Proceedings of the International Multiconference on Computer Science and Information Technology pp. 353–357
ISBN 978-83-60810-14-9
ISSN 1896-7094

Definition Extraction: Improving Balanced Random Forests

Łukasz Degórski
Institute of Computer Science
Polish Academy of Sciences
ul. J. K. Ordona 21
01-237 Warszawa, Poland
Email: ldegorski@bach.ipipan.waw.pl

Łukasz Kobyliński
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warszawa, Poland
Email: L.Kobylinski@elka.pw.edu.pl

Adam Przepiórkowski
Institute of Computer Science
Polish Academy of Sciences
ul. J. K. Ordona 21
01-237 Warszawa, Poland
Email: adamp@ipipan.waw.pl

***Abstract*—The article discusses methods of improving the ways of applying Balanced Random Forests (BRFs), a machine learning classification algorithm, used to extract definitions from written texts. These methods include different approaches to selecting attributes, optimising the classifier prediction threshold for the task of definition extraction and initial filtering by a very simple grammar.**

I. Introduction

THE paper deals with extracting definitions from relatively unstructured instructive texts (textbooks, learning materials in eLearning, etc.) in a morphologically rich, relatively free word order, determinerless language (Polish). The same methods could easily be used for other similar languages without or with only minor changes, though. The work reported here is a continuation of work carried out within the recently finished *Language Technology for eLearning* project (LT4eL; http://www.lt4el.eu/).

The aim of the paper is to show how the results of previous attempts, presented in [1], can be improved by choosing the optimal threshold of classifier's prediction, with respect to the task of definition extraction, as well as to show that these improved results are close to optimal, in the sense that preliminary filtering by a simple grammar does not improve them significantly, as it was the case in experiments described in [2]. Attempts to use a different set of attributes will also be mentioned.

We used the same corpus of instructive texts as in [1] and [2]. It was automatically annotated morphosyntactically and then manually annotated for definitions, and contains over 30000 tokens, almost 11000 sentences and 558 definitions. These were divided by annotators into 6 types, depending on the most recognisable marker of being a definition:

- copula verb (e.g. cat **is a** domestic animal...)
- other verbs (e.g. we **define** a cat as a domestic animal...)
- punctuation (e.g. cat**:** a domestic animal...)
- layout (e.g. defined phrase in bold, large font, the definition in smaller font in the next line)
- pronoun
- other

We performed the experiments on two corpora: the whole set (described above) and its copula-type subset (the same sentences, but only 173 definitions). The experiments for other languages, conducted by other members of the LT4eL project, have shown that copula definitions have the highest probability of being successfully extracted by means of machine learning methods.

Note that the number of definitions in both sets is not exactly equal to the number of what we later call *definitional sentences*. Manually annotated definitions may begin or end in the middle of a sentence, and span multiple sentences. However, the ML methods operate on sentence level: a definitional sentence in this context is a sentence that has a nonempty intersection with at least one definition. For instance, in the whole set there are 546 definitional sentences.

The rest of the paper is organized as follows. In Section II we describe the classification method used for definition extraction. In Section III we discuss the possibilities of choosing representative attributes of words for the task of definition extraction. In Section IV we present differences in the achieved results, with respect to chosen methodology of interpreting classifier's outcome. In Section V we present the influence of manually constructed grammars on the accuracy of our definition extraction approach. Finally, we present the previous work done in the field in Section VI and conclude in Section VII.

II. BRF algorithm

Random Forest (RF; [3]) is a homogeneous ensemble of unpruned decision trees (e.g., CART, C4.5; [4]), 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 from the original training set. Decisions are reached by simple voting.

Balanced Random Forest (BRF; [5]) 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.

Similarly as in [1], for the task of extracting definitions from a set of documents by sentence classification, we use the following version of the BRF algorithm:

- split the training corpus into n_d definitions and n_{nd} non-definitions; the input data is heavily imbalanced, so $n_d \ll n_{nd}$;
- 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 ($m \ll M$; 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, although other sufficiently small values of m could be used, as discussed in [3].

Up to $k = 800$ random trees were generated in each experiment. We always quote the results for the best-performing number of iterations in a given configuration (corpus, attributes, optimisation and filtering). The best-performing number varied between 300 and 700 for different configurations.

