



**University of
Zurich**^{UZH}

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2009

Detecting Protein-Protein Interactions in Biomedical Literature Using a Parser

Schneider, Gerold

Posted at the Zurich Open Repository and Archive, University of Zurich
ZORA URL: <https://doi.org/10.5167/uzh-24602>
Book Section

Originally published at:

Schneider, Gerold (2009). Detecting Protein-Protein Interactions in Biomedical Literature Using a Parser. In: Clematide, Simon; Klenner, Manfred; Volk, Martin. Searching Answers. Münster: MV Verlag, 109-118.

Detecting Protein-Protein Interactions in Biomedical Literature Using a Parser

Gerold Schneider

Institute of Computational Linguistics, University of Zurich
<http://www.cl.uzh.ch/gschneid>

Abstract. We describe the task of automatically detecting interactions between proteins in biomedical literature. We use a syntactic parser, a corpus annotated for proteins, and manual decisions as training material.

After automatically parsing the GENIA corpus, which is manually annotated for proteins, all syntactic paths between proteins are extracted. These syntactic paths are manually disambiguated between meaningful paths and irrelevant paths. Meaningful paths are paths that express an interaction between the syntactically connected proteins, irrelevant paths are paths that do not convey any interaction. The resource created by these manual decisions is used in two ways. First, words that appear frequently inside a meaningful path are learnt using simple machine learning. Second, these resources are applied to the task of automatically detecting interactions between proteins in biomedical literature.

Key words: IR, NLP, text mining, parsing, biomedicine

1 Introduction

Scientific articles reporting results of biomedical studies are growing exponentially in number¹. Publicly available literature services such as Pubmed (<http://pubmed.gov>) already contain more than 17 million articles. Even for the expert it has become difficult to keep an overview of new results. Fully or partly automated systems that extract biological knowledge from text have thus become a necessity. Particularly, knowledge about protein-protein interactions (PPI) is needed in biomedical and genetic research, as exemplified by the LLL genic interaction challenge [Nedellec, 2006] and the BioCre-AtIvE challenge PPI track [Krallinger et al., 2008].

A number of methods have been applied to this task. Simple approaches classify two proteins as interacting when mentioned in the same sentence, or when their co-occurrence in an abstract is very frequent [Rebholz-Schuhmann et al., 2006]. Such approaches often yield high recall at low precision and can be used as baselines (see section 4.1) for more involved approaches.

¹ This research is partially funded by the Swiss National Science Foundation (grant 100014-118396/1). Additional support is provided by Novartis Pharma AG, NITAS, Text Mining Services, CH-4002, Basel, Switzerland. For more information on our project, visit our OntoGene project web page at <http://www.ontogene.org>

Other approaches apply handcrafted rules, for example regular expressions for surface searches [Giuliano et al., 2006], or syntactic patterns on automatically parsed corpora [Rinaldi et al., 2006, Fundel et al., 2007]. These approaches typically achieve high precision at the cost of recall.

Third, machine learning methods are increasingly used to construct a model from large annotated sources. To extract meaningful features for the model construction, dependency parsing is often used. [Erkan et al., 2007] extract sentences in which two proteins and an interaction word co-occur. Their features include the interaction words and the parents of the proteins, according to the dependency analysis. [Kim et al., 2008] use a walk kernel which contains fragments of the paths between two proteins. Due to sparse data, the paths were partitioned into patterns, each consisting of two vertices and their intermediate edge (*vertex-walk*), and of two edges and their common vertex (*edge-walk*). While this alleviates the sparse data problem, it neglects that many semantic configurations are not local, they depend on considerably larger tree fragments. We suggest to use a single feature consisting of the entire path, but to reduce sparseness by using very little lexical information and linguistic insights to shorten the paths. Our approach is characterised by using linguistic insights and lightweight resources, allowing us to achieve good results despite using simple statistical methods and learning algorithms.

[Landeghem et al., 2008] extends the approach of [Kim et al., 2008] by using a feature-based approach instead of a kernel, where e.g. each vertex-walk and each edge-walk is a feature, on the one hand a lexical feature containing words, on the other hand a syntactic feature containing tags. The lexical features are quite sparse due to Zipf's law.

