A Local Alignment Kernel in the Context of NLP

Sophia Katrenko
Informatics Institute
University of Amsterdam
the Netherlands
katrenko@science.uva.nl

Pieter Adriaans
Informatics Institute
University of Amsterdam
the Netherlands
pietera@science.uva.nl

Abstract

This paper discusses local alignment kernels in the context of the relation extraction task. We define a local alignment kernel based on the Smith-Waterman measure as a sequence similarity metric and proceed with a range of possibilities for computing a similarity between elements of sequences. We propose to use distributional similarity measures on elements and by doing so we are able to incorporate extra information from the unlabeled data into a learning task. Our experiments suggest that a LA kernel provides promising results on some biomedical corpora largely outperforming a baseline.

1 Introduction

Relation extraction is one of the tasks in the natural language processing which is constantly revisited. To date, there are many methods which have been proposed to tackle it. Such approaches often benefit from using syntactic information (Bunescu and Mooney, 2006) and background knowledge (Sekimizu et al., 1998). However, it would be interesting to employ additional information not necessarily contained in the training set. This paper presents a contribution to the work on relation extraction by combining statistical information with string distance measures. In particular, we propose to use a local alignment kernel to detect relations.

The paper is organized as follows. We start with the definition of a local alignment kernel and show how it is defined on the Smith-Waterman measure. We proceed by discussing how a substitution matrix can be constructed in the context of natural language processing tasks. Once a method is described, we turn to the task of relation extraction and present an experimental part. We conclude by mentioning possible future directions.

2 A Local Aligment Kernel

Kernel methods are widely used for a variety of natural language processing task, starting from PoS tagging to information extraction. Many of the approaches employ the idea of combining kernels together which leads to a convolution kernel (Haussler, 1999). The examples of convolution methods being successfully used in NLP are kernels based on dependency trees and shallow parsing (Moschitti, 2006; Zelenko et al., 2003). Local alignment (LA) kernels also belong to the family of convolution kernels but have not yet been applied to NLP problems.

Although the approaches listed above proved to be accurate, they only use kernels which are designed by computing inner products between vectors of sequences. Intuitively, methods using more elaborate measures of similarity could provide better results but kernels defined on such measures are not necessarily positive semi-definite. Recent work in the biomedical field shows that it is possible to design valid kernels based on a similarity measure by solving the diagonal dominance problem to ensure the semi-definiteness (Saigo et al., 2006). To illustrate it, Saigo et al. (2004) consider the Smith-Waterman (SW) similarity measure (Smith and Waterman, 1981) which has often been used to compare two sequences of amino acids. The original Smith-Waterman score is calculated to achieve the best local alignment allow-

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 417–424
Manchester, August 2008

ing gaps.

The Smith-Waterman measure belongs to the string distance metrics which can be divided into term-based, edit-distance and HMM based metrics (Cohen et al., 2003). Term-based distances such as metrics based on TF-IDF score, consider a pair of word sequences as two sets of words neglecting their order. In contrast, edit string distances treat the entire sequences and, by comparing them, calculate the minimal number of the transformation operations converting a sequence x into a sequence y. Examples of string edit distances are Levenshtein, Needleman-Wunsch and Smith-Waterman metrics. Levenshtein distance has been used in natural language processing field as a component in the variety of tasks, including semantic role labeling (Tjong Kim Sang et al., 2005), construction of the paraphrase corpora (Dolan et al., 2004), evaluation of machine translation output (Leusch et al., 2003), and others. Smith-Waterman distance is mostly used in the biological domain, there are, however, some applications of a modified Smith-Waterman distance to the text data as well (Monge and Elkan, 1996), (Cohen et al., 2003). HMM based measures present probabilistic extensions of edit distances.

