

Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature

Claudio Giuliano and Alberto Lavelli and Lorenza Romano

ITC-irst

Via Sommarive, 18

38050, Povo (TN)

Italy

{giuliano,lavelli,romano}@itc.it

Abstract

We propose an approach for extracting relations between entities from biomedical literature based solely on shallow linguistic information. We use a combination of kernel functions to integrate two different information sources: (i) the whole sentence where the relation appears, and (ii) the local contexts around the interacting entities. We performed experiments on extracting gene and protein interactions from two different data sets. The results show that our approach outperforms most of the previous methods based on syntactic and semantic information.

1 Introduction

Information Extraction (IE) is the process of finding relevant entities and their relationships within textual documents. Applications of IE range from Semantic Web to Bioinformatics. For example, there is an increasing interest in automatically extracting relevant information from biomedical literature. Recent evaluation campaigns on bio-entity recognition, such as BioCreAtIvE and JNLPBA 2004 shared task, have shown that several systems are able to achieve good performance (even if it is a bit worse than that reported on news articles). However, relation identification is more useful from an applicative perspective but it is still a considerable challenge for automatic tools.

In this work, we propose a supervised machine learning approach to relation extraction which is applicable even when (deep) linguistic processing is not available or reliable. In particular, we explore a kernel-based approach based solely on shallow linguistic processing, such as tokeniza-

tion, sentence splitting, Part-of-Speech (PoS) tagging and lemmatization.

Kernel methods (Shawe-Taylor and Cristianini, 2004) show their full potential when an explicit computation of the feature map becomes computationally infeasible, due to the high or even infinite dimension of the feature space. For this reason, kernels have been recently used to develop innovative approaches to relation extraction based on syntactic information, in which the examples preserve their original representations (i.e. parse trees) and are compared by the kernel function (Zelenko et al., 2003; Culotta and Sorensen, 2004; Zhao and Grishman, 2005).

Despite the positive results obtained exploiting syntactic information, we claim that there is still room for improvement relying exclusively on shallow linguistic information for two main reasons. First of all, previous comparative evaluations put more stress on the deep linguistic approaches and did not put as much effort on developing effective methods based on shallow linguistic information. A second reason concerns the fact that syntactic parsing is not always robust enough to deal with real-world sentences. This may prevent approaches based on syntactic features from producing any result. Another related issue concerns the fact that parsers are available only for few languages and may not produce reliable results when used on domain specific texts (as is the case of the biomedical literature). For example, most of the participants at the Learning Language in Logic (LLL) challenge on Genic Interaction Extraction (see Section 4.2) were unable to successfully exploit linguistic information provided by parsers. It is still an open issue whether the use of domain-specific treebanks (such as the Genia treebank¹)

¹<http://www-tsujii.is.s.u-tokyo.ac.jp/>

can be successfully exploited to overcome this problem. Therefore it is essential to better investigate the potential of approaches based exclusively on simple linguistic features.

In our approach we use a combination of kernel functions to represent two distinct information sources: the global context where entities appear and their local contexts. The whole sentence where the entities appear (*global context*) is used to discover the presence of a relation between two entities, similarly to what was done by Bunescu and Mooney (2005b). Windows of limited size around the entities (*local contexts*) provide useful clues to identify the roles of the entities within a relation. The approach has some resemblance with what was proposed by Roth and Yih (2002). The main difference is that we perform the extraction task in a single step via a combined kernel, while they used two separate classifiers to identify entities and relations and their output is later combined with a probabilistic global inference.

We evaluated our relation extraction algorithm on two biomedical data sets (i.e. the AImed corpus and the LLL challenge data set; see Section 4). The motivations for using these benchmarks derive from the increasing applicative interest in tools able to extract relations between relevant entities in biomedical texts and, consequently, from the growing availability of annotated data sets. The experiments show clearly that our approach consistently improves previous results. Surprisingly, it outperforms most of the systems based on syntactic or semantic information, even when this information is manually annotated (i.e. the LLL challenge).

