Kernel-based Logical and Relational Learning with kLog for Hedge Cue Detection

Mathias Verbeke[1], Paolo Frasconi[2], Vincent Van Asch[3], Roser Morante[3], Walter Daelemans[3], and Luc De Raedt[1]

[1] Department of Computer Science, Katholieke Universiteit Leuven, Belgium
{mathias.verbeke,luc.deraedt}@cs.kuleuven.be
[2] Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, Italy
p-f@dsi.unifi.it
[3] Department of Linguistics, Universiteit Antwerpen, Belgium
{vincent.vanasch,roser.morante,walter.daelemans}@ua.ac.be

Abstract. Hedge cue detection is a Natural Language Processing (NLP) task that consists of determining whether sentences contain hedges. These linguistic devices indicate that authors do not or cannot back up their opinions or statements with facts. This binary classification problem, i.e. distinguishing factual versus uncertain sentences, only recently received attention in the NLP community. We use kLog, a new logical and relational language for kernel-based learning, to tackle this problem. We present results on the CoNLL 2010 benchmark dataset that consists of a set of paragraphs from Wikipedia, one of the domains in which uncertainty detection has become important. Our approach shows competitive results compared to state-of-the-art systems.

Keywords: statistical relational learning, kernel methods, natural language learning

1 Introduction

Information Extraction (IE) is a subdomain of Natural Language Processing (NLP) concerned with the automatic extraction of structured, factual information from unstructured or semi-structured machine-readable texts. Since it has been shown that a number of IE tasks, such as question answering systems [3] and IE from biomedical texts [4, 5], benefit from being able to distinguish facts from unreliable or uncertain information, research about hedge cue detection has increased in recent years.

Hedge cues are linguistic devices that indicate whether information is being presented as uncertain or unreliable within a text [1, 2]. They are lexical resources used by the author to indicate caution or uncertainty towards the content of the text, and in this sense they can be taken as signals of the presence of an author's opinion or attitude. Hedge cues can be expressed by several word classes: modal verbs (e.g. *can, may*), verbs (e.g. *seem, appear*), adjectives (e.g. *possibly, likely*), etc. Furthermore hedge cues can be expressed by multiword expressions, i.e.

expressions that contain more than a word, with non-compositional meaning, i.e. the meaning of the expression cannot be derived from the individual meanings of the words that form the expression. This can be seen from Example 1, where *call into question* is a multiword hedge cue.

(1) The low results {**call into question** the applicability of this method}.

Neither the verb *call* nor the noun *question* are hedge cues on their own, but the whole phrase conveys a speculative meaning, which explains why the sentence would be marked as uncertain.

Recently, the NLP community has shown interest in problems that involve analysing language beyond the propositional meaning of sentences, i.e. whether the sentence is true or false. Apart from performing well established NLP tasks such as parsing or semantic role labeling, there is a growing interest in tasks that involve processing non-propositional aspects of meaning, i.e. opinions, attitudes, emotions, figurative meaning. To perform these tasks, the local token-based approaches based on the lexico-syntactic features of individual words do not suffice. The broader context of words at sentence or discourse level has to be considered in order to account for aspects of meaning that are expressed by certain combinations of words, like "call into question" in the sentence above. Performing hedge cue detection involves finding the linguistic expressions that express hedging. In many cases it is not possible to know whether a word belongs to a hedge cue without taking into account its context. This formed our motivation to use kLog [8], a new language for logical and relational learning with kernels. kLog is able to transform the relational representations into graph-based representations and then apply kernel methods. The question we would like to answer in this paper is whether a logical and relational learning approach - and kLog in particular - is able to process contextual aspects of language. As we will see, the results indicate that kLog is suitable for this task.

This paper is organized as follows. In section 2, we give an overview of related work. kLog and the modeling approach for the task at hand are presented in section 3. Section 4 discusses the experimental findings. Finally, in section 5, we conclude and present our future work.