The approach that we present in this paper is hybrid. It uses a large, partly annotated resource and manual annotations, and parsed data in order to obtain a suitable level of abstraction and reducing the number of manual annotation decisions, thus creating two new linguistic resources: a knowledge base of syntactic patterns expressing protein-protein relations, and a list of words that are low in semantic content. Our approach achieves higher precision than cooccurrence methods because it uses stricter requirements. It achieves higher recall than handcrafted syntactic patterns because all syntactic connections that are observed in a large corpus are taken into consideration. Machine Learning methods and backoff techniques are applied to the linguistic resource thus created. For training, we have used the GENIA corpus, to which we have manually added interaction information. Our approach shows a new application of the GENIA corpus. For the application phase, we use various corpora, for example the IntAct corpus [Kerrien et al., 2006]. The IntAct corpus was devised for the PPI task but has been underused so far.

The aim of our application is twofold. On the one hand, we use the IntAct data as a gold standard for evaluating the performance of our PPI algorithm, on the other hand we propose an algorithm that may help IntAct annotators by suggesting protein-protein interactions to them.

The paper is structured as follows. In chapter 2, we describe how we collect and annotate the syntactic data. In chapter 3, our application to the IntAct corpus is described. We give an evaluation in chapter 4. In chapter 5, we discuss further applications and describe typical errors.

2 Collection and Annotation of Syntactic Data

2.1 Parsing and Tree Walks

The GENIA corpus has been manually annotated for biomedical terms and proteins. It consists of 2,000 abstracts, containing over 18,000 sentences. We parse the GENIA corpus with a state-of-the-art dependency parser which has been adapted to and evaluated on the biomedical domain [Schneider et al., 2007, Schneider, 2008, Haverinen et al., 2008].

Figure 1 shows the output of the parser for the sentence *Significant amounts of Tom40 were also coprecipitated by anti - Tom20 and anti - Tom22*

After parsing, we collect all syntactic connections that exist between all the terms as follows. For each term-cooccurrence, i.e. two terms appearing in the same sentence, a collector traverses the tree from one term up to the lowest common mother node, and down the second term, recording all intervening nodes. Such traversals have been used in many PPI applications [Kim et al., 2008], they are commonly called tree walks or paths. If one records all the information that an intermediate node contains, for example its lexical items and subnodes, the path would be extremely specific, which leads to sparse data and hence a recall problem for most applications. If one only records the grammatical role labels, the paths are too general, which leads to a precision problem for most applications. As a working assumption, we have recorded the lexical head lemma of the top node, and the grammatical labels plus prepositions connecting all intervening nodes. We have split the path into a left and a right half on either side of the top node. The sentence in figure 1 contains 3 proteins: *Tom40*, *Tom20*, and *Tom22*. The path between *Tom40* and *Tom22* consists of the top node *coprecipitate*, the left path [*subj.modpp-of*] and the right path [*pobj.conj*]. The path is treated as a single feature, unlike in similar approaches, e.g. [Kim et al., 2008]. They use a kernel with fragments of the paths between two proteins. Each pattern consists of two vertices and their intermediate edge (*vertex-walk*), and of two edges and their common vertex (*edge-walk*). While this alleviates the sparse data problem, it neglects that many semantic configurations are not local, they depend on considerably larger tree fragments.

We suggest to use a single feature consisting of the entire path, but using only little lexical information and linguistic insights to shorten the paths. In [Landeghem et al., 2008], each vertex-walk and each edge-walk leads to two features, on the one hand a lexical feature containing words, on the other hand a syntactic feature containing tags. The lexical features are sparse due to Zipf's law. There is a small closed class of lexical items that is crucial to syntax [Collins and Brooks, 1995, Collins, 2003], namely prepositions, which we have thus introduced into the path. But also the syntactic features are potentially sparser than what is linguistically meaningful, as they contain tags. A subject relation, for example, is mostly between a noun and a verb. Since there are 4 noun tags and almost a dozen verb tags, sparseness is inflated. Our paths are also shorter and less sparse than in many other representations, because our syntactic graphs are based on chunks. We present linguistic insights that allow us to further reduce data sparseness by shortening paths in section 3.2.

