

Extracting Protein-Protein Interactions with Language Modelling

Ali Reza Ebadat

INRIA-INSA

ali_reza.ebadat@inria.fr

Abstract

In this paper, we model the corpus-based relation extraction task, namely protein-protein interaction, as a classification problem. In that framework, we first show that standard machine learning systems exploiting representations simply based on shallow linguistic information can rival state-of-the-art systems that rely on deep linguistic analysis. We also show that it is possible to obtain even more effective systems, still using these easy and reliable pieces of information, if the specifics of the extraction task and the data are taken into account. Our original method combining lazy learning and language modelling out-performs the existing systems when evaluated on the LLL2005 protein-protein interaction extraction task data¹.

1 Introduction

Since the nineties, a lot of research work has been dedicated to the problem of corpus-based knowledge acquisition, whether the aimed knowledge is terminology, special cases of vocabulary (e.g. named entities), lexical relations between words or more functional ones. This paper focuses on this last kind of acquisition, i.e., relation extraction, and more specifically on Protein-Protein Interaction (PPI) extraction from bio-medical texts. The goal of PPI is to find pairs of proteins within sentences such that one protein is described as regulating, inhibiting, or binding the other. In functional genomics, these interactions, which are not available in structured database but scattered in scientific papers, are central to determine the function of the genes.

In order to extract PPIs, the texts which contain the interactions have to be analyzed. Two kinds of linguistic analysis can be performed for this purpose: deep and shallow. Automatic deep analysis,

¹This work was achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

which provides a syntactic or semantic parsing of each sentence, can be a useful source of information. However, tools for automatic deep analysis are available only for a limited number of natural languages, and produce imperfect results. Manual deep analysis, on the other hand, is time consuming and expensive. Another way to analyze texts is to rely only on a shallow linguistic analysis, taking into account the sole words, lemmas or parts of speech (POS) tags. Automatic tools for shallow analysis are available for many languages, and are (sufficiently) reliable.

In this paper, we advocate the use of shallow linguistic features for relation extraction tasks. First, we show that these easy and reliable pieces of information can be efficiently used as features in a machine learning (ML) framework, resulting in good PPI extraction systems, as effective as many systems relying on deep linguistic analysis. Furthering this idea, we propose a new simple yet original system, called LM-kNN and based on language modeling, that out-performs the state-of-the-art systems.

The paper is organized as follows. Section 2 reviews related work on PPI extraction from bio-medical texts. Section 3 specifies the problem and our methodology. Results when using classical machine learning algorithms are given in Section 4, together with a comparison with existing systems. The last section presents a conclusion and some future work.

2 Related Work

In this literature review, focus is set on researches dedicated to relation extraction from bio-medical texts, especially those evaluated in a PPI framework. The systems proposed for this task can be organized into different groups, depending on the source of knowledge (deep vs. shallow linguistic information) and on the approach used (manual vs. ML).

For instance, RelEx (Fundel et al., 2007) exploits manually built extraction rules handling deep and shallow linguistic information. This system yields good results, yet using such an hand-elaborated knowledge is a bottleneck requiring expertise for any new domain. Thus, many ML-based approaches were proposed to overcome this limitation. The ML techniques range from SVM with complex kernels (Airoola et al., 2008; Kim et al., 2010) or CRF (?), to expressive techniques like inductive logic programming (Phuong et al., 2003). Lexical or linguistic features of words surrounding a pair of proteins can be considered as shallow linguistic features to train the systems (Bunescu and Mooney, 2006; ?; Sun et al., 2007). Yet, most of the techniques rely on deep linguistic analysis like syntactic parsing. Indeed, grammatical relations are assumed to be important for PPI extraction, especially when few training data compared to test data are available (Fayruzov et al., 2009). Yet, the performance of extraction systems being sensitive to the accuracy of automatic parsers (Fayruzov et al., 2008), shallow linguistic information still remains an option (Xiao et al., 2005), though up-to-now less effective than deep one.

In this work, we defend the hypothesis that shallow linguistic information combined with standard ML approaches is sufficient to reach good results. Furthermore, we propose a system demonstrating that when this simple information is cleverly used, it even out-performs these state-of-the-art systems.