2 Related Work

Although the term *hedging* was already introduced by Lakoff in 1972 [1], and has been studied from a theoretical linguistics point since two decades [2], the interest from the computational linguistics (CL) community only arose in recent years. Light et al. [6] introduced the problem of identifying speculative language in bioscience literature. The authors used a hand-crafted list of hedge cues to identify speculative sentences in MEDLINE abstracts. They also presented two systems for automatic classification of sentences in the abstracts; one based on support vector machines (SVMs), the other one based on substring matching. Medlock and Briscoe [4] extended this work and discussed the specificities of hedge classification as a weakly supervised machine learning task and presented

a probabilistic learning model. Furthermore they offered an improved and expanded set of annotation guidelines and provided a publicly available data set. Based on this work, Medlock [7] carried out experiments using an expanded feature space and novel representations. Szarvas [5] followed Medlock and Briscoe [4] in classifying sentences as being speculative or non-speculative. He developed a Maximum Entropy classifier that incorporates bigrams and trigrams in the feature representation and performs a reranking based feature selection procedure. Kilicoglu and Bergler [14] applied a linguistically motivated approach to the same classification task by using knowledge from existing lexical resources and incorporating syntactic patterns. Additionally, hedge cues were weighted by automatically assigning them an information gain measure and by assigning weights semi–automatically based on their types and centrality to hedging.

Ganter and Strube [15] were the first ones in developing a system for automatic detection of sentences containing *weasels* in Wikipedia. As Ganter and Strube indicated, weasels are closely related to hedges and private states, i.e. states that are not open to objective observation or verification. They experimented with two classifiers, one based on words preceding the weasel and another one based on syntactic patterns. The similar results of the two classifiers on sentences extracted from Wikipedia showed that word frequency and distance to the weasel tag provide sufficient information. However, the classifier that used syntactic patterns outperformed the classifier based on words on data manually re-annotated by the authors, suggesting that the syntactic patterns detected weasels that have not yet been tagged.

The increased attention for hedge detection reflects in the fact that it became a subtask of the BioNLP Shared Task in 2009 [9], and the topic of the Shared Task at CoNLL 2010 [10]. The latter comprised two levels of analysis: the focus of task 1 was learning to detect sentences containing uncertainty, whereas the objective of task 2 was resolving the in-sentence scope of hedge cues. As indicated above, the present paper will focus on task 1. As noted in [10], the approaches to this task can be classified into two major categories. Several systems approached the problem as a sentence classification problem and used a bag-of-words (BoW) feature representation. Also the individual tokens of the sentence can be classified, instead of the overall sentence. In a postprocessing step, the sentences that contain hedge cues are classified as uncertain.

3 Approach

The presented approach can be seen as a variant of the sentence classification approach that is able to represent both the lexico-syntactic information as well as the sequence information and dependency relationships. This is realized in an extended feature space, which is calculated from graph kernels. This section first shortly describes kLog in section 3.1 and subsequently describes the approach taken for the hedge cue detection task (section 3.2).

3.1 kLog

kLog is a logical and relational language for kernel-based learning, that is embedded in Prolog, and builds upon and links together concepts from database theory, logic programming and learning from interpretations. It is based on a novel technique called *graphicalization* that transforms relational representations into graph based ones and derives features from a grounded entity/relationship diagram using graph kernels after which a statistical learning algorithm can be applied. The general workflow is depicted in Figure 1 and will be explained by means of the approach for the task at hand.

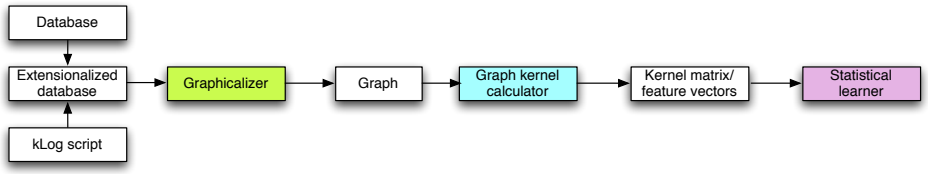

Fig. 1. General kLog workflow

3.2 Model

kLog is built upon a logical and relational data representation and is rooted in the entity-relationship (E/R) model [17]. For the problem under consideration, the E/R-model is shown in Figure 2. It gives an abstract representation of the interpretations, which are sentences in the current problem. They consist of a number of consecutive words w, for which the order is represented by the next relation. There are also dependency relations between certain words, which represent the structure of syntactic relations between the words of a sentence. This is modeled by depHead, where depRel specifies the type of the dependency.

(2) Often the response variable may not be continuous but rather discrete.