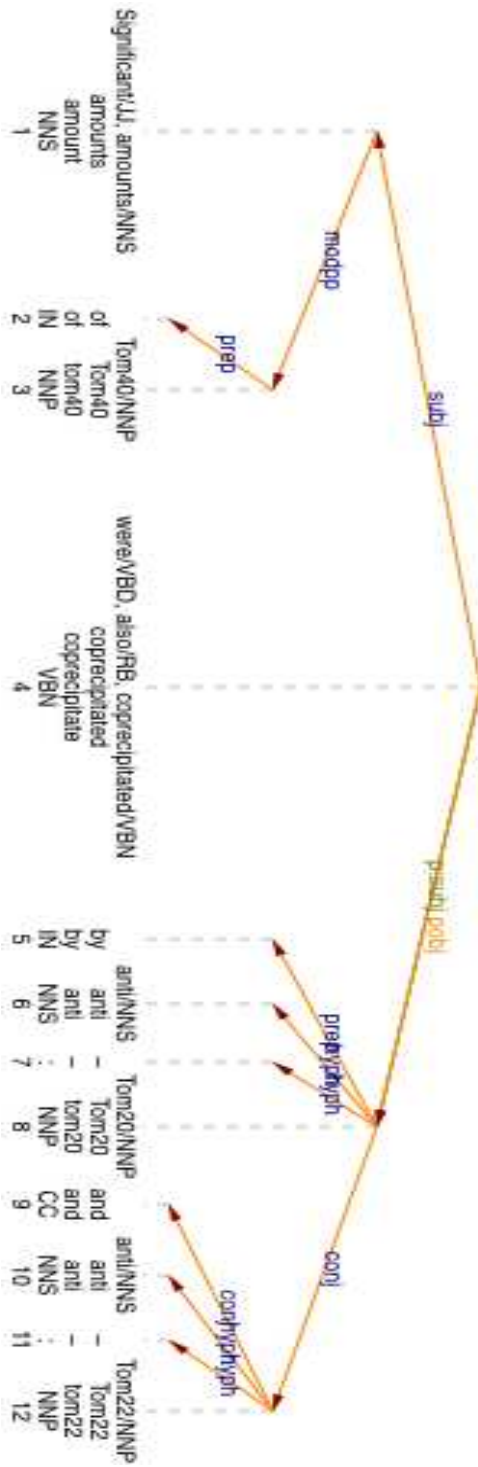


Fig. 1. Dependency parser output example

2.2 Manual Annotation

Only a minority of the paths extracted by the method just introduced actually express a biomedical interaction. The path between *Tom20* and *Tom22* in our example, which consists of the top node *Tom20*, the empty left node, and the right node [*conj*], does not express any biomedical interaction, it does not state that *Tom20* and *Tom22* interact. In order to apply the paths to the PPI task we need to classify paths into those expressing an interaction relation and those that do not. We decided to classify manually.

Ideally, one should classify every individual co-occurrence of two terms in the entire corpus. Since we did not have the resources to conduct such a large-scale, token-based annotation, we have opted for a type-based annotation at the level of the extracted paths. If our working assumption holds that these paths are a useful level of abstraction, our type-wise annotation offers a useful middle-ground in the trade-off between token-wise annotation and unsupervised machine learning. We have discarded singletons, i.e. paths only appearing once in GENIA, since they are too sparse and often arise from parsing errors. The frequency-ranked list of paths tails off sharply, indicating a Zipfian distribution, more than half of all paths are singletons.

A major advantage of annotating a large corpus over formulating hand-written patterns is that no instance is missed (except for very rare ones that happen to be absent from a large corpus). This insight has given rise to the methodology of corpus linguistics in descriptive linguistics.

We manually annotated the about 3000 paths that appear at least twice. Each decision, i.e. whether the target path expresses a relation ('yes') or not ('no'), was based on at least three example sentences containing the target path (except for paths that only appeared twice in GENIA). We annotated 309 paths with 'yes'. During the annotation we observed that there are relatively few paths for which the example sentences suggested diverging decisions. We also observed that many paths express subset relations, for example *A is a B protein*, where *A* is a subset of *B*. We have decided to annotate these cases with a third class in addition to 'yes' and 'no', saving them for future ontology applications. Additionally, we observed that semantically lightweight nouns seem to play an important role for the decision: in order to test if a relation is expressed by an example sentence, it typically helps to paraphrase it, using the top node and the two terms. The sentence *A activates groups of B* essentially expresses that *A activates B*, or *A blocks activation of B* expresses that *A blocks B*, whereas *A activates C, which has a binding site for B* does not express that *A activates B*. There is a large set of words like *group* and *activation*, for which we would like to use the term *transparent words* (*sortals* are a subgroup of them). We compiled lists of them and extended the list with a simple machine-learning approach described in section 2.3.