3 Approach

This section is dedicated to the different machine learning approaches, based on shallow linguistic features, that we experimented. The two first subsections respectively present how to model the PPI task as a machine learning problem—and in particular how relations are described—and the classification tools commonly used for similar tasks. In the last subsection, we propose a new relation extraction technique, based on language modelling, which is expected to be more efficient than the existing ones.

3.1 Modelling the Relation Extraction Task as a Machine Learning Problem

The goal of relation extraction is to predict, at the occurrence level, if two entities share a defined relation. Expert systems, with manually defined ex-

traction patterns, are usually very costly to build, cannot be adapted to new domains and require an expert knowledge both for the pattern design and the domain which is rarely available. Thus, it is usual to try to build relation extraction systems by machine learning. Such approaches require examples of the spotted relations, but the necessary expert knowledge is cheaper in this case than for pattern design. Moreover, bootstrapping and iterative approaches (Hearst, 1992) or active learning can be used to lower this cost.

In PPI extraction, the goal is to predict if there is any interaction between two proteins. In such a case, the relation is directed, that is, one of the entity is an agent and the other is the target. For example, in the sentence reported in Figure 1 in which entities (proteins) are in bold, there is a relation between *GerE* and *cotD* for which *GerE* is the agent and *cotD* is the target.

GerE stimulates cotD transcription and inhibits cotA transcription in vitro by sigma K RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (sigK) that encode sigma K.

Figure 1: Sample sentence for protein-protein interaction

To handle this directed relation problem, we model it as a 3-class machine learning task. For each training sentence, each pair of entities is either tagged as *None* if the entity pair does not have any interaction, *LTR* if the interaction is from the left to the right (agent to target in the sentence word order), and *RTL* if the interaction is from the right to the left.

The representation, that is, the set of features describing our examples for the machine learning algorithms is voluntarily chosen as very simple. Indeed, a relation is simply represented by the bag of lemmas occurring between the two entities. Grammatical words are kept since they may be important clues to detect the direction of the interaction (like the word *by*). For instance, Table 1 reports the examples found in the sentence: *Most cot genes, and the gerE, are transcribed by sigma K RNA polymerase*. More formally, each example is described by a vector; each dimension of this vector corresponds to a lemma and its value is 1 if the word occurs between the entities and 0 otherwise. The sparse vector obtained is expected to be

a representation both performant and robust since it does not rely on any complex pre-processing.

Example pair	Bag of lemmas	Class
cot,gerE	gene,and,the	None
cot,sigmaK	gene,and,the,gerE, gene,be,transcribe,by	RTL
gerE,sigmaK	gene,be,transcribe,by	RTL

Table 1: Examples of bag of lemmas to be used as feature vector

3.2 Machine Learning for the Bag of Lemmas Representation

In the experiments reported below, this bag-of-lemmas representation is exploited with machine learning techniques popularly used for similar tasks: Support Vector Machine (SVM), Random Tree and Random Forest (as implemented in the Weka toolkit (Hall et al., 2009)).

SVM aims at constructing a set of hyperplanes in the representation space dividing the examples according to their class. When used with complex kernels, the hyperplanes are searched in a higher space, resulting in a complex separation in the original representation space. Random Tree and Random Forest (Breiman, 2001) are two classification algorithms based on the well-known decision trees offering a better robustness especially when tackling problems with small or noisy training data. Random Tree constructs a classical decision tree but considers only a subset of attributes (features) that are randomly selected at each node. Random Forest extends this technique: it builds a large set of decision trees by randomly sampling the training data and the features. It is important to note that all these techniques learn explicitly or implicitly to divide the representation space—in our case the lemma vector space—into different parts corresponding to our 3 classes.

3.3 Nearest Neighbors with Language Modelling

Besides these somewhat classical machine learning approaches, we propose a new technique to extract relations. As the previous ones, it still uses shallow linguistic information, which is easy to obtain and ensures the necessary robustness. One of the main differences with the previous approaches concerns the representation of the examples: it takes into account the sequential aspect of

the task with the help of n-gram language models. Thus, a relation is represented by the sequence of lemmas occurring between the agent and the target, if the agent occurs before the target, or between the target and the agent otherwise. A language model is built for each example Ex , that is, the probabilities based on the occurrences of n-grams in Ex are computed; this language model is written \mathcal{M}_{Ex} . The class (LTR, RTL or none) of each example is also memorized.