According to the definition of a LA kernel, two strings (sequences) are considered similar if they have many local alignments with high scores (Saigo et al., 2006). Given two sequences $x = x_1x_2 \dots x_n$ and $y = y_1y_2 \dots y_m$ of length n and m respectively, Smith-Waterman distance is defined as the local alignment score of their best alignment:

$$SW(x,y) = \max_{\pi \in A(x,y)} s(x,y,\pi) \qquad (1)$$

In the equation above, $s(x,y,\pi)$ is a score of a local alignment π of sequence x and y and A denotes the set of all possible alignments. This definition can be rewritten by means of dynamic programming as follows:

$$SW(i,j) = max \begin{cases} 0 \\ SW(i-1,j-1) + d(x_i,y_j) \\ SW(i-1,j) - G \\ SW(i,j-1) - G \end{cases} \qquad (2)$$

In Equation 2, $d(x_i,y_j)$ denotes a substitution score between two elements x_i and y_j and G stands for a gap penalty.

Unfortunately, the direct application of the Smith-Waterman score will not result in the valid kernel. A valid kernel based on the Smith-Waterman distance can be defined by summing up the contribution of all possible alignments as follows (Saigo et al., 2004):

$$K_{LA} = \sum_{\pi \in A(x,y)} \epsilon^{\beta \cdot s(x,y,\pi)} \qquad (3)$$

It is shown that in the limit a LA kernel approaches the Smith-Waterman score:

$$\lim_{\beta \to \infty} ln\Big(\frac{1}{\beta} K_{LA}(x,y)\Big) = SW(x,y) \qquad (4)$$

The results in the biological domain suggest that kernels based on the Smith-Waterman distance are more relevant for the comparison of amino acids than string kernels. It is not clear whether this holds when applied to natural language processing tasks. In our view, it depends on the parameters which are used, such as a substitution matrix and the penalty gaps. It has been shown by Saigo (2006) that given a substitution matrix which is equal to the identity matrix and no penalty gap, the Smith-Waterman score is a string kernel.

2.1 How to define a substitution matrix $d(\cdot,\cdot)$?

In order to use Smith-Waterman distance for our purposes, it is necessary to define a substitution matrix. Unlike a matrix in the original Smith-Waterman measure defined by the similarity of amino acids or a substitution matrix in (Monge and Elkan, 1996) based on the exact and approximate match of two characters (for instance, m and n), we introduce a matrix based on the distributional similarity measures. In our view, they are the most natural measures for the text data. In other words, if we are to compare any two words given two sequences of words, the elements sharing the same contexts should be more similar to each other than those that do not. In the context of the LA kernel, such metrics can be especially useful. Consider, for instance, the labeled sequences of words which are used as input for a machine learning method. To compare the sequences, we have to be able to compare their elements, i.e. words. Now, if there are some words in the test data that do not occur in the training set, it is still possible to carry out a comparison if additional evidence is present. Such evidence can be provided by the distributional similarity metrics.

There are a number of measures proposed over the years, including such metrics as cosine, dice coefficient, and Jaccard distance. Distributional similarity measures have been extensively studied in (Lee, 1999; Weeds et al., 2004).

We have chosen the following metrics: *dice*, *cosine* and *l2 (euclidean)* whose definitions are given in Table 1. Here, x_i and y_j denote two words and c stands for a context. Similarly to (Lee, 1999), we use unsmoothed relative frequencies to derive probability estimates P. In the definition of the *dice* coefficient, $F(x_i) = \{c : P(c|x_i) > 0\}$. We are mainly interested in the symmetric measures ($d(x_i, y_j) = d(y_j, x_i)$) because of a symmetric positive semi-definite matrix required by kernel methods. Consequently, such measures as the skew divergence were excluded from the consideration (Lee, 1999).

The Euclidean measure as defined in Table 1 does not necessarily vary from 0 to 1. It was therefore normalized by dividing an *l2* score in Table 1 by a maximum score and retracting it from 1.

Measure	**Formula**
cosine	$d(x_i, y_j) = \frac{\sum_c P(c\|x_i) \cdot P(c\|y_j)}{\sqrt{\sum_c P(c\|x_i)^2 \sum_c P(c\|y_j)^2}}$
dice	$d(x_i, y_j) = \frac{2 \cdot F(x_i) \cap F(y_j)}{F(x_i) \cup F(y_j)}$
l2	$d(x_i, y_j) = \sqrt{\sum_c (P(c\|x_i) - P(c\|y_j))^2}$

Table 1: Distributional similarity measures.