In Example 2 an example dependency relation exists between the determiner *the* and the noun *variable*, where the first is a noun modifier of the latter. This is indicated by dh(nmod) in the figure.

Other properties of the word that are taken into account as features are the word string itself, its lemma, the Part-of-Speech tag (i.e. the linguistic type of the word in the sentence), the chunk tag (which indicates that a word is part of a subsequence of constituents) and a binary feature that represents whether the word is part of a predefined list of speculative strings. weaselSentence represents the target relation.

This E/R model representation can be transformed into a kLog script that describes (the structure of) the data. Table 1 shows a (part of an) example interpretation z, that is a grounded version of the E/R-model, where e.g. w(w1,'often',rb,i-dvp,1,'often') specifies an entity where w1 is the identifier and the other arguments represent the properties. next(w1,w2) gives an example relation between

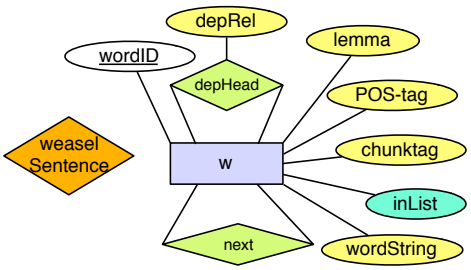

Fig. 2. E/R diagram modeling the hedge cue detection task

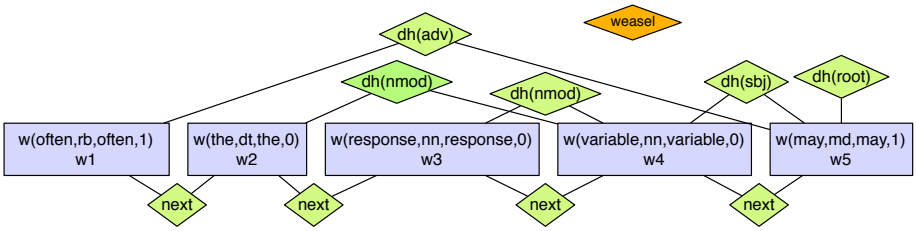

Fig. 3. Graphicalization G_z of interpretation z (Table 1)

w1 and w2. These interpretations are then graphicalized, i.e. transformed into graphs. This can be interpreted as unfolding the E/R diagram over the data, for which an example is given in Figure 3. It represents the graphicalization of the interpretation in Table 1. This forms the input to the next level, where graph learning is applied to convert these graphicalized interpretations into extended, high-dimensional feature vectors using a graph kernel. The result is a propositional learning setting, for which any statistical learner can be used. Currently, kLog employs LibSVM [11] for parameter learning.

4 Results and Discussion

Dataset For our experiments, the dataset we used is the CoNLL 2010 Shared Task dataset [10] on Wikipedia, one of the current benchmark datasets for hedge cue resolution. The Wikipedia paragraphs were selected based on the hedge cue (called *weasels* in Wikipedia) tags that were added by the Wikipedia editors,

Table 1. Example interpretation z

```
wwc(2).
next(w1,w2).
w(w1,'often',rb,i-advp,1,'often').
dh(w1,w5,adv).
next(w2,w3).
w(w2,'the',dt,i-np,0,'the').
dh(w2,w4,nmod).
next(w3,w4).
w(w3,'response',nn,i-np,0,'response').
dh(w3,w4,nmod).
...
```

which were subsequently manually annotated. A sentence is considered uncertain if it contains at least one weasel cue. The proportion of training and test data, and their respective class ratios can be found in Table 2.

Table 2. Number of instances per class in the training and test partitions of the CoNLL Shared Task Wikipedia dataset

	Train	Test
Certain	8627	7288
Uncertain	2484	2234
Total	11111	9634

Preprocessing For preprocessing, the approach of Morante et al. [12] was followed, in which the input files where converted into a token-per-token representation, following the standard CoNLL format [21]. Hereby a sentence consists of a sequence of tokens, each one starting on a new line. Consequently the data was processed with the Memory Based Shallow Parser (MBSP) [13] in order to obtain lemmas, part-of-speech tags, and syntactic chunks, and with the MaltParser [16] to obtain dependency trees.

kLog Parametrization Learning in kLog is performed using an extension of the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [20]. NSPDK is a decomposition kernel, where the parts are pairs of subgraphs. These are defined by the relation $R_{r,d}(A_v, B_u, G)$ between two rooted graphs A_v, B_u and a graph G, which selects all pairs of neighborhood graphs of radius r whose roots are at distance d in a given graph G.