III. Choosing the attributes

In [1], a set of 10 permutations of n-gram types was used for document representation as machine learning attributes (Table I). The set was carefully chosen by a half-statistical, half-heuristic method (having in mind the χ^2 statistic value with respect to the class attribute and statistical independence of the attributes). In these experiments 100 most common n-grams of each of the 10 types were used for document representation, resulting in a dataset of ca. 900 binary attributes (fewer than 100 values for *ctag* unigrams exist) and 10830 instances. Data instances correspond to document sentences, while the values of binary attributes indicate whether a particular n-gram appears in the sentence.

TABLE I
THE PREVIOUSLY USED 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*		

In our current experiments we have tried a slightly different method. For each of the possible 39 permutations of 1-grams, 2-grams and 3-grams of available features: *base* (base word form), *case* (grammatical case) and *ctag* (part of speech of the word), we generate up to 100 most frequent n-grams. As not for all permutations 100 different n-grams exist, the final set has around 3750 attributes.

In each iteration of 10-fold cross-validation we proceed as follows:

- in the training set (90% of the corpus):
  1) order the attributes according to the value of the χ^2 statistic with respect to the class attribute,
  2) select the top 900 attributes (those fitting the example class best),
  3) train the Balanced Forest classifier on the set;
- in the test set (10% of the corpus):
  4) reject all attributes not on the top 900 list,
  5) apply the classifier.

The number of attributes was not chosen arbitrarily. Previous experiments (cf. Table 4 in [1]) have shown that increasing the number of n-grams of each of the selected types over 100 does not improve the classification results. That is the reason why in that method (with 10 n-gram types, and not all types had 100 n-grams) about 900 attributes were used. For comparability, in the new method we used a similar number of attributes – chosen differently though.

The experiments were performed on the whole set of definitions (as in [1]), and also on a version of the corpus in which only the copula definitions were marked. We have used the two known versions of the F measures to assess the results:

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

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

The new method gave promising results for the copula definitions:

TABLE II
COMPARISON OF ATTRIBUTE SELECTION METHODS, COPULA DEFINITIONS

attributes	precision	recall	$F_{\alpha=1}$	$F_{\alpha=2}$	$F_{\beta=2}$	$F_{\alpha=5}$
preselected	16.50	**84.40**	27.60	35.59	46.30	50.06
χ^2	**17.60**	81.70	**28.96**	**36.90**	**47.27**	**50.84**

Unfortunately it turned out to be disappointing when applied to all definitions:

TABLE III
COMPARISON OF ATTRIBUTE SELECTION METHODS, ALL DEFINITIONS

attributes	precision	recall	$F_{\alpha=1}$	$F_{\alpha=2}$	$F_{\beta=2}$	$F_{\alpha=5}$
preselected	**21.37**	**69.04**	**32.64**	**39.60**	**47.74**	**50.33**
χ^2	20.60	65.20	31.31	37.87	45.50	47.91

This leads to a conclusion that the more general method of choosing n-gram types for the task of definition extraction may still perform better than direct selection of specific n-grams in each classification iteration. The advantage of performing a purely statistical attribute selection lies in eliminating any preconceived notions about the role of certain word n-gram types in discriminating definitional sentences from non-definitional. On the other hand, a preselected set of n-gram types may be used without any further data analysis for

document representation in other, similar problems, maybe even different languages.

IV. Optimising the threshold

The task of extracting definitions from an annotated corpus of documents was defined by the LT4eL project mentioned above, which focused on facilitating the construction and retrieval of learning objects (instructive material) in eLearning with the help of language technology. The results of automatic definition extraction were to be presented to the author or the maintainer of a learning object as candidates for the glossary of this learning object.

The intended use determines the appropriate approach. It is obviously easier to reject wrong definition candidates than to go back to the text and search for missed good definitions manually, so in this application recall was more important than precision. In [1] this assumption was exploited at the evaluation level only. $F_{\alpha=2}$ and $F_{\beta=2}$ were taken into account when comparing the approaches and datasets, to acknowledge the preference for recall. The classifier's prediction threshold of being a definition was set arbitrarily to 0.5 there. As Balanced Forest algorithm takes care of weighting the imbalanced number of examples of both classes (definitions and non-definitions), this approach does not favour any class, so the ratio of correctly classified examples to all examples was maximised.

However, it is worth noting that this is not exactly what we need here. Favouring recall over precision, we would like to focus more on the correctly classified positive examples, at the inevitable cost of misclassifying some of the negative ones. On the other hand, exactly how many times the recall is more important than precision in this case is an empirical issue. Answering this question would require user case evaluation experiments and as such is out of the scope of this article.