2 Problem Formalization

The problem considered here is that of identifying interactions between genes and proteins from biomedical literature. More specifically, we performed experiments on two slightly different benchmark data sets (see Section 4 for a detailed description). In the former (AImed) gene/protein interactions are annotated without distinguishing the type and roles of the two interacting entities. The latter (LLL challenge) is more realistic (and complex) because it also aims at identifying the roles played by the interacting entities (agent and target). For example, in Figure 1 three entities are mentioned and two of the six ordered pairs of

entities actually interact: ($\sigma(K)$, $cwlH$) and ($gerE$, $cwlH$).

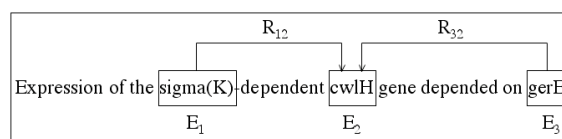


Figure 1: A sentence with two relations, R_{12} and R_{32} , between three entities, E_1 , E_2 and E_3 .

In our approach we cast relation extraction as a classification problem, in which examples are generated from sentences as follows.

First of all, we describe the complex case, namely the protein/gene interactions (LLL challenge). For this data set entity recognition is performed using a dictionary of protein and gene names in which the type of the entities is unknown. We generate examples for all the sentences containing at least two entities. Thus the number of examples generated for each sentence is given by the combinations of distinct entities (N) selected two at a time, i.e. ${}_N C_2$. For example, as the sentence shown in Figure 1 contains three entities, the total number of examples generated is ${}_3 C_2 = 3$. In each example we assign the attribute *CANDIDATE* to each of the *candidate interacting* entities, while the other entities in the example are assigned the attribute *OTHER*, meaning that they do not participate in the relation. If a relation holds between the two candidate interacting entities the example is labeled 1 or 2 (according to the roles of the interacting entities, agent and target, i.e. to the direction of the relation); 0 otherwise. Figure 2 shows the examples generated from the sentence in Figure 1.

Example #1	Expression of the $\sigma(K)$ -dependent $cwlH$ gene depended on $gerE$	Label = 1
	CANDIDATE CANDIDATE OTHER	
Example #2	Expression of the $\sigma(K)$ -dependent $cwlH$ gene depended on $gerE$	Label = 0
	CANDIDATE OTHER CANDIDATE	
Example #3	Expression of the $\sigma(K)$ -dependent $cwlH$ gene depended on $gerE$	Label = 2
	OTHER CANDIDATE CANDIDATE	

Figure 2: The three *protein-gene* examples generated from the sentence in Figure 1.

Note that in generating the examples from the sentence in Figure 1 we did not create three neg-

ative examples (there are six potential ordered relations between three entities), thereby implicitly under-sampling the data set. This allows us to make the classification task simpler without losing information. As a matter of fact, generating examples for each ordered pair of entities would produce two subsets of the same size containing similar examples (differing only for the attributes CANDIDATE and OTHER), but with different classification labels. Furthermore, under-sampling allows us to halve the data set size and reduce the data skewness.

For the *protein-protein* interaction task (AImed) we use the correct entities provided by the manual annotation. As said at the beginning of this section, this task is simpler than the LLL challenge because there is no distinction between types (all entities are proteins) and roles (the relation is symmetric). As a consequence, the examples are generated as described above with the following difference: an example is labeled 1 if a relation holds between the two candidate interacting entities; 0 otherwise.

3 Kernel Methods for Relation Extraction

The basic idea behind kernel methods is to embed the input data into a suitable feature space \mathcal{F} via a mapping function $\phi : \mathcal{X} \rightarrow \mathcal{F}$, and then use a linear algorithm for discovering nonlinear patterns. Instead of using the explicit mapping ϕ , we can use a kernel function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, that corresponds to the inner product in a feature space which is, in general, different from the input space.

Kernel methods allow us to design a modular system, in which the kernel function acts as an interface between the data and the learning algorithm. Thus the kernel function is the only domain specific module of the system, while the learning algorithm is a general purpose component. Potentially any kernel function can work with any kernel-based algorithm. In our approach we use Support Vector Machines (Vapnik, 1998).

In order to implement the approach based on shallow linguistic information we employed a linear combination of kernels. Different works (Gliozzo et al., 2005; Zhao and Grishman, 2005; Culotta and Sorensen, 2004) empirically demonstrate the effectiveness of combining kernels in this way, showing that the combined kernel always improves the performance of the individual ones.

