the accuracy of definition extraction. Firstly, we have included the information about the relative position of an n-gram in a sentence into the feature vector. By dividing the sentences into three equal parts and counting the n-gram occurrences in each of the parts separately, we have increased the attribute space three times, but achieved no increase in performance (Table 3). We may speculate that the positional information introduced too much noise, as the available dataset was too small to benefit from the significantly larger feature space.

Similarly, there was no gain in definition extraction accuracy after including the information about the actual number of occurrence counts of particular n-grams in the analyzed sentences. This may also be explained by a relatively small size of the available dataset and sparseness of the feature vector. The calculated numbers of occurrences were negligibly small and provided no additional information to the classifier.

Finally, applying a stop-list of most common words and filtering non-alphanumeric characters from the documents also proved to reduce both the value of $F_{\alpha=2}$ and $F_{\beta=2}$ measures. Thus, neither of the attribute modifications and data

Table 3. The influence of additional preprocessing steps on classification accuracy. Ten-fold cross-validation results, with 100 iterations of random trees generation.

dataset	precision	recall	F_1	$F_{\alpha=2}$	$F_{\alpha=5}$	$F_{\beta=2}$	$F_{\beta=5}$	AUC
base	18.1%	66.1%	28.4%	**35.1%**	45.9%	**43.2%**	60.0%	82.4%
n-gram position	16.2%	63.9%	25.9%	32.3%	42.9%	40.2%	57.4%	81.2%
n-gram occurrence	17.2%	65.0%	27.2%	33.7%	44.4%	41.8%	58.7%	81.4%
base form stoplist	17.3%	63.0%	27.2%	33.5%	43.7%	41.2%	57.2%	81.4%

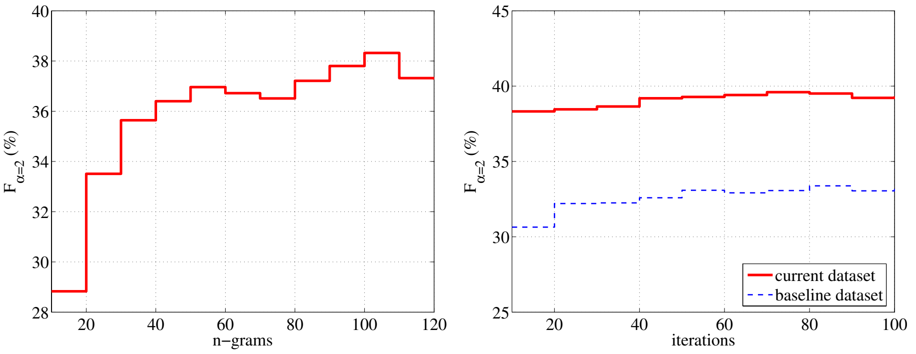
preprocessing steps mentioned above have been used in further experiments. A detailed comparison of each of the approaches has been presented in Figure 2a.

In an effort to determine the optimal size of feature space for classification, we have conducted a series of experiments with an increasing number of n-grams used for sentence representation (Table 4 and Figure 1a). On the basis of the results, we have decided to use 100 n-grams of each type in further experiments, as increasing their number above that threshold does not seem to have any positive influence on the classification accuracy. By choosing that number, we obtained a training set consisting of 10830 instances and 929 attributes (as there are less than 100 different n-grams of the type *ctag*).

Table 4. The influence of the number of used n-grams of each type on classification accuracy. Ten-fold cross-validation results, with 100 iterations of random trees generation.

n-grams	precision	recall	F_1	$F_{\alpha=2}$	$F_{\alpha=5}$	$F_{\beta=2}$	$F_{\beta=5}$	AUC
10	14.4%	57.7%	23.1%	28.8%	38.5%	36.0%	51.7%	76.6%
20	17.2%	63.7%	27.1%	33.5%	43.9%	41.4%	57.7%	81.7%
30	18.7%	65.4%	29.0%	35.6%	46.1%	43.6%	59.6%	82.7%
40	19.1%	66.8%	29.7%	36.4%	47.1%	44.5%	61.0%	82.6%
50	19.3%	67.9%	30.1%	37.0%	47.9%	45.2%	62.0%	83.1%
60	19.3%	67.2%	29.9%	36.7%	47.5%	44.9%	61.3%	82.9%
70	19.1%	66.8%	29.8%	36.5%	47.2%	44.6%	61.0%	83.2%
80	19.7%	67.2%	30.4%	37.2%	47.9%	45.3%	61.5%	83.7%
90	19.8%	69.4%	30.8%	37.8%	49.0%	46.2%	63.3%	84.5%
100	20.1%	70.1%	31.2%	**38.3%**	49.6%	**46.8%**	64.0%	83.8%
110	19.6%	68.1%	30.4%	37.3%	48.2%	45.6%	62.2%	84.1%
120	19.6%	67.8%	30.4%	37.3%	48.1%	45.5%	61.9%	84.1%