2.3 Learning Transparent Words from the Type-Based Annotation

Although we collected the paths based only on the head lemma of the top node and the labels of intervening nodes, we also kept record of all intervening words in order to be able to learn specific rules where necessary. All the words intervening inside a path are, for instance, candidates for being transparent words, as introduced in section 2.2. For each word appearing inside a path, we calculate a score which simply divides its

frequency inside a path by its total frequency. Words above a threshold are treated as transparent in the application phase. Depending on the threshold, our transparent words resource contains between 100 and 800 words.

3 Application to IntAct

We chose the IntAct data as a gold standard. The paths that are extracted from GENIA can directly be used for PPI detection. For example, in sentence 2, a pattern with the decision ‘yes’ exists for the relation between *Tim18* and *Tim12*, i.e. the pattern with top node *coimmunoprecipitate*, left path [*subj*] and right path [*pobj*]. Although we have constructed the paths in a way that aims to reduce sparseness, and although we have used a corpus-based annotation method instead of introspective creation of hand-crafted rules, recall is very poor when the patterns are applied directly. The following are the main reasons why recall is low. In each subsection we also discuss how we have improved the situation.

3.1 Term Recognition and Upper Bound

The protein detection and grounding algorithm which we use [Kaljurand et al., 2009a] has a recall of about 72% on the IntAct corpus. Since any interaction involves two proteins, the performance for recognition and grounding of protein pairs can be expected to be about 50% recall. It is beyond the scope of this paper to improve term recognition, so we need to accept this upper bound for the protein-protein interaction task².

3.2 Transparent Words

Sparse data problems can be reduced significantly by applying the transparent words resource that we have created. If no annotated path from GENIA exists, the following sparse data reduction methods are used as backoffs:

- First, proteins occurring inside noun chunks are allowed to replace the head of the chunk if the head is a transparent word (for example, in *the p85 C-terminal SH2 domain*, *p85 C-terminal SH2* replaces the head because *domain* is a transparent word).
- Secondly (if still no path from GENIA exists), the relations for appositions, conjunctions and hyphens are cut (see example below).
- Third (if still no path from GENIA exists), parts of trees that are headed by an transparent word are cut (see example below).

We have discussed that in example sentence 2 a pattern directly reports the relation between *Tim18* and *Tim12*, but the others are missed. Cutting conjunctions (second backoff) has the effect that *portion of Tim54* now appears at the same level as *Tim12*, cutting the transparent words *portion* and *all* (third backoff) means that *Tim54* appears at the same level as *Tim12*. Now, the pattern that reported the relation between *Tim18* and *Tim12* without backoff, also finds the relation between *Tim18* and *Tim54*.