3 A relation extraction task

Many approaches to relation extraction consider syntactic information. In this paper we focus on dependency parsing. The experiments in the past have already shown syntactic analysis to be useful for relation learning. Like other work we extract a path between two nodes which correspond to the arguments of a binary relation. We also assume that each analysis results in a tree and since it is an acyclic graph, there exists only one path between each pair of nodes. We do not consider, however, the other structures that might be derived from the full syntactic analysis as in, for example, subtree kernels (Moschitti, 2006).

Consider, for instance, an example of interaction among proteins (5) whose syntactic analysis is given in Fig. 1. Here, there is a relation between *Cbf3* and three proteins, *Cbf3a*, *Cbf3b* and *Cbf3c* expressed by a verb *contain*. We believe that this partial information extracted from the dependency trees should be sufficient for relation learning and can be used as a representation for the learning method.

(5) Cbf3 contains three proteins, Cbf3a, Cbf3b and Cbf3c.

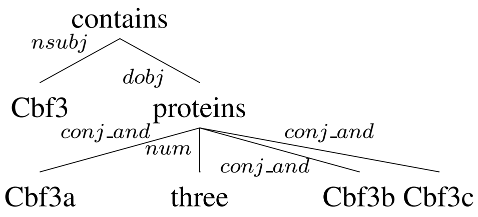

Figure 1: Stanford parser output

Representation: dependency paths
Cfb3 $\overset{nsubj}{\rightarrow}$ contains $\overset{dobj}{\leftarrow}$ proteins $\overset{conj\_and}{\leftarrow}$ Cbf3a
Cfb3 $\overset{nsubj}{\rightarrow}$ contains $\overset{dobj}{\leftarrow}$ proteins $\overset{conj\_and}{\leftarrow}$ Cbf3b
Cfb3 $\overset{nsubj}{\rightarrow}$ contains $\overset{dobj}{\leftarrow}$ proteins $\overset{conj\_and}{\leftarrow}$ Cbf3c

4 Experiments

4.1 Set-up

Data We use two corpora which both come from the biomedical field and contain annotations of either interacting proteins `BC-PPI` (1,000 sentences)[1] or the interactions among proteins and genes `LLL`[2] (77 sentences in the training set and 87 in the test set) (Nédellec, 2005). The `BC-PPI` corpus was created by sampling sentences from the BioCreAtive challenge, the `LLL` corpus was composed by querying Medline with the term *Bacillus subtilis*. The difference between the two corpora lies in the directionality of interactions. The former corpus contains both symmetric and asymmetric interactions while in the latter they are strictly asymmetric. We analyzed the `BC` corpus with the Stanford parser. [3] The `LLL` corpus has already been preprocessed by the Link parser.

To estimate distributional similarity, we use `TREC 2006 Genomics` collection (Hersch, 2006) which contains 162,259 documents from

[1] available from `http://www2.informatik.hu-berlin.de/~hakenber/`

[2] available from `http://genome.jouy.inra.fr/texte/LLLchallenge/`

[3] available from `http://nlp.stanford.edu/software/lex-parser.shtml\#Download`

49 journals. All documents have been preprocessed by removing HTML-tags, citations in the text and reference sections and stemmed by the Porter stemmer (van Rijsbergen et al., 1980). Furthermore, the query-likelihood approach with Dirichlet smoothing (Chen, 1996) is used to retrieve document passages given a query. All words occurring in the set of input sequences are fed as queries. Immediate context surrounding each pair of words is used as features to calculate distributional similarity of these words. We set the context window to ±2 (2 tokens to the right and 2 tokens to the left of a word in focus) and do not perform any kind of further preprocessing such as PoS tagging.

Recall that in Section 2.1 we defined a substitution matrix solely based on the words. However, the representation we employ also contains information on syntactic functions and directions (Fig. 1). To take this into account, we revise the definition of $d(\cdot,\cdot)$. We assume sequences $x = x_1x_2\ldots x_n$ and $y = y_1y_2\ldots y_m$ to contain words ($x_i \in W$) and syntactic functions accompanied by direction ($x_i \notin W$). Then,

$$d'(x_i, y_j) = \begin{cases} d(x_i, y_j) & x_i, y_j \in W \\ 1 & x_i, y_j \notin W \;\&\; x_i = y_j \\ 0 & x_i, y_j \notin W \;\&\; x_i \neq y_j \\ 0 & x_i \in W \;\&\; y_j \notin W \\ 0 & x_i \notin W \;\&\; y_j \in W \end{cases} \quad (6)$$