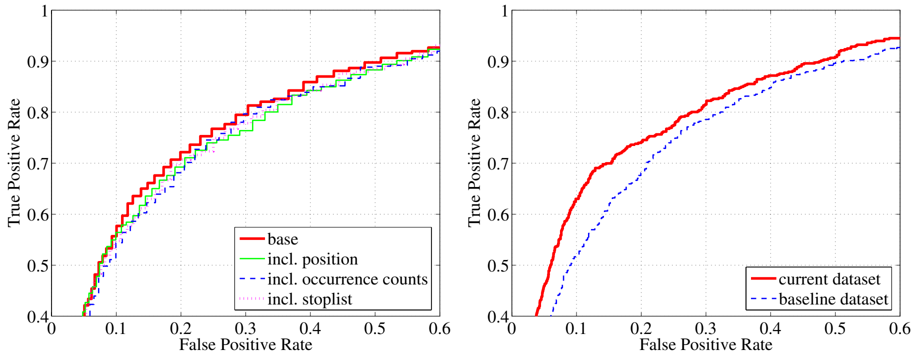
As the accuracy of Random Forest classification depends heavily on the number of generated random trees used in voting, we have conducted the experiments both on the current dataset and on the baseline dataset provided by Degórski *et al.* 2008 for several different numbers of iterations (Tables 5 and 6, Figure 1b). We have performed ten-fold cross-validation experiments instead of counting the out-of-bag error of the bagging classifier, so as to make the results



(a) **(b)**

Fig. 1. 1a Performance of classification with respect to the number of used n-grams, 1ba comparison between classification performance using the baseline dataset and the current dataset for different number of iterations

as closely comparable with those of Degórski *et al.* 2008 as possible. The detailed comparison of both sets, with respect to BRF classification accuracy for the number of iterations which proved to give the best results for each of the sets, is presented in Figure 2b.



(a) **(b)**

Fig. 2. ROC curve of classification: 2a using additional data preprocessing steps, 2b using the baseline dataset and the current dataset

As may be seen from the results of the consecutive experiments, increasing the number of generated random trees improves the accuracy of definitional sentences classification only up to a certain point. Above that threshold the performance reaches a plateau and no further iterations are necessary.

While the use of Balanced Random Forest classification method alone significantly improves the definition extraction performance over other pure machine

Table 5. Ten-fold cross-validation results of the baseline dataset classification

iterations	precision	recall	F_1	$F_{\alpha=2}$	$F_{\alpha=5}$	$F_{\beta=2}$	$F_{\beta=5}$	AUC
100	15.1%	63.0%	24.4%	30.6%	41.2%	38.6%	56.2%	80.6%
200	16.1%	64.3%	25.8%	32.2%	42.9%	40.3%	57.7%	81.3%
300	16.2%	63.8%	25.9%	32.3%	42.8%	40.2%	57.3%	81.6%
400	16.5%	63.6%	26.2%	32.6%	43.1%	40.5%	57.3%	81.6%
500	16.8%	64.1%	26.6%	33.1%	43.7%	41.0%	57.9%	81.7%
600	16.7%	63.9%	26.5%	32.9%	43.5%	40.9%	57.7%	81.7%
700	16.9%	63.6%	26.7%	33.1%	43.5%	40.9%	57.5%	81.8%
800	17.0%	64.1%	26.9%	**33.4%**	43.9%	**41.3%**	58.0%	81.9%
900	16.9%	63.6%	26.7%	33.1%	43.5%	40.9%	57.5%	81.9%
1000	16.9%	64.0%	26.8%	33.2%	43.7%	41.1%	57.8%	81.9%