The decomposition kernel $\kappa_r, d(G, G')$ on the relation $R_{r,d}$ is defined as

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_v,\, B_u \in R^{-1}_{r,d}(G) \\ A'_{v'},\, B'_{u'} \in R^{-1}_{r,d}(G')}} \delta(A_v, A'_{v'})\delta(B_u, B'_{u'}) \tag{3}$$

If δ is an exact matching kernel, $\kappa_{r,d}$ counts the number of identical pairs of neighboring graphs of radius r at distance d between two graphs. This results in the following (non-normalized) definition of NSPDK:

$$K(G, G') = \sum_r \sum_d \kappa_{r,d}(G, G'). \tag{4}$$

For efficiency reasons a zero-extension of K, obtained by imposing an upper bound on the radius and the distance parameter, was introduced: $K_{r^*,d^*}(G, G') = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} \kappa_{r,d}(G, G')$, that is, we limit the sum of the $\kappa_{r,d}$ kernels for all increasing values of the radius (distance) parameter up to a maximum given value r^* (d^*).

From the kernel definition it follows that the distance and radius parameters may influence the results. Consequently, it is important to make a deliberate

choice during parametrization. For the task at hand, expert knowledge and the literature suggest using bigrams (one word before *or* one word after the word in focus) or trigrams (one word before *and* one word after the word in focus), since unigrams include too little context, and 5-grams introduce too much noise. This is confirmed by a 10-fold cross-validation on the training set that was performed for parametrization, using all combinations of distances 0, 1, 2 and radii 0, 1, 2 for the kLog hyperparameters. The setting with both distance and radius set to 1 gave the best results (60.59 F-measure, where we took 60.2, the F-measure of the top performing system in the CoNLL shared task, as decision threshold).

The distance indicates the number of hops between the two subgraphs to be compared for feature generation. Distance 1 implies that the subgraph (in orange) is compared to another subgraph, for which its center is only 1 hop away from the center of this subgraph. The radius determines the size of the subgraph; the subgraphs in the figure are centered around the dh(nmod) relation and word $w4$ (i.e. *variable*), and the edges in bold indicate the span.

Important to note is that also the modeling plays an important role, which is demonstrated by means of the dependency relation dh. A subgraph around word $w3$ with the current parameter settings will not only take the neighboring words $w3$ and $w4$ into consideration during feature generation, but also the dependency relation dh(nmod) between $w2$ and $w4$. Furthermore, with this parameter settings more pairs of words are taken into account than just bigrams, for which the words need to be adjacent. As can be seen, with a distance and radius of 1 also the (non-adjacent) words $w2$ and $w4$, respectively *the* and *variable*, are considered through the dh relation. This makes it able to take more context into account, which demonstrates the power of the graph-based approach of kLog. Also the background knowledge can have an impact on the results, as we will discuss next.

For the statistical learner, we used the linear kernel of LibSVM, for which we optimized the regularization parameter and the class weighting parameters as part of the cross-validation process.

Background Knowledge Since kLog is built on deductive databases, besides listing the tuples and atoms of an interpretation explicitly, they can also be deduced using rules. In kLog this is realized by using intensional signatures, whereby tuples can be defined through definite clauses as in Prolog. This is very useful to introduce additional background knowledge in the learning process. Since the newly constructed ground facts in the database are used to construct graphs from which features are derived during graphicalization, this amounts to the ability of constructing features in a declarative fashion. For the Wikipedia dataset, we introduced the following piece of background knowledge, which retains only the words that appear in a predefined list of weasel words compiled from the training data, together with their two surrounding words in the sentence and the respective lemmas and POS-tags.