Baseline To test how well local alignment kernels perform compared to the kernels proposed in the past, we implemented a method described in (Bunescu and Mooney, 2005) as a baseline. Here, similarly to our approach, the shortest path between relation arguments is extracted and a kernel between two sequences (paths) x and y is computed as follows:

$$K(x, y) = \begin{cases} 0 & m \neq n \\ \prod_{i=1}^{n} f(x_i, y_i) & m = n \end{cases} \quad (7)$$

In Eq. 7, $f(x_i, y_i)$ is the number of common features shared by x_i and y_i. Bunescu and Mooney (2005) use several features such as word ($protesters$), part of speech tag (NNS), generalized part of speech tag ($Noun$), and entity type (e.g., $PERSON$) if applicable. In addition, a direction feature (→ or ←) is employed. In our experiments we also use lemma, part of speech tag and direction but we do not consider an entity type or negative polarity of items.

Kernels that we compute are used together with LibSVM (Chang and Lin, 2001) to detect hyperplanes separating positive examples from the negative ones. Before plugging all kernel matrices into LibSVM, they were normalized as in Eq. 8.

$$K(x', y') = \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}} + 1 \quad (8)$$

To compute LA matrices we use the distributed ASCI supercomputer 3 (DAS-3) [4] which allows us to speed up the process of sequence comparison. In particular, because of symmetricity of the resulting matrices for n sequences we need to carry out $n(n-1)/2$ comparisons to build a matrix. Computations are done in parallel by reserving a number of nodes of DAS-3 and concatenating the outputs later on.

4.2 Experiment I: Distributional measures and their impact on the final performance

Distributional similarity measures have been used for various tasks in the past. For instance, (Lee, 1999) employs them to detect similar nouns based on the verb-object cooccurrence pairs. The results suggest the *Jaccard's* coefficient to be one of the best performing measures followed by some others including *cosine*. *Euclidean* distance fell into the group with the largest error rates. It is of considerable interest to test whether these metrics have an impact on the performance of a LA kernel. We do not employ *Jaccard's* measure but the *dice* coefficient is monotonic in it.

While computing a distributional similarity, it may happen that a given word x does not occur in the corpus. To handle such cases, we always set $d(x, x) = 1$. To estimate distributional similarity, a number of hits returned by querying the TREC collection is set to 500. Gaps are defined through the gaps opening and extension costs. In our experiments, the gap opening cost is set to 1.2, the extension cost to 0.2 and the scaling parameter β to 1.

The 10-fold cross-validation results on the `BC-PPI` corpus are presented in Table 2 and on the `LLL` training data set in Table 3. The LA kernel

[4] `http://www.cs.vu.nl/das3`

based on the distributional similarity measures performs significantly better than the baseline. In contrast to the baseline, it is able to handle sequences of different lengths including gaps. According to the Eq. 7, a comparison of any two sequences of different lengths results in the 0-score. Nevertheless it still yields high recall while precision is much lower. Interestingly, the results of the shortest path approach on the ACE corpus (Bunescu and Mooney, 2005) were reversed by boosting precision while decreasing recall.

Method	Pr,%	R,%	F_1,%
LAK-dice	75.56	79.72	77.56
LAK-cosine	76.4	**80.66**	78.13
LAK-l2	**77.56**	79.31	**78.42**
Baseline	32.04	75.63	45.00

Table 2: Results on the BC-PPI data set

At first glance, the LA kernel based on the distributional similarity measures that we selected provides similar performance. We can notice that the $l2$ metric seems to be the best performing measure. On the BC-PPI data, the method based on the $l2$ measure outperforms the methods based on $dice$ and on $cosine$ but the differences are not significant.

On the LLL data set, the LA method using distributional similarity measures significantly outperforms both baselines and also yields better results than an approach based on shallow linguistic information (Giuliano et al., 2006). Giuliano et al. (2006) use no syntactic information. Recent work reported in (Fundel, 2007) also uses dependency information but in contrast to our method, it serves as representation on which extraction rules are defined.