Given a relation candidate (that is, two proteins or genes in a sentence), it is possible to evaluate its proximity with any example, or more precisely the probability that this example has generated the candidate. Let us note $C = \langle w_1, w_2, \dots, w_m \rangle$ the sequence of lemmas between the proteins. For n-grams of n lemmas, this probability is classically computed as:

$$P(C|\mathcal{M}_{Ex}) = \prod_{i=1}^m P(w_i|w_{i-n}..w_{i-1}, \mathcal{M}_{Ex})$$

As for any language model in practice, probabilities are smoothed in order to prevent unseen n-grams to yield 0 for the whole sequence. In the experiments reported below, we consider bigrams of lemmas and simply use interpolation with lower order n-grams (unigram in this case) combined with an absolute discounting (Ney et al., 1994).

In order to prevent examples with long sequences to be favored, the probability of generating the example from the candidate ($P(Ex|\mathcal{M}_C)$) is also taken into account. Finally, the similarity between an example and a candidate is

$$sim(Ex, C) = \min(P(Ex|\mathcal{M}_C), P(C|\mathcal{M}_{Ex})) .$$

The class is finally attributed to the candidate by a k-nearest neighbor algorithm: the 10 most similar examples (highest sim) are calculated and a majority vote is performed. This lazy-learning technique is expected to be more suited to this kind of tasks than the model-based ones proposed in the previous sub-section since it better takes into account the variety of ways to express a relation (see Section 4.3 for a discussion on this issue).

4 Experiments

In this section, the experiments with the different relation extraction systems described above are presented. The data used and the evaluation metrics and methodologies are first detailed. Then

the results obtained through cross-validation and on held-out test data are given and compared with existing systems. Finally, some insights raised by these results are given.

4.1 LLL Data

To evaluate the different relation extraction systems, we use the data developed for the Learning Language in Logic 2005 (LLL05) shared task (Nédellec, 2005). The goal of LLL05 was to extract protein/gene interactions in abstracts from the Medline bibliography database.

The provided training set is composed of sentences in which a total of 161 interactions between genes/proteins are identified. Since only positive examples (RTL or LTR in our case) are provided in the training data, we need to consider negative examples for training. As explained before, all interactions are directed; thus, each pair of proteins within a sentence having no interaction between its constituents is considered as a negative example. The test set is composed of another set of sentences for which the groundtruth is kept unknown; the results are computed by submitting the predictions to a web service. The original LLL challenge offered the possibility to train and test the systems only on interactions expressed without the help of co-references (mostly with pronouns designating a previously mentioned entity). Also, the training and test data were also provided with or without manual syntactic annotations of the sentences (dependency analysis). Of course, in order to evaluate our systems in a realistic way, we used the data containing interactions expressed with or without co-references, and we did not consider the manual syntactic annotation.

4.2 Evaluation

The evaluation metrics chosen in our experiments are those classically used in this domain: precision, recall and f-measure. It is important to note that in this evaluation, partially correct answers, like an interaction between two entities correctly detected but with the wrong interaction direction, are considered as wrong answers.

We evaluate our LM approach and compare it with the more traditional machine learning techniques and the state-of-the-art systems in two ways. First, we classically use cross-validation. Yet, with so few examples, it is important to choose a number of folds important enough to provide reliable figures; in the results presented be-

low, 30-fold cross-validation is considered. The second way is by using an unseen test dataset. This dataset was developed for the evaluation of the LLL challenge. The groundtruth is kept unknown; and the results are computed by submitting the predictions to a web service.

The differences between these two evaluation procedures shed light on inherent difficulties and biases in some studies that we discuss after presenting our results.

4.2.1 Cross Validation Evaluation

Table 2 reports the recall (R), precision (P) and f-measure (F) computed by 30-fold cross-validation on the different machine learning techniques presented in the previous section. More precisely, the SVM used is the popular libSVM implementation (Chang and Lin, 2001), which was tested with usual kernels (linear and RBF); Random Forest was used with 700 trees, and Naive Bayes and Random tree were used with their default parameters in Weka if any.