We have focused on maximising the F_2 measure, in two known approaches to its calculation, supposing recall is twice as important as precision.[1] Note that this *intended bias* towards recall has nothing to do with the imbalance of the classes in the training data (definitions vs. non-definitions). Thus, instead of maximising the ratio of correctly classified examples, we maximise the values of both F_2 measures by selecting the classification threshold appropriately. This means we may favour one of the classes over another, if this leads to an increase of the value of the chosen measure.

For the results, cf. Table IV and Table V. There is a clear improvement in terms of the chosen measures that can be explained by the accompanying four figures. The peaks of the graphs, especially those representing F-measures on the copula definition subcorpora, are located quite far to the right from 0.5 (that is, the default value used when there is no optimisation). However, we have to be well aware of the needs: fine-tuning the threshold value to one measure might also make the results with regard to other measures worse. On the other hand, both measures tend to peak close to each other (and not always close to 0.5). That may suggest that in case of an unknown corpus it is better to optimise with regard to a similar measure than not to optimise at all—as the graph for this corpus might peak far away from 0.5. The question what is a similar measure and what is not remains open though, and we will not attempt to address it in this paper.

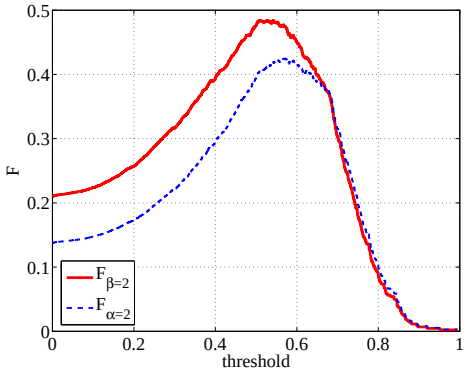

Fig. 1. F-measures values with respect to the chosen threshold on the dataset with all definitions and preselected set of attributes

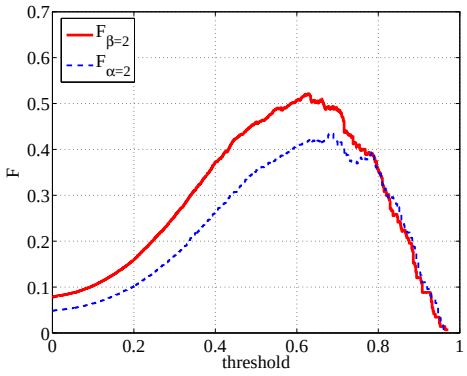

Fig. 2. F-measures values with respect to the chosen threshold on the dataset with copula definitions only and preselected set of attributes

V. Applying a manually created grammar

As described in [2], applying a very simple partial grammar before the classifiers such as Naïve Bayes, decision trees ID3 and C4.5, AdaBoostM1 with Decision Stump, Support Vector Machines and lazy classifier IB1 significantly improves the results. In that approach all sentences rejected by the grammar are unconditionally marked as non-definitions, and only those accepted by the grammar may be marked as definitions in the Machine Learning stage.

[1]There are different views in literature on how this should be done. For instance, [6] uses F_β, which is in fact the same formula as F_α, but giving quadratic importance to the parameter instead of linear: $F_{\alpha=4} = F_{\beta=2}$. Something that could be interpreted as third version is used for instance in [7], but at a closer look it turns out to be effectively equivalent to F_α – used also in [2] and some medical papers – but encoding the intended result differently: $F_{0.5}$ is used to denote a measure giving equal weights to precision and recall (as $F_{\alpha=1}$), and $F_{0.75}$ is said to value recall three times more than precision (as $F_{\alpha=3}$).

TABLE IV
GAIN IN F-MEASURES FROM OPTIMISING THE THRESHOLD, ALL DEFINITIONS

	preselected attributes				χ^2 attributes			
threshold	precision	recall	$F_{\alpha=2}$	$F_{\beta=2}$	precision	recall	$F_{\alpha=2}$	$F_{\beta=2}$
no optimisation	21.37	69.04	39.60	47.74	20.60	65.20	37.87	45.50
optimised for $F_{\alpha=2}$	27.80	57.69	**42.47**	47.48	26.38	55.13	**40.44**	45.26
optimised for $F_{\beta=2}$	22.30	68.50	40.52	**48.43**	23.48	60.99	39.80	**46.22**