In addition, this formulation allows us to evaluate the individual contribution of each information source. We designed two families of kernels: *Global Context* kernels and *Local Context* kernels, in which each single kernel is explicitly calculated as follows

$$K(x_1, x_2) = \frac{\langle \phi(x_1), \phi(x_2) \rangle}{\|\phi(x_1)\| \|\phi(x_2)\|}, \quad (1)$$

where $\phi(\cdot)$ is the embedding vector and $\|\cdot\|$ is the 2-norm. The kernel is normalized (divided) by the product of the norms of embedding vectors. The normalization factor plays an important role in allowing us to integrate information from heterogeneous feature spaces. Even though the resulting feature space has high dimensionality, an efficient computation of Equation 1 can be carried out explicitly since the input representations defined below are extremely sparse.

3.1 Global Context Kernel

In (Bunescu and Mooney, 2005b), the authors observed that a relation between two entities is generally expressed using only words that appear simultaneously in one of the following three patterns:

Fore-Between: tokens before and between the two candidate interacting entities. For instance: *binding of [P₁] to [P₂], interaction involving [P₁] and [P₂], association of [P₁] by [P₂].*

Between: only tokens between the two candidate interacting entities. For instance: *[P₁] associates with [P₂], [P₁] binding to [P₂], [P₁], inhibitor of [P₂].*

Between-After: tokens between and after the two candidate interacting entities. For instance: *[P₁] - [P₂] association, [P₁] and [P₂] interact, [P₁] has influence on [P₂] binding.*

Our global context kernels operate on the patterns above, where each pattern is represented using a *bag-of-words* instead of sparse subsequences of words, PoS tags, entity and chunk types, or WordNet synsets as in (Bunescu and Mooney, 2005b). More formally, given a relation example R , we represent a pattern P as a row vector

$$\phi_P(R) = (tf(t_1, P), tf(t_2, P), \dots, tf(t_l, P)) \in \mathbb{R}^l, \quad (2)$$

where the function $tf(t_i, P)$ records how many times a particular token t_i is used in P . Note that,

this approach differs from the standard bag-of-words as punctuation and stop words are included in ϕ_P , while the entities (with attribute CANDIDATE and OTHER) are not. To improve the classification performance, we have further extended ϕ_P to embed n-grams of (contiguous) tokens (up to $n = 3$). By substituting ϕ_P into Equation 1, we obtain the n-gram kernel K_n , which counts common uni-grams, bi-grams, ..., n-grams that two patterns have in common². The *Global Context* kernel $K_{GC}(R_1, R_2)$ is then defined as

$$K_{FB}(R_1, R_2) + K_B(R_1, R_2) + K_{BA}(R_1, R_2), \quad (3)$$

where K_{FB} , K_B and K_{BA} are n-gram kernels that operate on the Fore-Between, Between and Between-After patterns respectively.

3.2 Local Context Kernel

The type of the candidate interacting entities can provide useful clues for detecting the agent and target of the relation, as well as the presence of the relation itself. As the type is not known, we use the information provided by the two local contexts of the candidate interacting entities, called left and right local context respectively. As typically done in entity recognition, we represent each local context by using the following basic features:

Token The token itself.

Lemma The lemma of the token.

PoS The PoS tag of the token.

Orthographic This feature maps each token into equivalence classes that encode attributes such as capitalization, punctuation, numerals and so on.

Formally, given a relation example R , a local context $L = t_{-w}, \dots, t_{-1}, t_0, t_{+1}, \dots, t_{+w}$ is represented as a row vector

$$\psi_L(R) = (f_1(L), f_2(L), \dots, f_m(L)) \in \{0, 1\}^m, \quad (4)$$

where f_i is a feature function that returns 1 if it is active in the specified position of L , 0 otherwise³. The *Local Context* kernel $K_{LC}(R_1, R_2)$ is defined as

$$K_{left}(R_1, R_2) + K_{right}(R_1, R_2), \quad (5)$$

where K_{left} and K_{right} are defined by substituting the embedding of the left and right local context into Equation 1 respectively.