Algorithm	P	R	F
libSVM linear kernel	77.1	77.4	77.2
libSVM RBF kernel ($\gamma = 0.1$)	40.7	63.8	49.7
libSVM RBF kernel ($\gamma = 0.5$)	81.4	74.9	78
Random Forest	80.4	80.6	80.4
Random Tree	77.6	77.4	77.5
Naive Bayes	75.1	68.1	69.3
Naive Bayes Multinomial	70.4	70.3	70.3
LM-kNN	82.2	80.3	81.2

Table 2: Performance of shallow linguistic based techniques with 30-fold cross validation

It is interesting to note that all the techniques perform very well, achieving very high scores, except for the SVM with a RBF kernel and $\gamma = 0.1$. This negative result can be explained by the fact that the SVM with such settings and so few training data has a tendency to over-fit, especially because of the training data amount. Apart from this problem, the closeness of the other results tends to show that, for the same bag-of-lemmas representation, the choice of the classifier does not strongly impact on the performance. Yet, overall, Random Forest, SVM with adequate settings and our LM-kNN technique show the highest f-measures.

4.2.2 Held Out Data Evaluation

The held out test data provided for the LLL challenge allows us to evaluate the previous techniques in another evaluation framework. Table 3 reports the performance obtained by these techniques on the complete test set (interaction expressed with or without co-references). For comparison purposes, the results on this dataset reported by other studies are also included. Since many teams have only considered the evaluation without coreferences, which is supposed to correspond to an easier task, we also report the results of our LM-kNN approach and other state-of-the-art systems in this context in Table 4. The first part of each table concerns systems using raw data (no manual annotation), which corresponds to a realistic evaluation of the systems, and the second part contains results of other systems using the provided manual syntactic analysis.

System	P	R	F
systems on raw data			
Goadrich et al. (2005)	25.0	81.4	38.2
Random Forest	57.9	48.1	52.6
libSVM linear kernel	58.0	56.6	57.3
LM-kNN	70.9	79.5	75
systems on manually annotated data			
Katrenko et al. (2005)	51.8	16.8	25.4
Goadrich et al. (2005)	14.0	93.1	24.4

Table 3: Results for held-out test set of LLL, with or without co-references

System	P	R	F
systems on raw data			
Hakenberg et al. (2005)	50.0	53.8	51.8
Greenwood et al. (2005)	10.6	98.1	19.1
Kim et al. (2010)	68.5	68.5	68.5
Fundel et al. (2007)	68	78	72
LM-kNN	67.1	87	75.8
systems on manually annotated data			
Popelínský and Blaťák (2005)	37.9	55.5	45.1
Riedel and Klein (2005)	60.9	46.2	52.6
Kim et al. (2010)	79.3	85.1	82.1

Table 4: Results for held-out test set of LLL, without co-references

The first thing one can note from Table 3 is that the results are lower than those obtained by

cross-validation. This loss is particularly important for the classical approaches based on a bag-of-lemmas representation. This point is not specific to our approaches and was already noticed by previous studies using the LLL dataset. It is due in part to a difference between the way the training and the test sets were built: the distributions of positive examples and negative ones are very different in these two sets since the test data contains much more sentences without any valid interaction. With respect to this, our LM-kNN approach over-performs the other ones and still produces high results for this task.

Besides our LM-kNN technique which ranks first (+6.5% over the best known results for fully automatic systems), it is interesting to note that our other machine learning approaches also perform well compared with state-of-the-art techniques, even though the latter could be considered as more complex than our methods. Indeed, Hakenberg et al. (2005) used finite state automata to generate extraction patterns. In addition to LLL corpora, these authors took advantage of 256 additional positive examples manually annotated. The method of Greenwood et al. (2005) generates candidate patterns from examples with the help of MiniPar for a syntactic analysis and WordNet and PASBio for a semantic analysis and tagging. Goadrich et al. (2005) applied Inductive Logic Programming and Markov Logic methods. The approach used by Kim et al. (2010), as we explained in Section 2, relies on the shortest path in the syntactic parse tree and a specially developed kernel for SVM.