² baseline 1 in the evaluation section calculates the exact upper bound

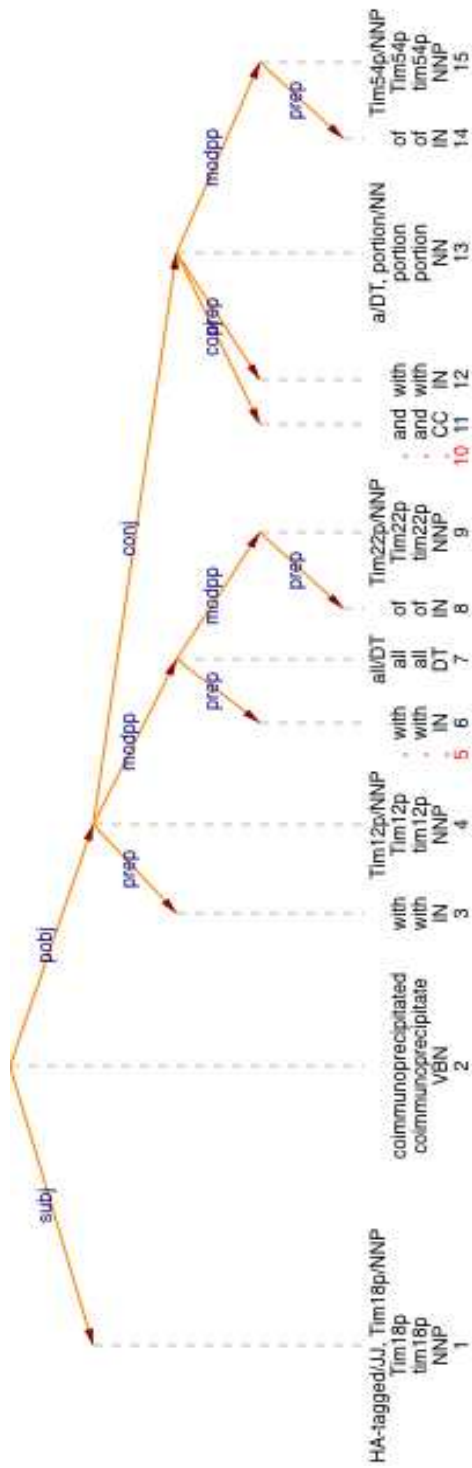


Fig. 2. Dependency parser output example

3.3 Surface Patterns to Address Tagging and Parsing Errors

Tagging and parsing errors are frequent, despite using taggers and parsers that are adapted to the domain. Errors are an important reason for the remaining sparseness. We have therefore developed surface patterns. They apply only at a backoff level, i.e. if no syntactic path is found in GENIA after all syntactic backoffs.

There are three patterns. Each pattern consists of two proteins and a keyword:

- A verb B, e.g. *A interacts-with B*
- noun A B, e.g. *association between A and B*
- A B noun, e.g. *A - B binding*

The distance between *A* and the keyword, as well as the distance between *B* and the keyword are restricted, as is typical for observation window-based approaches. These surface patterns typically achieve relatively good precision, but insufficient recall. If observation windows are large, recall increases but precision drops off.

4 Evaluation

We have evaluated our approach and established an upper bound and a number of baselines, in order to measure the relative success of our approach. We have used the first 1000 sentences of the IntAct data for the evaluation. We have mentioned that, given the performance of our term recognition and grounding tool, the upper bound is about 50% recall. The term grounding tool sometimes delivers one UniProt ID and sometimes several UniProt IDs, on average 2.02 IDs. Since the ultimate aim of our approach is to deliver one and exactly one ID, we speak of exact precision and recall if there is an interaction and if both proteins are given only one ID. We speak of loose precision and recall if there is an interaction and if one or both of the proteins are given one or several IDs by the grounding algorithm (i.e. if the UniProt ID could not be fully disambiguated), if one of the delivered IDs is correct for each protein. UniProt IDs are fine-grained, including the organism in which the protein functions (ortholog). Since the task described in this paper is interaction detection rather than full term grounding disambiguation, we will mainly report loose precision and recall figures.

Our currently best system achieves 80.5% loose precision and 21.0% loose recall, and 59% exact precision at 15% exact recall. In order to assess the relative success that these performance figures mean, we will now compare them to a number of increasingly more advanced baselines.

4.1 Baselines

Baseline 1 Cooccurrence of two proteins in a sentence is a low baseline, one that heavily overgenerates. Precision of this baseline tells one how much one gets for free, while recall tells one how good one can maximally get (upper bound). We achieve 39.2% loose precision and 41.2% loose recall. Compared to this baseline, our best system has more than doubled precision at the cost of losing about half of the recall.

Baseline 2 In a purely syntactic approach, measuring all syntactically connected proteins lead to a second baseline. Since our parser does not always deliver an analysis spanning the entire sentence, especially when sentences are complex, this is not a variant of baseline 1. We achieve 50.1% loose precision and 29.8% loose recall with baseline 2.

Baseline 3 In a purely “non-syntactic” surface-based approach, observation windows are often used. We apply the surface patterns introduced in section 3.3, but no paths, and do not use information on transparent words. The window size is 5 words, which means that maximally 3 words may occur between the head (e.g. the verb) and the term. We achieve 78.6% loose precision and 11.4% loose recall with baseline 3.