²In the literature, it is also called *n-spectrum* kernel.

³In the reported experiments, we used a context window of ± 2 tokens around the candidate entity.

Notice that K_{LC} differs substantially from K_{GC} as it considers the ordering of the tokens and the feature space is enriched with PoS, lemma and orthographic features.

3.3 Shallow Linguistic Kernel

Finally, the *Shallow Linguistic* kernel $K_{SL}(R_1, R_2)$ is defined as

$$K_{GC}(R_1, R_2) + K_{LC}(R_1, R_2). \quad (6)$$

It follows directly from the explicit construction of the feature space and from closure properties of kernels that K_{SL} is a valid kernel.

4 Data sets

The two data sets used for the experiments concern the same domain (i.e. gene/protein interactions). However, they present a crucial difference which makes it worthwhile to show the experimental results on both of them. In one case (AImed) interactions are considered symmetric, while in the other (LLL challenge) agents and targets of genic interactions have to be identified.

4.1 AImed corpus

The first data set used in the experiments is the AImed corpus⁴, previously used for training protein interaction extraction systems in (Bunescu et al., 2005; Bunescu and Mooney, 2005b). It consists of 225 Medline abstracts: 200 are known to describe interactions between human proteins, while the other 25 do not refer to any interaction. There are 4,084 protein references and around 1,000 tagged interactions in this data set. In this data set there is no distinction between genes and proteins and the relations are symmetric.

4.2 LLL Challenge

This data set was used in the Learning Language in Logic (LLL) challenge on Genic Interaction extraction⁵ (Nedéllec, 2005). The objective of the challenge was to evaluate the performance of systems based on machine learning techniques to identify gene/protein interactions and their roles, agent or target. The data set was collected by querying Medline on *Bacillus subtilis* transcription and sporulation. It is divided in a training set (80 sentences describing 271 interactions) and a

⁴<ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/interactions.tar.gz>

⁵<http://genome.jouy.inra.fr/texte/LLLchallenge/>

test set (87 sentences describing 106 interactions). Differently from the training set, the test set contains sentences without interactions. The data set is decomposed in two subsets of increasing difficulty. The first subset does not include coreferences, while the second one includes simple cases of coreference, mainly appositions. Both subsets are available with different kinds of annotation: *basic* and *enriched*. The former includes word and sentence segmentation. The latter also includes manually checked information, such as lemma and syntactic dependencies. A dictionary of named entities (including typographical variants and synonyms) is associated to the data set.

5 Experiments

Before describing the results of the experiments, a note concerning the evaluation methodology. There are different ways of evaluating performance in extracting information, as noted in (Lavelli et al., 2004) for the extraction of slot fillers in the Seminar Announcement and the Job Posting data sets. Adapting the proposed classification to relation extraction, the following two cases can be identified:

- One Answer per Occurrence in the Document – OAOD (each individual occurrence of a protein interaction has to be extracted from the document);
- One Answer per Relation in a given Document – OARD (where two occurrences of the same protein interaction are considered one correct answer).

Figure 3 shows a fragment of tagged text drawn from the AImed corpus. It contains three different interactions between pairs of proteins, for a total of seven occurrences of interactions. For example, there are three occurrences of the interaction between *IGF-IR* and *p52Shc* (i.e. number 1, 3 and 7). If we adopt the OAOD methodology, all the seven occurrences have to be extracted to achieve the maximum score. On the other hand, if we use the OARD methodology, only one occurrence for each interaction has to be extracted to maximize the score.

On the AImed data set both evaluations were performed, while on the LLL challenge only the OAOD evaluation methodology was performed because this is the only one provided by the evaluation server of the challenge.

Our findings can be summarized as follows: (i) the tyrosine kinase activity of the **IGF-IR**_[1,2] is essential for the interaction with **p52Shc**_[1] and **IRS-1**_[2], (ii) **p52Shc**_[3] and **IRS-1**_[4] bind to the **IGF-IR**_[3,4] in the NPEY-juxtamembrane motif, (iii) contrary to **p52Shc**, **IRS-1**_[5] binds also to the major autophosphorylation sites (Tyr-1131, -1135, and -1136) of the **IGF-IR**_[5], and (iv) the amino-terminal domain of **p52Shc**_[6,7] is required for its association with the **IR**_[6] and the **IGF-IR**_[7].