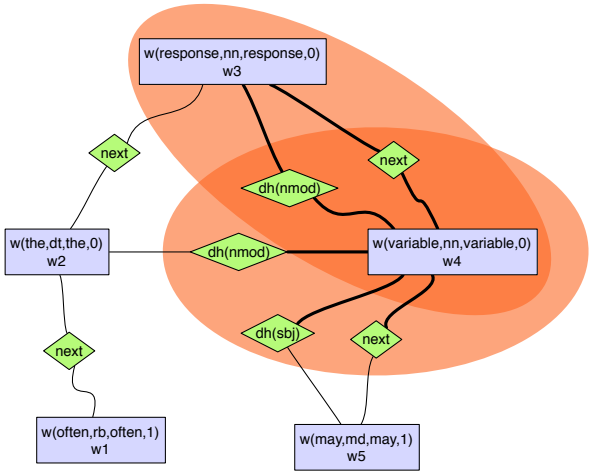

Fig. 4. Part of graphicalization G_z with parameters distance 1 and radius 1

$\texttt{cw}(\texttt{CW},\texttt{L},\texttt{P}) :- \texttt{w}(\texttt{W},\texttt{L},\texttt{P},\_,1,\_),\texttt{atomic\_concat}(\texttt{c},\texttt{W},\texttt{CW}).$
$\texttt{leftof}(\texttt{CW},\texttt{L},\texttt{P}) :- \texttt{cw}(\texttt{W},\_,\_),\texttt{atomic\_concat}(\texttt{c},\texttt{W},\texttt{CW}),\texttt{next}(\texttt{W1},\texttt{W}),\texttt{w}(\texttt{W1},\texttt{L},\texttt{P},\_,\_,\_).$
$\texttt{rightof}(\texttt{CW},\texttt{L},\texttt{P}) :- \texttt{cw}(\texttt{W},\_,\_),\texttt{atomic\_concat}(\texttt{c},\texttt{W},\texttt{CW}),\texttt{next}(\texttt{W},\texttt{W1}),\texttt{w}(\texttt{W1},\texttt{L},\texttt{P},\_,\_,\_).$

This resulted in an increase of 2.66 in F-measure, from 58.82 to 61.48, which shows the advantage of the declarative feature construction through the introduction of additional background knowledge. This is - combined with the powerful graph kernel - one of the main strengths of kLog.

Results The results of our approach are listed in Table 3, together with results of the 5 best listed participants in the CoNLL-Shared Task 2010. Figure 5 shows the precision/recall curve for kLog with optimal parameter settings and the data points for the top 5 CoNLL systems, where the labels correspond with their position in the ranking. As can be noted, kLog outperforms the systems in terms of F-measure.

Table 3. Evaluation performance in terms of precision, recall and F1 of the top 5 CoNLL 2010 systems and the kLog approach for the Wikipedia dataset

Official Rank	System	P	R	F
-	**kLog**	67.04	56.77	61.48
1	Georgescul	72.0	51.7	60.2
2	Ji[1]	62.7	55.3	58.7
3	Chen	68.0	49.7	57.4
4	Morante	80.6	44.5	57.3
5	Zhang	76.6	44.4	56.2

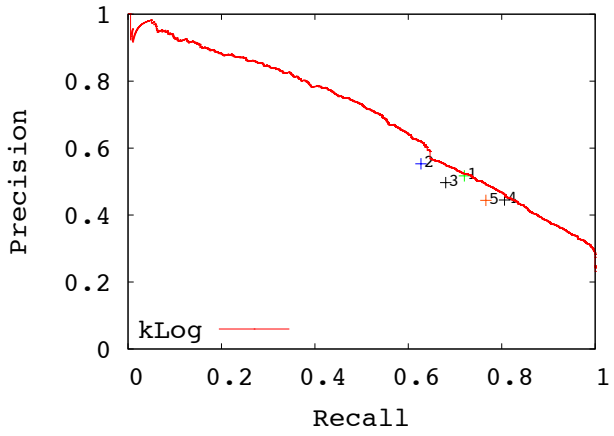

Fig. 5. Precision/recall curve for kLog and the individual points for the top 5 CoNLL systems in Table 3 (numbers correspond with ranking)

5 Conclusions and Future Work

We presented a new approach for solving the hedge cue resolution task, based on kernel-based logical and relational learning with kLog. Our system outperforms state-of-the-art systems, which can be ascribed to the graphicalization step, which transforms the data into a graph-based format. This enables us to use graph kernels on a full relational representation. Since the linguistic relations between words in a sentence can be represented as a graph structure, kLog seems to have the appropriate characteristics for CL problems. Furthermore, the ability to construct features in a declarative fashion through the introduction of additional background knowledge showed to have a positive influence on the results.