TABLE V
GAIN IN F-MEASURES FROM OPTIMISING THE THRESHOLD, COPULA DEFINITIONS ONLY

	preselected attributes				χ^2 attributes			
threshold	precision	recall	$F_{\alpha=2}$	$F_{\beta=2}$	precision	recall	$F_{\alpha=2}$	$F_{\beta=2}$
no optimisation	16.50	84.40	35.59	46.30	17.60	81.70	36.90	47.27
optimised for $F_{\alpha=2}$	25.42	67.78	**43.58**	50.84	31.78	60.56	**46.52**	51.27
optimised for $F_{\beta=2}$	22.68	77.22	42.86	**52.14**	28.26	65.00	45.35	**51.59**

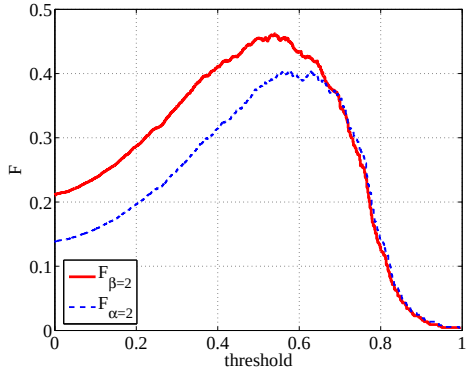

Fig. 3. F-measures values with respect to the chosen threshold on the dataset with all definitions and χ^2 attribute selection

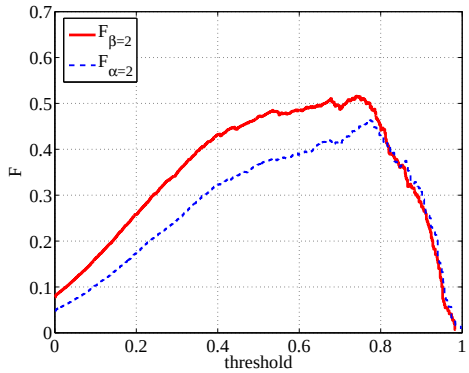

Fig. 4. F-measures values with respect to the chosen threshold on the dataset with copula definitions only and χ^2 attribute selection

Even such a primitive grammar (that could also be described as a set of pattern-matching rules) rejected a significant part of potential false positives, i.e. those sentences that would be mistakenly marked as definitions by the classifiers. Thus, a significant relative increase of precision (for different classifiers from 36% up to 72%) was observed, accompanied only by a minor decrease of recall (between 3.4% and 7.5%). In terms of $F_{\alpha=2}$ measure, the increase was between 21% and 40%.

TABLE VI
THE RESULTS OF THE SIMPLE PARTIAL GRAMMAR BY ITSELF

corpus	precision	recall	$F_{\alpha=2}$	$F_{\beta=2}$
whole	9.10	89.60	22.69	32.36
copula	3.30	99.40	9.28	14.57

In case of the Balanced Random Forest classifier the gain turned out to be much smaller, up to 3.4%—cf. Table VII. Note that we look at the relative gain, not the absolute values of precision, recall and F-measures, because those numbers are not directly comparable: in [2] the experiments were not performed as a ten-fold cross-validation, but on a separate training and test subcorpora.

TABLE VII
GAIN OF APPLYING A SIMPLE GRAMMAR BEFORE THE CLASSIFIERS, ALL DEFINITIONS

pre-filtering	$F_{\alpha=2}$ standard	$F_{\alpha=2}$ optimised	$F_{\beta=2}$ standard	$F_{\beta=2}$ optimised
no	39.60	42.47	47.74	48.43
yes	**40.95**	**43.09**	**48.62**	**49.30**
relative gain	3.4%	1.5%	1.8%	1.8%

Balanced Random Forest classifier, especially with threshold optimisation, is inherently good enough not to require the initial pre-filtering by the grammar. We conclude that there is not that many potential false positives to be removed. This is clear when we look at the results for copula definitions in Table VIII.

TABLE VIII
GAIN OF APPLYING A SIMPLE GRAMMAR BEFORE THE CLASSIFIERS, COPULA DEFINITIONS

pre-filtering	$F_{\alpha=2}$ standard	$F_{\alpha=2}$ optimised	$F_{\beta=2}$ standard	$F_{\beta=2}$ optimised
no	35.59	43.58	46.30	52.14
yes	**36.35**	**43.67**	**47.07**	**52.29**
relative gain	2.1%	0.2%	1.7%	0.3%

VI. PREVIOUS WORK

There is a substantial previous work on definition extraction, as this is a subtask of many applications, including terminol-

ogy extraction [8], the automatic creation of glossaries [9], [10], question answering [11], [12], learning lexical semantic relations [13], [14] and the automatic construction of ontologies [15]. 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 and other Germanic languages, as well as French.