The choice of the distributional measure does not seem to affect the overall performance very much. But in contrast to the BC-PPI data set, the kernels which use $dice$ and $cosine$ measures significantly outperform the one based on $l2$ (paired t-test, α = 0.01).

Method	Pr,%	R,%	F_1,%
LAK-dice	**74.25**	87.94	**80.51**
LAK-cosine	73.99	**88.23**	80.48
LAK-l2	69.28	87.6	77.37
(Fundel, 2007)	68	83	75
(Giuliano et al., 2006)	62.10	61.30	61.70
Baseline	39.02	100.00	56.13

Table 3: Results on the LLL data set

coreferences	Pr,%	R,%	F_1,%
with (LAK-dice)	**60.00**	31.00	**40.90**
w/o (LAK-dice)	**71.00**	50.00	58.60
with (Giuliano et al., 2006)	29.00	31.00	30.00
w/o (Giuliano et al., 2006)	54.80	**62.90**	58.60

Table 4: Results on the LLL test data set

We also verified how well our method performs on the LLL test data. Surpisingly, precision is still high (for both subsets, with co-references and without them) while recall suffers. We hypothesize that it is due to the fact that for some sentences only incomplete parses are provided and, consequently, no dependency paths between the entities are found. For 91 out of 567 possible interaction pairs generated on the test data, there is no dependency path extracted. In contrast, work reported in (Giuliano et al., 2006) does not make use of syntactic information which on the data without coreferences yields higher recall.

4.2.1 Experiment Ia: Impact of distributional measures estimation

We believe that accuracy of LA kernel crucially depends on the substitution matrix, i.e. an accurate estimate of distributional similarity. In most cases, to obtain accurate estimates it is needed to use a large corpus. However, it is unclear whether differences in the estimates derived from corpora of different sizes would affect the overall performance of the LA kernel. To investigate it, we conducted several experiments by varying a number of retrieved passages.

Table 5 contains the most similar words to *adhere*, *expression* and *sigF* detected by the $dice$ measure in descending order (by varying the number of passages retrieved per query). While the order of the most similar words for *sigF* does not change very much from one setting to another, estimates for *adhere* and *expression* depend more on the number of passages retrieved. Moreover, not only the actual ordering changes, but also the number of similar words does. For instance, while there are only four words similar to *adhere* found when 100 passages per each query are used, already 12 similar words to *adhere* are detected when the count of extracted documents is set to 1,000 passages per query.

We also notice that the most similar words to

	adhere	**expression**	**sigF**
dice@100	contribute, belong, bind, map	processing, overlap, production, localization, sequestration	cotC, tagA, rocG, tagF, whiG
dice@500	contribute, belong, bind, occur, result	end, localization, presence, processing, absence	sigE, comK, cotC, sigG, tagA
dice@1,000	contribute, bind, convert, occur, belong	presence, assembly, localization, processing, activation	sigE, comK, cotC, sigG, tagA
dice@1,500	bind, contribute, convert, correspond, belong	localization, assembly, presence, activation, processing	sigE, comK, cotC, sigG, tagA

Table 5: Top 5 similar words (LLL data set)

sigF are all named entities. Even though *sigF* does not occur in the training data, we can still hypothesize that it is likely to be a target of the relation because of *sigE*, *cotC* and *tagA*. These three genes can be found in the training set and they are usually targets (second argument) of the interaction relation.

Table 6 shows results on the LLL data set by varying the size of the data used for estimation of distributional similarity ($dice$ measure). We observe the decrease in precision and in recall when increasing the number of hits to 1,500. Changing the number of hits from 500 to 1,000 results in a subtle increase in recall.

Size	Pr,%	R,%	F_1,%
dice@500	74.25	87.94	80.51
dice@1,000	74.38	**88.02**	80.62
dice@1,500	69.87	86.85	77.43

Table 6: Estimation settings for the LLL data set

Size	Pr,%	R,%	F_1,%
dice@100	75.56	79.72	77.58
dice@500	76.72	**81.01**	78.8
dice@1,000	76.56	80.78	78.61

Table 7: Estimation settings for the BC-PPI data set

The results on the BC-PPI data set show a similar tendency. However the observed differences are not statistically significant in the latter case. These subtle changes in recall and precision can be attributed to the relatively low absolute values of the similarity scores. For instance, even though an order of similar words in Table 5 changes while increasing the data used for estimation, a difference between the absolute values can be quite small.