Figure 3: Fragment of the AImed corpus with all proteins and their interactions tagged. The protein names have been highlighted in bold face and their same subscript numbers indicate interaction between the proteins.

5.1 Implementation Details

All the experiments were performed using the SVM package *LIBSVM*⁶ customized to embed our own kernel. For the LLL challenge submission, we optimized the regularization parameter C by 10-fold cross validation; while we used its default value for the AImed experiment. In both experiments, we set the cost-factor W_i to be the ratio between the number of negative and positive examples.

5.2 Results on AImed

K_{SL} performance was first evaluated on the AImed data set (Section 4.1). We first give an evaluation of the kernel combination and then we compare our results with the Subsequence Kernel for Relation Extraction (ERK) described in (Bunescu and Mooney, 2005b). All experiments are conducted using 10-fold cross validation on the same data splitting used in (Bunescu et al., 2005; Bunescu and Mooney, 2005b).

Table 1 shows the performance of the three kernels defined in Section 3 for *protein-protein* interactions using the two evaluation methodologies described above.

We report in Figure 4 the precision-recall curves of ERK and K_{SL} using OARD evaluation methodology (the evaluation performed by Bunescu and Mooney (2005b)). As in (Bunescu et al., 2005; Bunescu and Mooney, 2005b), the graph points are obtained by varying the threshold on the classifi-

⁶<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

OAOB			
Kernel	Precision	Recall	F_1
K_{GC}	57.7	60.1	58.9
K_{LC}	37.3	56.3	44.9
K_{SL}	60.9	57.2	59.0
OARD			
Kernel	Precision	Recall	F_1
K_{GC}	58.9	66.2	62.2
K_{LC}	44.8	67.8	54.0
K_{SL}	64.5	63.2	63.9
ERK	65.0	46.4	54.2

Table 1: Performance on the AImed data set using the two evaluation methodologies, OAOB and OARD.

cation confidence⁷. The results clearly show that K_{SL} outperforms ERK, especially in term of recall (see Table 1).

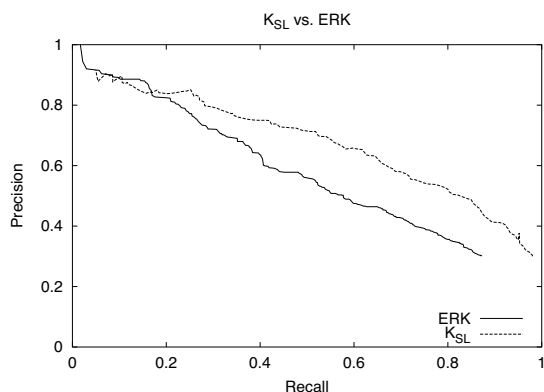


Figure 4: Precision-recall curves on the AImed data set using OARD evaluation methodology.

Finally, Figure 5 shows the learning curve of the combined kernel K_{SL} using the OARD evaluation methodology. The curve reaches a plateau with around 100 Medline abstracts.

5.3 Results on LLL challenge

The system was evaluated on the “basic” version of the LLL challenge data set (Section 4.2).

Table 2 shows the results of K_{SL} returned by the scoring service⁸ for the three subsets of the training set (with and without coreferences, and with their union). Table 3 shows the best results obtained at the official competition performed in April 2005. Comparing the results we see that K_{SL} trained on each subset outperforms the best

⁷For this purpose the probability estimate output of *LIB-SVM* is used.

⁸<http://genome.jouy.inra.fr/texte/LLLchallenge/scoringService.php>

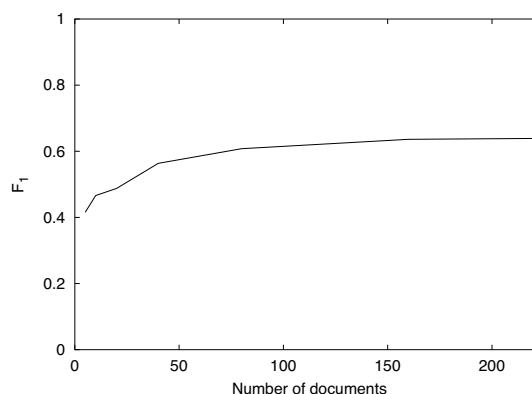


Figure 5: K_{SL} learning curve on the AImed data set using OARD evaluation methodology.