For Polish, first attempts at constructing definition extraction systems are described—in the context of other Slavic languages—in [16], and improved results are presented in [17]. [2] describes improvements achieved by using a simple manually created grammar.

The first NLP applications of the plain Random Forests are apparently those reported in [18] and in [19], 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.

The use of Balanced Random Forests for definition extraction in textual datasets was proposed in [1].

VII. Conclusion

For definition extraction, the Balanced Random Forest classification method may be further improved by optimising the threshold above which we classify a given sentence as a definition. With this improvement, the algorithm does not gain much more from initial filtering of the data by a very simple, high-recall hand-crafted grammar, as it was in case of other ML classifiers we experimented with; however, the gain, being small, is always positive, so it may be worth trying, when the best possible result is desired, even at the cost of complicating the algorithm and lengthening the execution time. The same applies to using a more advanced set of attributes that are selected for each training set separately instead of using a preselected single set.

References

[1] Ł. Kobyliński and A. Przepiórkowski, "Definition extraction with balanced random forests," in *Proceedings of the 6th International Conference on Natural Language Processing, GoTAL 2008, Gothenburg, Sweden*, ser. Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag, 2008.

[2] Ł. Degórski, M. Marcińczuk, and A. Przepiórkowski, "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, 2008.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[4] J. R. Quinlan, *Programs for Machine Learning*. Morgan Kaufmann, 1993.

[5] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," University of California, Berkeley, Tech. Rep. 666, 2004, http://www.stat.berkeley.edu/tech-reports/666.pdf.

[6] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2008.

[7] M. Jansche, "Maximum expected f-measure training of logistic regression models," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*. Vancouver: ACL, 2005, pp. 692–699.

[8] J. Pearson, "The expression of definitions in specialised texts: a corpus-based analysis," in *Proceedings of the Seventh Euralex International Congress*, M. Gellerstam, J. Järborg, S. G. Malmgren, K. Norén, L. Rogström, and C. Papmehl, Eds., Göteborg, 1996, pp. 817–824.

[9] J. L. Klavans and S. Muresan, "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.

[10] ——, "Evaluation of the DEFINDER system for fully automatic glossary construction," in *Proceedings of AMIA Symposium 2001*, 2001.

[11] S. Miliaraki and I. Androutsopoulos, "Learning to identify single-snippet answers to definition questions," in *Proceedings of COLING 2004*, Geneva, Switzerland, 2004, pp. 1360–1366.

[12] I. Fahmi and G. Bouma, "Learning to identify definitions using syntactic features," in *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*, 2006.

[13] V. Malaisé, P. Zweigenbaum, and B. Bachimont, "Detecting semantic relations between terms in definitions," in *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, S. Ananadiou and P. Zweigenbaum, Eds., Geneva, Switzerland, 2004, pp. 55–62.

[14] A. Storrer and S. Wellinghoff, "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*. Genoa: ELRA, 2006.

[15] S. Walter and M. Pinkal, "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, 2006, pp. 20–28. [Online]. Available: http://www.aclweb.org/anthology/W/W06/W06-0203

[16] A. Przepiórkowski, Ł. Degórski, M. Spousta, K. Simov, P. Osenova, L. Lemnitzer, V. Kuboň, and B. Wójtowicz, "Towards the automatic extraction of definitions in Slavic," in *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing at ACL 2007*, J. Piskorski, B. Pouliquen, R. Steinberger, and H. Tanev, Eds., 2007, pp. 43–50.

[17] A. Przepiórkowski, Ł. Degórski, and B. Wójtowicz, "On the evaluation of Polish definition extraction grammars," in *Proceedings of the 3rd Language & Technology Conference*, Z. Vetulani, Ed., Poznań, Poland, 2007, pp. 473–477.

[18] R. D. Nielsen and S. Pradhan, "Mixing weak learners in semantic parsing," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, D. Lin and D. Wu, Eds. Barcelona: ACL, 2004, pp. 80–87.

[19] P. Xu and F. Jelinek, "Random forests in language modeling," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, D. Lin and D. Wu, Eds. Barcelona: ACL, 2004, pp. 325–332.

[20] D. Lin and D. Wu, Eds., *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*. Barcelona: ACL, 2004.