Results of systems tested with manual syntactic information are also worth noting. Katrenko et al. (2005) used the manual syntactic annotations and a ad’hoc ontology to induce extraction patterns. Popelínský and Blaťák (2005) also applied Inductive Logic Programming on the manual syntactic annotation and enriched the data by using WordNet. It is interesting to note that, even with this manual syntactic analysis and the fact that some systems carried tests only on the easiest part of the test set, most of these systems (the case of Kim et al. (2010) is discussed below) perform worse than our simple machine learning approaches.

4.3 Discussion

With the development and the availability of powerful machine learning systems, many NLP problems are now modelled as classification tasks. As

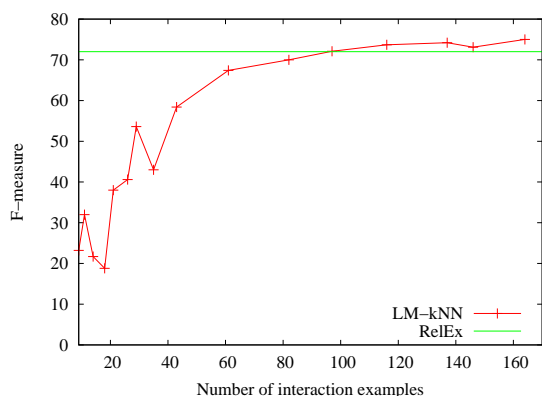


Figure 2: F-measure according to the number of interaction examples

with our Random Forest or SVM experiments, such approaches usually yield good results. Yet, when taking into account the specifics of the task and the data, a huge improvement can be expected.

As the performance of our LM-kNN approach suggests it, lazy learning approaches combined with simple tools like language modelling can offer an interesting alternative to complex tools, especially when dealing with small dataset and a complex classification task.

Another advantage of using a lazy-learning approach such as LM-kNN is that it may offer more robustness than model-based learning approaches when dealing with few examples. And if one wants to reduce the cost of the development of a relation extraction system, it is interesting to see how few examples are necessary to yield good enough results. Figure 2 shows the evolution of f-measure of our LM-kNN system on the LLL test set according to the number of interaction given as examples. For comparison, we also report the result of the rule-based system RelEx (Fundel et al., 2007), which was up to now the best performing system for this task (on raw data, but only tested on interaction without co-references). The evolution of the LM-kNN performance describes an expected curve: important variations are noticed when dealing with very few examples, the improvement is more important when adding examples to a small set of examples, and then the improvement is getting smaller; yet it is interesting to note that the curve suggests that more examples could still improve the f-measure of the system. The performance of RelEx is reached by our technique with less than 100 examples. Therefore, it suggests that instead of hand-crafting complex extraction rules that cannot be adapted to another

extraction task, annotating only 100 examples is enough, which corresponds to about 50 sentences.

Systems using syntax for relation extraction obtain promising results; yet, as we pointed it out before, they are highly dependent on the availability and the quality of the syntactic analysis (see (Fayruzov et al., 2008)). For instance, the f-measure of Kim et al. (2010) declines by 15% when moving from a manual, perfect syntactic annotation to an automatic one.

5 Conclusion

In this paper we have presented and experimented several systems, that can be easily implemented, to extract directed Protein-Protein Interactions in bio-medical texts. We have shown that modeling the PPI extraction task as a classification problem and simply using shallow linguistic information is sufficient to reach good results. Moreover, we have proposed a simple yet very efficient relation extraction system, LM-kNN, based on language modeling which better takes the specifics of the task and data into account. The results, evaluated on a publicly available dataset, underlined the interest of using shallow linguistic information and our new LM-kNN method yielded the best known results.

This good result is very promising, and many perspectives are foreseen. From a technical point of view, it is possible to integrate these machine learning frameworks into an iterative process: newly retrieved relations are used as additional examples to re-train a system. Such approaches, like the one of (Hearst, 1992), as well as active learning techniques are of course straightforward for our lazy-learning approach. From an applicative point of view, our LM-kNN has to be tested over other relation extraction tasks. In particular, we foresee its use for the detection of relations in speech transcripts of sporting events. As it was previously said, shallow linguistic approaches is a necessity in such a context in which the oral characteristics and the speech-to-text process prevent the use of any deep linguistic analysis tools.