4.2.2 Experiment Ib: Impact of the scaling parameter β

Saigo et al. (2004) have already shown that the parameter β has the significant impact on the results accuracy. We have also carried out some preliminary experiments by setting the opening gap to 12, the extension gap to 2 and by varying the parameter β. The kernel matrices were normalized as in Eq. 8. The results on the BC-PPI data set (dice500) are given in Table 8.

β	Pr,%	R,%	F_1,%
0.5	17.72	94.87	29.85
1	38.84	89.42	54.14
10	67.72	76.67	71.90

Table 8: Impact of β on the performance on the BC-PPI data set

The results indicate that decreasing β leads to the decrease in the overall performance. However, if the values of gap penalties are lower and β is set to 1, the results are better. This suggests that the final performance of the LA kernel is influenced by a combination of parameters and their choice is crucial for obtaining the good performance.

5 Related Work

We have already mentioned some relevant work on relation extraction while introducing the local alignment kernel. Most work done for relation extraction considers binary relations in sentential context (McDonald, 2005). Current techniques for relation extraction include hand-written patterns (Sekimizu et al., 1998), kernel methods (Zelenko et al., 2003), pattern induction methods (Snow et al., 2005), and finite-state automata (Pustejovsky et al., 2002).

Kernel methods have become very popular in natural language processing in general and for learning relations, in particular (Culotta and Sorensen, 2004). There are many kernels defined for the text data. For instance, string kernels are special kernels which consider inner products of all subsequences from the given sequences of elements (Lodhi et al., 2002). They can be further extended to syllable kernels which proved to perform well for text categorization (Saunders et al., 2002).

For relation learning, Zelenko et al.(2003) use shallow parsing in conjunction with contiguous and non-contiguous kernels to learn relations. Bunescu et al.(2006) define several kernels to accomplish the same task. First, they introduce the sequential kernels and show that such method out-performs the longest match approach. Next, Bunescu et al. (2006) propose kernels for the paths in dependency trees (which is referred to as a shortest path between two arguments of a given relation). In this paper we used their method based on dependency parsing as one of the baselines. Giuliano (2006) takes this approach further by defining several kernels using local context and sentential context. An advantage of Giuliano's method (2006) lies in the simpler representation which does not use syntactic structure. In this case, even if parsing fails on certain sentences, it is still possible to handle them.

6 Conclusions

We presented a novel approach to relation extraction which is based on the local alignments of sequences. To compare two sequences, additional information is used which is not necessarily contained in the training data. By employing distributional measures we obtain a considerable improvement over two baselines and work reported before.

The choice of a distributional similarity measure does not seem to affect the overall performance very much. Based on the experiments we have conducted, we conclude that the LA kernel using *dice* and *cosine* measures perform similarly on the LLL data set and the BC-PPI corpus. On the LLL corpus, the LA kernel employing *l2* shows a significant decrease in performance. But concerning statistical significance, the method using *dice* significantly outperforms the one based *l2* measure only on LLL corpus while there is no significant improvement on the BC-PPI data set noticed.

We use contextual information to measure distributional similarity. In this setting any two words can be compared no matter which parts of speech they belong to. As dependency paths contain various words along with nouns and verbs, other methods often mentioned in the literature would be more difficult to use. However, in the future we are going to extend this approach by using syntactically analyzed corpora and by estimating distributional similarity from it. It would allow us to use more accurate estimates and to discriminate between lexically ambiguous words. Similarity measures on the words that belong to other parts of speech can be still estimated using the local context only.