Table 6. Ten-fold cross-validation results of the current dataset classification

iterations	precision	recall	F_1	$F_{\alpha=2}$	$F_{\alpha=5}$	$F_{\beta=2}$	$F_{\beta=5}$	AUC
100	20.1%	70.1%	31.2%	38.3%	49.6%	46.8%	64.0%	83.8%
200	20.5%	68.7%	31.5%	38.5%	49.3%	46.7%	63.0%	84.4%
300	20.6%	68.7%	31.7%	38.7%	49.5%	46.8%	63.0%	84.5%
400	21.0%	69.2%	32.2%	39.2%	50.0%	47.4%	63.6%	84.6%
500	21.1%	68.9%	32.3%	39.3%	50.0%	47.4%	63.4%	84.7%
600	21.2%	68.9%	32.5%	39.4%	50.1%	47.5%	63.4%	84.7%
700	21.4%	69.0%	32.6%	**39.6%**	50.3%	**47.7%**	63.6%	84.7%
800	21.3%	69.0%	32.5%	39.5%	50.3%	47.7%	63.6%	84.8%
900	21.1%	68.7%	32.3%	39.2%	49.9%	47.3%	63.2%	84.8%
1000	21.2%	68.7%	32.4%	39.3%	50.0%	47.4%	63.2%	84.8%

learning based approaches (e.g., as reported by Degórski *et al.* 2008), it is worth pointing out that a careful feature selection is an equally important step. We achieve an over 18% increase in accuracy, as indicated by the $F_{\alpha=2}$ measure, by describing the sentences with a more representative set of attribute types.

5 Previous Work

To the best of our (and Google's) knowledge, there is no previous NLP work taking advantage of the Balanced variety of RFs. Apparently, the first NLP applications of the plain Random Forests are those reported in Nielsen and Pradhan 2004, for PropBank-style (Kingsbury and Palmer, 2002) role classification, and in Xu and Jelinek 2004 (followed by a series of papers by the same authors, culminating in Xu and Jelinek 2007), where they are used in the classical language modelling task (predicting a sequence of words) for speech recognition and give better results than the usual n-gram based approaches.

On the other hand, there is some substantial previous work on definition extraction, as this is a subtask of many applications, including terminology extraction (Pearson, 1996), the automatic creation of glossaries (Klavans and Muresan, 2000, 2001), question answering (Miliaraki and Androutsopoulos, 2004; Fahmi and Bouma, 2006), learning lexical semantic relations (Malaisé *et al.*, 2004; Storrer and Wellinghoff, 2006) and the automatic construction of ontologies (Walter and Pinkal, 2006). Despite the current dominance of the ML paradigm in NLP, tools for definition extraction are invariably language-specific and involve shallow or deep processing, with most work done for English (Pearson, 1996; Klavans and Muresan, 2000, 2001) and other Germanic languages (Fahmi and Bouma, 2006; Storrer and Wellinghoff, 2006; Walter and Pinkal, 2006), as well as French (Malaisé *et al.*, 2004).

When ML methods are used, it is in combination with linguistic processing. For example, Fahmi and Bouma 2006 applied a robust wide-coverage parser of Dutch to select candidate definition sentences, which were then subject to an ML classifier. They experimented with three classifiers (Naïve Bayes, SVM and Maximum Entropy) and a number of possible feature configurations and obtained the best results for the Maximum Entropy classifier and feature configurations, which included some syntactic features.

For Polish, first attempts at constructing definition extraction systems are described — in the context of other Slavic languages — in Przepiórkowski *et al.* 2007b, and improved results are presented in Przepiórkowski *et al.* 2007a. In that work definitions were identified on the basis of a manually constructed partial grammar (a cascade of regular grammars over morphosyntactically annotated XML-encoded texts), with the best grammar giving the precision of 18.7% and recall of 59.3%, which amounts to $F_{\alpha=2} = 34.4\%$. Przepiórkowski *et al.* 2007a note that these relatively low results are at least partially due to the inherent difficulty of the task: the inter-annotator agreement measured as Cohen's κ is only 0.31 (the value of 1 would indicate perfect agreement, the value of 0 — complete randomness). The same dataset was used in the experiments reported here.

An approach more directly comparable to ours is presented in Degórski *et al.* 2008. The general idea is analogous to that of Fahmi and Bouma 2006: first candidate definition sentences are selected via linguistic methods and then they are classified using ML methods. What is novel in Degórski *et al.* 2008 is the very basic character of the linguistic knowledge (a small low-precision collection of n-grams typical for definitions, including the copula, sequences corresponding to *that is* and *i.e.*, etc.), and the use of ensembles of classifiers in the second stage. The best results reported there, the precision of 19.9%, recall of 69.2%, and $F_{\alpha=2} = 38.0\%$, are significantly better than those of Przepiórkowski *et al.* 2007a, but still, despite some use of *a priori* language-specific knowledge, worse than the pure ML results reported here.