Coref.	Precision	Recall	F_1
all	56.0	61.4	58.6
with	29.0	31.0	30.0
without	54.8	62.9	58.6

Table 2: K_{SL} performance on the LLL challenge test set using only the basic linguistic information.

systems of the LLL challenge⁹. Notice that the best results at the challenge were obtained by different groups and exploiting the linguistic “enriched” version of the data set. As observed in (Nedéllec, 2005), the scores obtained using the training set without coreferences and the whole training set are similar.

We also report in Table 4 an analysis of the kernel combination. Given that we are interested here in the contribution of each kernel, we evaluated the experiments by 10-fold cross-validation on the whole training set avoiding the submission process.

5.4 Discussion of Results

The experimental results show that the combined kernel K_{SL} outperforms the basic kernels K_{GC} and K_{LC} on both data sets. In particular, precision significantly increases at the expense of a lower recall. High precision is particularly advantageous when extracting knowledge from large corpora, because it avoids overloading end users with too many false positives.

Although the basic kernels were designed to model complementary aspects of the task (i.e.

⁹After the challenge deadline, Reidel and Klein (2005) achieved a significant improvement, $F_1 = 68.4\%$ (without coreferences) and $F_1 = 64.7\%$ (with and without coreferences).

Test set	Coref.	Precision	Recall	F_1
Enriched	all	55.6	53.0	54.3
	with	29.0	31.0	24.4
	without	60.9	46.2	52.6
Basic	all	n/a	n/a	n/a
	with	14.0	82.7	24.0
	without	50.0	53.8	51.8

Table 3: Best performance on basic and enriched test sets obtained by participants in the official competition at the LLL challenge.

Kernel	Precision	Recall	F_1
K_{GC}	55.1	66.3	60.2
K_{LC}	44.8	60.1	53.8
K_{SL}	62.1	61.3	61.7

Table 4: Comparison of the performance of kernel combination on the LLL challenge using 10-fold cross validation.

presence of the relation and roles of the interacting entities), they perform reasonably well even when considered separately. In particular, K_{GC} achieved good performance on both data sets. This result was not expected on the LLL challenge because this task requires not only to recognize the presence of relationships between entities but also to identify their roles. On the other hand, the outcomes of K_{LC} on the AImed data set show that such kernel helps to identify the presence of relationships as well.

At first glance, it may seem strange that K_{GC} outperforms ERK on AImed, as the latter approach exploits a richer representation: sparse sub-sequences of words, PoS tags, entity and chunk types, or WordNet synsets. However, an approach based on n-grams is sufficient to identify the presence of a relationship. This result sounds less surprising, if we recall that both approaches cast the relation extraction problem as a text categorization task. Approaches to text categorization based on rich linguistic information have obtained less accuracy than the traditional bag-of-words approach (e.g. (Koster and Seutter, 2003)). Shallow linguistics information seems to be more effective to model the local context of the entities.

Finally, we obtained worse results performing dimensionality reduction either based on generic linguistic assumptions (e.g. by removing words from stop lists or with certain PoS tags) or using statistical methods (e.g. *tf.idf* weighting schema). This may be explained by the fact that, in tasks like entity recognition and relation extraction, useful

clues are also provided by high frequency tokens, such as stop words or punctuation marks, and by the relative positions in which they appear.

6 Related Work

First of all, the obvious references for our work are the approaches evaluated on AImed and LLL challenge data sets.

In (Bunescu and Mooney, 2005b), the authors present a generalized subsequence kernel that works with sparse sequences containing combinations of words and PoS tags.