Baseline 4 We extend baseline 3 by using the transparent words resource that we have created. This baseline is still purely surface-based and window size is 5, but transparent words are cut from the observation window, which means that remote words may move into the observation window if they are mainly separated by transparent words. We achieve 81.8% loose precision and 18.7% loose recall with baseline 4.

Best system The currently best system achieves 80.5% precision and 21.0% loose recall. It uses syntactic patterns, surface patterns and the transparent words resource at both levels.

Table 1. Baselines compared to the system

Method	Description	Loose Precision	Loose Recall
Baseline 1	sentence cooccurrence	39.2%	41.2%
Baseline 2	syntactically connected	50.1%	29.8%
Baseline 3	surface, no transparent words	78.6%	11.4%
Baseline 4	surface, transparent words	81.8%	18.7%
Best system	syntax, surfaces, transparent words	80.5%	21.0%

The step-wise improvements from the baselines to the currently best system are summarised in table 1. The performance of the last baseline, surface-based but using transparent words, is impressive. Adding the transparent words resource to the system increased performance more than the syntactic filter which the best system uses. The best system achieves 51% of the upper bound recall in baseline 1. Given gold-standard term information, the system therefore, all other parameters being equal, achieves a performance of 80.5% precision and 51% recall on the PPI detection task, which amounts to an F-score of 62.4%.

4.2 Breakdown of Results

We have broken down the precision results in table 2, which also quantifies the backoff method we use. If a syntactic method gives a decision, it is used, otherwise the *same chunk* method is applied. If that does not give a decision, the surface patterns are used. When available, the syntax-based method delivers the highest precision, but the surface method with the transparent words resource performs almost equally well. Absolute numbers are given in the third column.

Table 2. Breakdown of results

Backoff used	Loose Precision Percent	Loose Precision Count
Syntax	83.8%	62/74
Same chunk	75%	3/4
surface A verb B	76%	38/50
surface noun A B	82.4%	14/17
surface A B noun	66.7%	2/3
TOTAL	80.5%	120/149

5 Portability of Resources

Given our success with this method, relatively good precision at acceptable recall on IntAct, we are applying our approach to various corpora. We give a brief overview of our current activities in this chapter.

5.1 BioNLP Shared Task

We participated in the BioNLP shared task [Kaljurand et al., 2009b], where we achieved a recall of 26% and a precision of 44% in the official test run, which placed us at rank 8 of 24 teams. The task was difficult due to the fact that events needed to be labelled and complex events (e.g. interactions between interactions) were included. Our performance on non-complex events was 57% precision at 40% recall, which is comparable to the results achieved with IntAct. We included more term resources in order to increase recall, at a certain cost for precision. This is beneficial for the envisaged annotator tool. It turned out that transparent words were valuable indicators for the event type label, so that we could not use the transparent words resource to reduce sparseness.

5.2 BioCreative Shared Task

The BioCreative Shared Task uses unlabelled events, the transparent words resource was beneficial. We added further term resources to boost recall, and extended the back-off chain, including WordNet and training data from the BioNLP shared task. At the time of writing, our performance on the evaluation set is not yet known, but we achieved 32% recall at 10% precision on the development set. Since complete term disambiguation is required, this figure needs to be compared to exact precision and recall on IntAct ($p=59%$, $r=15%$). Where term grounding is correct, about 60% of the interactions are found. The ranking of the returned interactions plays a crucial role in this shared task, which entails that a high recall approach is vital. Precision is also quite low because the task includes the difficult distinction between novel and background interactions: only interactions that are mentioned in the target article for the first time are relevant.

Close manual inspection of the test data showed us that most of the missed interactions cannot be found without several logical conclusions, or they involve several sentences, which means that they are beyond the scope of our approach.

6 Conclusions

We have created three new resources: annotated paths from the GENIA corpus, automatically learnt transparent words, and transparent words noted while annotating and

testing. We have applied the resources to IntAct, both as a PPI task, and in order to develop an algorithm helping annotators.

We have evaluated our algorithm and performed better than all baselines. On the PPI task, our best system achieves 80.5% loose precision and 21% loose recall. 21% loose recall corresponds to 51% of the upper bound, if gold standard term recognition were used. We have based our path representations on linguistic insights. We use syntactic paths as features with very little lexical information (only the top node word and prepositions in PPs), and based on chunks, both of which lead to fewer sparse data problems. We have shown that transparent words, words with low semantic content, play an important role in allowing us to further reduce sparseness: we have cut transparent words and their nodes from our path representations. Finally we have presented recent applications.