6 Conclusions and Future Work

Our currently reported results seem to restore hope in the machine learning approach to the vaguely specified task of definition extraction from a small set of text documents. It is usually the case that the smaller, less structured and more noisy the available training data, the lesser is the advantage of such methods over hand-crafted rules and grammars, utilizing linguistic knowledge. Thus, achieving better results in such circumstances by a pure machine learning approach seems to justify the necessary work on feature and classification method selection.

It would still be interesting to combine the current classification method with manually constructed grammars, similarly as in Degórski *et al.* 2008, to see if such a sequential processing scheme would further improve the definition extraction performance. On the basis of the experiments described there, we might expect a considerable increase in retrieval precision, at the cost of a slight decrease in recall.

Acknowledgements

We are very grateful to Prof. Jacek Koronacki for suggesting the use of Balanced Random Forests in the task of definition extraction and his further helpful comments. We also thank the anonymous reviewers for providing valuable remarks.

References

Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)

Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Technical Report 666, University of California, Berkeley (2004), `http://www.stat.berkeley.edu/tech-reports/666.pdf`

Degórski, Ł, Marcińczuk, M., Przepiórkowski, A.: Definition extraction using a sequential combination of baseline grammars and machine learning classifiers. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008). ELRA, Forthcoming (2008)

Fahmi, I., Bouma, G.: Learning to identify definitions using syntactic features. In: Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications (2006)

Kingsbury, P., Palmer, M.: From TreeBank to PropBank. In: Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, pp. 1989–1993. ELRA, Las Palmas (2002)

Klavans, J.L., Muresan, S.: DEFINDER: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text. In: Proceedings of the Annual Fall Symposium of the American Medical Informatics Association (2000)

Klavans, J.L., Muresan, S.: Evaluation of the DEFINDER system for fully automatic glossary construction. In: Proceedings of AMIA Symposium (2001)

Lin, D., Wu, D. (eds.): Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004). ACL, Barcelona (2004)

Malaisé, V., Zweigenbaum, P., Bachimont, B.: Detecting semantic relations between terms in definitions. In: Ananadiou, S., Zweigenbaum, P. (eds.) COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology, Geneva, Switzerland, pp. 55–62 (2004)

Miliaraki, S., Androutsopoulos, I.: Learning to identify single-snippet answers to definition questions. In: Proceedings of COLING 2004, Geneva, Switzerland, pp. 1360–1366 (2004)

Nielsen, R.D., Pradhan, S.: In: Lin,, Wu (eds.) Mixing weak learners in semantic parsing, pp. 80–87 (2004)

Pearson, J.: The expression of definitions in specialised texts: a corpus-based analysis. In: Gellerstam, M., Järborg, J., Malmgren, S.G., Norén, K., Rogström, L., Papmehl, C. (eds.) Proceedings of the Seventh Euralex International Congress, Göteborg, pp. 817–824 (1996)

Przepiórkowski, A., Degórski, Ł, Wójtowicz, B.: On the evaluation of Polish definition extraction grammars. In: Vetulani, Z. (ed.) Proceedings of the 3rd Language & Technology Conference, Poznań, Poland, pp. 473–477 (2007a)

Przepiórkowski, A., Degórski, Ł, Spousta, M., Simov, K., Osenova, P., Lemnitzer, L., Kuboň, V., Wójtowicz, B.: Towards the automatic extraction of definitions in Slavic. In: Piskorski, J., Pouliquen, B., Steinberger, R., Tanev, H. (eds.) Proceedings of the Workshop on Balto-Slavonic Natural Language Processing at ACL 2007, Prague, pp. 43–50 (2007b)

Przepiórkowski, A., Marcińczuk, M., Degórski, Ł.: Dealing with small, noisy and imbalanced data: Machine learning or manual grammars? In: Sojka, P., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue: 9th International Conference (TSD 2008), Brno, Czech Republic, September 2008. LNCS (LNAI). Springer, Berlin (2008)

Storrer, A., Wellinghoff, S.: Automated detection and annotation of term definitions in German text corpora. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, ELRA, Genoa (2006)

Walter, S., Pinkal, M.: Automatic extraction of definitions from German court decisions. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, pp. 20–28 (2006)

Xu, P., Jelinek, F.: In: Lin,, Wu (eds.) Random forests in language modeling, pp. 325–332 (2004)

Xu, P., Jelinek, F.: Random forests and the data sparseness problem in language modeling. Computer Speech and Language 21(1), 105–152 (2007)