Acknowledgments

The authors thank Vera Hollink, Victor de Boer and anonymous reviewers for their valuable comments. This work was carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl). This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

References

Yevgeny (Eugene) Agichtein. 2005. Extracting Relations from Large Text Collections. *Ph.D. Thesis, Columbia University.*

Razvan C. Bunescu and Raymond J.Mooney. 2006. Extracting Relations from Text. From Word Sequences to Dependency Paths. *In book "Text Mining and Natural Language Processing", Anne Kao and Steve Poteet (Eds).*

Razvan C. Bunescu and Raymond J.Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. *In Proceedings of the Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, BC.*

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a library for support vector machines. *Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm*

Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. *In ACL'96.*

William W. Cohen, Pradeep Ravikumar and Stephen Fienberg. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks. *In IIWeb 2003, pages 73-78.*

Aron Culotta and Jeffrey Sorensen. 2003. Dependency Tree Kernels for Relation Extraction. *In ACL 2003.*

William B. Dolan, Chris Quirk and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. *In Proceedings of COLING 2004, Geneva, Switzerland.*

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. RelEx - Relation Extraction using dependency parse trees. *In Bioinformatics, vol. 23, no. 3.*

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. *In EACL 2006.*

David Haussler. 1999. Convolution Kernels on Discrete Structures. *UC Santa Cruz Technical Report UCS-CRL-99-10.*

William Hersch, Aaron M. Cohen, Phoebe Roberts and Hari K. Rakapalli. 2006. TREC 2006 Genomics Track Overview. *In Proceedings of the 15th Text Retrieval Conference.*

Lillian Lee. 1999. Measures of distributional similarity. *In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, pages 25-32.*

G. Leusch, N. Ueffing and H. Ney. 2003. A Novel String-to-String Distance Measure with Applications to Machine Translation Evaluation. *In Machine Translation Summit IX, New Orleans, LO, pages 240-247.*

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Christianini, and Chris Watkins. 2002. Text Classification using String Kernels. *In Journal of Machine Learning Research, 2, pages 419-444.*

Ryan McDonald. 2005. Extracting Relations from Unstructured Text. *Technical Report: MS-CIS-05-06.*

Alvaro E. Monge and Charles Elkan. 1996. The Field Matching Problem: Algorithms and Applications. *In KDD 1996, pages 267-270.*

Alessandro Moschitti. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. *In ECML 2006, pages 318-329.*

Cl. Nédellec. 2005. Learning Language in Logic - Genic Interaction Extraction Challenge. *In Proceedings of the Learning Language in Logic workshop.*

J. Pustejovsky, J. Castano, J. Zhang, B. Cochran, M. Kotecki. 2002. Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations. *Pacific Symposium on Biocomputing.*

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, James Dowdall, Christos Andronis, Andreas Persidis, and Ourania Konstanti. 2004. Mining relations in the GENIA corpus. *In "Second European Workshop on Data Mining and Text Mining for Bioinformatics", in conjunction with ECML/PKDD 2004, September.*

Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda and Tatsuya Akutsu. 2004. Protein homology detection using string alignment kernels. *In "Bioinformatics", vol. 20 no. 11, pages 1682-1689.*

Hiroto Saigo, Jean-Philippe Vert, and Tatsuya Akutsu. 2006. Optimizing amino acid substitution matrices with a local alignment kernel. *In "BMC Bioinformatics", 7:246.*

C. Saunders, H. Tschach, and J. Shawe-Taylor. 2002. Syllables and other String Kernel Extensions. *In Proceesings of the Nineteenth International Conference on Machine Learning (ICML'02).*

T. Sekimizu, H. Park, and J. Tsujii. 1998. Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. *In Genome Informatics.*

T. F. Smith and M. S. Waterman. 1987. Identification of Common Molecular Subsequences. *In J. Mol. Biol. 147, 195–197.*

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. *NIPS 17.*

Erik Tjong Kim Sang, Sander Canisius, Antal van den Bosch and Toine Bogers. 2005. Applying spelling error correction techniques for improving semantic role labeling. *In Proceedings of the Ninth Conference on Natural Language Learning, CoNLL-2005, June 29-30, 2005, Ann Arbor, MI.*

Lonneke van der Plas and Jörg Tiedemann. 2006. Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. *In Proceedings of ACL/Coling.*

C. J. van Rijsbergen, S. E. Robertson and M. F. Porter. 1980. New models in probabilistic information retrieval. *London: British Library. (British Library Research and Development Report, no. 5587).*

Julie Weeds, David Weir and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. *In Proceedings of CoLing 2004.*

Dmitry Zelenko, Ch. Aone, and A. Richardella. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research 3 (2003), 1083-1106.*