7 Acknowledgements

Much of the research presented here would have been impossible without the close co-operation in our OntoGene team: Kaarel Kaljurand, Fabio Rinaldi, Thomas Kappeler, Martin Romacker, Therese Vachon, Jean-Marc von Allmen. The support from the Computational Linguistics group at our University has been equally important. Many colleagues have participated directly in parts of the project, reviewed papers and grant proposals, and gave us valuable feedback. I would particularly like to thank Simon Clematide, Manfred Klenner, Tobias Kuhn, Martin Volk, and Michael Hess.

Without the generous support of Michael Hess, who supervised my doctoral thesis, wrote research grants and hired me in various functions, I would not be in the position that I am now. Thank you so much, Michael, and congratulations!

References

- [Collins, 2003] Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29:589 – 637.
- [Collins and Brooks, 1995] Collins, M. and Brooks, J. (1995). Prepositional attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA.
- [Erkan et al., 2007] Erkan, G., Ozgur, A., and Radev, D. R. (2007). Extracting interacting protein pairs and evidence sentences by using dependency parsing and machine learning techniques. In *Proceedings of BioCreAtIvE 2*.
- [Fundel et al., 2007] Fundel, K., Küffner, R., and Zimmer, R. (2007). RelEx – relation extraction extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.
- [Giuliano et al., 2006] Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of EACL 2006*.
- [Haverinen et al., 2008] Haverinen, K., Ginter, F., Pyysalo, S., and Salakoski, T. (2008). Accurate conversion of dependency parses: targeting the Stanford scheme. In *Proceedings of Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, Turku, Finland.
- [Kaljurand et al., 2009a] Kaljurand, K., Rinaldi, F., Kappeler, T., and Schneider, G. (2009a). Detecting and grounding terms in biomedical literature. In *CICLing 2009, 10th International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico.

- [Kaljurand et al., 2009b] Kaljurand, K., Schneider, G., and Rinaldi, F. (2009b). UZurich in the BioNLP 2009 shared task. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 28–36, Boulder, Colorado. Association for Computational Linguistics.
- [Kerrien et al., 2006] Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Liefink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roehert, B., Thornycroft, D., Zhang, Y., Apweiler, R., and Hermjakob, H. (2006). Intact: open source resource for molecular interaction data. *Nucleic Acids Res.*, (35 Database):D561–D565.
- [Kim et al., 2008] Kim, S., Yoon, J., and Yang, J. (2008). Kernel approaches for genic interaction extraction. *Bioinformatics*, 9:10.
- [Krallinger et al., 2008] Krallinger, M., Leitner, F., Rodriguez-Penagos, C., and Valencia, A. (2008). Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome Biology*, 9(Suppl 2).
- [Landeghem et al., 2008] Landeghem, S. V., Saeys, Y., and de Peer, Y. V. (2008). Extracting protein-protein interactions from text using rich feature vectors and feature selection. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, Turku, Finland.
- [Nedellec, 2006] Nedellec, C. (2006). Learning language in logic – genic interaction extraction challenge. In *Proceedings of LLL '05*, pages 31–37.
- [Rebholz-Schuhmann et al., 2006] Rebholz-Schuhmann, D., H.Kirsch, Arregui, M., Gaudan, S., Riethoven, M., and P.Stoehr (2006). EBIMed – text crunching to gather facts for proteins from Medline. *Bioinformatics*, 23(2):e237 – e244.
- [Rinaldi et al., 2006] Rinaldi, F., Schneider, G., Kaljurand, K., Hess, M., and Romacker, M. (2006). An environment for relation mining over richly annotated corpora: the case of GENIA. *BMC Bioinformatics*, 7(Suppl 3):S3.
- [Schneider, 2008] Schneider, G. (2008). *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis, Institute of Computational Linguistics, University of Zurich.
- [Schneider et al., 2007] Schneider, G., Kaljurand, K., Rinaldi, F., and Kuhn, T. (2007). Pro3Gres parser in the CoNLL domain adaptation shared task. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1161–1165, Prague.