References

- Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. A graph kernel for protein-protein interaction extraction. In *Proc. of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 1–9, Columbus, Ohio, USA.
- Christian Blaschke and Alfonso Valencia. 2002. The frame-based module of the SUISEKI information extraction system. *IEEE Intelligent Systems*, 17(2):14–20. doi:10.1109/5254.999215.
- Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32. doi:10.1023/A:1010933404324.
- Razvan Bunescu and Raymond Mooney. 2006. Subsequence kernels for relation extraction. *Advances in Neural Information Processing Systems*, 18:171–178.
- Chih-Chung Chang and Chih-Jen Lin, 2001. *LIB-SVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Timur Fayruzov, Martine De Cock, Chris Cornelis, and Veronique Hoste. 2008. The role of syntactic features in protein interaction extraction. In *Proc. of the 17th Conference on Information and Knowledge Management (CIKM'08)*, pages 61–68, Napa Valley, CA, USA. doi:10.1145/1458449.1458463.
- Timur Fayruzov, Martine De Cock, Chris Cornelis, and Veronique Hoste. 2009. Linguistic feature analysis for protein interaction extraction. *BMC Bioinformatics*, 10. doi:10.1186/1471-2105-10-374.
- Katrin Fundel, Robert Kuffner, and Ralf Zimmer. 2007. RelEx—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371. doi:10.1093/bioinformatics/btl616.
- Mark Goadrich, Louis Oliphant, and Jude Shavlik. 2005. Learning to extract genic interactions using gleaner. In Nédellec (Nédellec, 2005), pages 62–68.
- Mark A. Greenwood, Mark Stevenson, Yikun Guo, Henk Harkema, and Angus Roberts. 2005. Automatically acquiring a linguistically motivated genic interaction extraction system. In Nédellec (Nédellec, 2005), pages 46–52.
- Jorg Hakenberg, Conrad Plake, Ulf Leser, Harald Kirsch, and Dietrich Rebholz-Schuhmann. 2005. Identification of language patterns based on alignment and finite state automata. In Nédellec (Nédellec, 2005), pages 38–45.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.
- Min He, Yi Wang, and Wei Li. 2009. PPI finder: A mining tool for human protein-protein interactions. *PLoS ONE*, 4(2). doi:10.1371/journal.pone.0004554.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th International Conference on Computational Linguistics (COLING'92)*, pages 539–545, Nantes, France.
- Sophia Katrenko, M. Scott Marshall, Marco Roos, and Pieter Adriaans. 2005. Learning biological interactions from medline abstracts. In Nédellec (Nédellec, 2005), pages 53–58.
- Seonho Kim, Juntae Yoon, Jihoon Yang, and Seog Park. 2010. Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics*, 11. doi:10.1186/1471-2105-11-107.
- H. Ney, U. Essen, and R. Kneser. 1994. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38.
- Claire Nédellec, editor. 2005. *Learning language in logic – Genic interaction extraction challenge*, in Proc. of the 4th Learning Language in Logic Workshop (LLL'05), Bonn, Germany.
- Tu Minh Phuong, Doheon Lee, and Kwang Hyung Lee, 2003. *Learning rules to extract protein interactions from biomedical text*, volume 2637, pages 148–158. Springer Verlag. doi:10.1007/3-540-36175-8_15.
- Luboš Popelínský and Jan Blažák. 2005. Learning genic interactions without expert domain knowledge: Comparison of different ILP algorithms. In Nédellec (Nédellec, 2005), pages 59–61.
- Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3). doi:10.1186/1471-2105-9-S3-S6.
- Sebastian Riedel and Ewan Klein. 2005. Genic interaction extraction with semantic and syntactic chains. In Nédellec (Nédellec, 2005), pages 69–74.
- Chengjie Sun, Lei Lin, Xiaolong Wang, and Yi Guan, 2007. *Using maximum entropy model to extract protein-protein interaction information from biomedical literature*, volume 4681, pages 730–737. Springer Verlag. doi:10.1007/978-3-540-74171-8_72.
- Juan Xiao, Jian Su, GuoDong Zhou, and ChewLim Tan. 2005. Protein-protein interaction extraction: A supervised learning approach. In *Proc. of the 1st International Symposium on Semantic Mining in Biomedicine (SMBM 2005)*, pages 51–59, Hinxton, Cambridgeshire, UK.