The best results on the LLL challenge were obtained by the group from the University of Edinburgh (Reidel and Klein, 2005), which used Markov Logic, a framework that combines log-linear models and First Order Logic, to create a set of weighted clauses which can classify pairs of gene named entities as genic interactions. These clauses are based on chains of syntactic and semantic relations in the parse or Discourse Representation Structure (DRS) of a sentence, respectively.

Other relevant approaches include those that adopt kernel methods to perform relation extraction. Zelenko et al. (2003) describe a relation extraction algorithm that uses a tree kernel defined over a shallow parse tree representation of sentences. The approach is vulnerable to unrecoverable parsing errors. Culotta and Sorensen (2004) describe a slightly generalized version of this kernel based on dependency trees, in which a bag-of-words kernel is used to compensate for errors in syntactic analysis. A further extension is proposed by Zhao and Grishman (2005). They use composite kernels to integrate information from different syntactic sources (tokenization, sentence parsing, and deep dependency analysis) so that processing errors occurring at one level may be overcome by information from other levels. Bunescu and Mooney (2005a) present an alternative approach which uses information concentrated in the shortest path in the dependency tree between the two entities.

As mentioned in Section 1, another relevant approach is presented in (Roth and Yih, 2002). Classifiers that identify entities and relations among them are first learned from local information in the sentence. This information, along with constraints induced among entity types and relations, is used to perform global probabilistic inference

that accounts for the mutual dependencies among the entities.

All the previous approaches have been evaluated on different data sets so that it is not possible to have a clear idea of which approach is better than the other.

7 Conclusions and Future Work

The good results obtained using only shallow linguistic features provide a higher baseline against which it is possible to measure improvements obtained using methods based on deep linguistic processing. In the near future, we plan to extend our work in several ways.

First, we would like to evaluate the contribution of syntactic information to relation extraction from biomedical literature. With this aim, we will integrate the output of a parser (possibly trained on a domain-specific resource such the Genia Treebank). Second, we plan to test the portability of our model on ACE and MUC data sets. Third, we would like to use a named entity recognizer instead of assuming that entities are already extracted or given by a dictionary. Our long term goal is to populate databases and ontologies by extracting information from large text collections such as Medline.

8 Acknowledgements

We would like to thank Razvan Bunescu for providing detailed information about the Almed data set and the settings of the experiments. Claudio Giuliano and Lorenza Romano have been supported by the ONTOTEXT project, funded by the Autonomous Province of Trento under the FUP-2004 research program.

References

- Razvan Bunescu and Raymond J. Mooney. 2005a. A shortest path dependency kernel for relation extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., October.
- Razvan Bunescu and Raymond J. Mooney. 2005b. Subsequence kernels for relation extraction. In *Proceedings of the 19th Conference on Neural Information Processing Systems*, Vancouver, British Columbia.
- Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155. Special Issue on Summarization and Information Extraction from Medical Documents.
- Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, Spain.
- Alfio Gliozzo, Claudio Giuliano, and Carlo Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan, June.
- Cornelis H. A. Koster and Mark Seutter. 2003. Taming wild phrases. In *Advances in Information Retrieval, 25th European Conference on IR Research (ECIR 2003)*, pages 161–176, Pisa, Italy.
- Alberto Lavelli, Mary Elaine Califf, Fabio Ciravegna, Dayne Freitag, Claudio Giuliano, Nicholas Kushmerick, and Lorenza Romano. 2004. IE evaluation: Criticisms and recommendations. In *Proceedings of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining (ATEM 2004)*, San Jose, California.
- Claire Nedéllec. 2005. Learning language in logic - genic interaction extraction challenge. In *Proceedings of the ICML-2005 Workshop on Learning Language in Logic (LLL05)*, pages 31–37, Bonn, Germany, August.
- Sebastian Reidel and Ewan Klein. 2005. Genic interaction extraction with semantic and syntactic chains. In *Proceedings of the ICML-2005 Workshop on Learning Language in Logic (LLL05)*, pages 69–74, Bonn, Germany, August.
- D. Roth and W. Yih. 2002. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*, Taipei, Taiwan.
- John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.
- Vladimir Vapnik. 1998. *Statistical Learning Theory*. John Wiley and Sons, New York.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for information extraction. *Journal of Machine Learning Research*, 3:1083–1106.
- Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan, June.