In future work, we plan to test the generalizability of our approach on another dataset for this task, i.e. scientific texts from the biomedical domain, which have a different, more structured writing style and sentence structure. This opens the way to applying a cross dataset training phase, which showed improved results for one of the participants in the shared task. Also the addition of new (linguistic) background knowledge requires further investigation, for which we will start from an extensive error analysis of the obtained results. Due to the promising results, the goal is to test this approach also on more challenging NLP problems and to perform a detailed comparison with the state-of-the-art approaches.

[1] Remark that this system used a cross dataset approach, in which also the CoNLL 2010 biological dataset was used to train the system.

6 Acknowledgements

This research is funded by the Research Foundation Flanders (FWO-project G.0478.10 - Statistical Relational Learning of Natural Language) and made possible through financial support from the University of Antwerp (GOA project BIOGRAPH). The authors would like to thank Fabrizio Costa and Kurt De Grave for their valuable feedback.

References

1. Lakoff, G: Hedges: A study in meaning criteria and the logic of fuzzy concepts. Journal of Philosophical Logic, 2 (1973)
2. Hyland, K.: Hedging in scientific research articles. Amsterdam (1998)
3. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: Proc. of CoNLL 2003. Edmonton (2003)
4. Medlock, B., Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: Proc. ACL 2007. Prague (2007)
5. Szarvas, G.: Hedge classification in biomedical texts with a weakly supervised selection of keywords. In: Proc. of ACL 2008. Ohio (2008).
6. Light, M., Qiu, X., Srinivasan, P.: The language of bioscience: facts, speculations, and statements in between. In: Proc. of HLT-NAACL 2004 – BioLINK. (2004).
7. Medlock, B.: Exploring hedge identification in biomedical literature. Journal of Biomedical Informatics, 41 (2008)
8. Frasconi, P., Costa F., De Raedt L., De Grave K.: kLog - a language for logical and relational learning with kernels, Technical Report, `http://www.dsi.unifi.it/~paolo/ps/klog.pdf` (2011)
9. Kim, J., Ohta, T., Pyysalo, S., Kano, Y., Tsujii, J.: Overview of BioNLP'09 shared task on event extraction. In: Proc. of the Workshop on Current Trends in Biomedical NLP – Shared Task. Colorado (2009)
10. Farkas, R., Vincze, V., Móra, G., Csirik, J., Szarvas, G.: The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text. In: Proc. of CoNLL 2010 – Shared Task. Uppsala (2010)
11. Chang, C.-C., Lin C.-J.: LIBSVM: a library for support vector machines (2001)
12. Morante, R., Van Asch, V., Daelemans W.: Memory-based resolution of in-sentence scopes of hedge cues. In: Proc. of CoNLL 2010 – Shared Task. Uppsala (2010)
13. Daelemans, W., van den Bosch, A.: Memory-based language processing. Cambridge University Press, Cambridge (2005)
14. Kilicoglu, H., Bergler, S.: Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. In: BMC Bioinformatics (2008)
15. Ganter, V., Strube, M.: Finding hedges by chasing weasels: Hedge detection using Wikipedia tags and shallow linguistic features. In: Proc. of ACL-IJCNLP 2009 Conference Short Papers. Suntec (2009)
16. Nivre, J.: Inductive Dependency Parsing. In: Text, Speech and Language Technology. Springer (2006)
17. Garcia-Molina, H., Ullman, J. D., Widom, J.: Database Systems: The Complete Book. Prentice Hall Press (2008)
18. Vincze, V., Szarvas G., Farkas, R., Móra G., Csirik, J.: The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. In: BMC Bioinformatics (2008)

19. Velldal, E.: Detecting Uncertainty in Biomedical Literature: A Simple Disambiguation Approach Using Sparse Random Indexing. In: Proc. of the Fourth International Symposium on Semantic Mining in Biomedicine (SMBM). Cambridgeshire (2010)
20. Costa, F., De Grave, K.: Fast neighborhood subgraph pairwise distance kernel. In: Proc. of the 26th International Conference on Machine Learning. Haifa (2010)
21. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL-X '06). New York (2006)