The Extraction of Enriched Protein-Protein Interactions from Biomedical Text

Barry Haddow and Michael Matthews
School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh, Scotland, EH8 9LW
{bhaddow,mmatsews}@inf.ed.ac.uk

Abstract

There has been much recent interest in the extraction of PPIs (protein-protein interactions) from biomedical texts, but in order to assist with curation efforts, the PPIs must be enriched with further information of biological interest. This paper describes the implementation of a system to extract and enrich PPIs, developed and tested using an annotated corpus of biomedical texts, and employing both machine-learning and rule-based techniques.

1 Introduction

The huge volume of literature generated in the biomedical field is such that researchers are unable to read all the papers that interest them. Instead they must rely on curated databases, containing information extracted from the literature about, for example, which proteins interact.

These curated databases are expensive to produce as they rely on qualified biologists to select the papers, read them to extract the relevant information, enter this information into the database, and cross-check the information for quality control, a procedure which can be very time-consuming. If NLP techniques could be used to aid curators in their task then the costs of producing curated databases could be substantially reduced.

In the context of biomedical information extraction, there has been much recent interest in the automated extraction of PPIs (protein-protein interactions) from biomedical literature. The recent BioCreAtIvE Challenge highlights the desire to utilize these extraction techniques to automatically or semi-automatically populate curated PPI databases. However, just identifying the interactions is not necessarily sufficient, as curators typically require additional information about the interactions, such as the experimental method used to detect the interaction, and the names of any drugs used to influence the behaviour of the proteins. Furthermore, curators may only be interested in interactions which are experimentally proven within the paper, or where the proteins physically touch during the interaction.

This paper describes the implementation of a system designed to extract mentions of PPIs from biomedical text, and to enrich those PPIs with additional information of biological interest. The enriched information consists of properties (name-value pairs associated with a PPI, for example a directness property could indicate whether the interaction is direct or not direct) and attributes (relations between the PPI relation or its participating entities and other entities, such as the experimental method used to detect the PPI). This system for extracting and enriching PPIs was developed as part of the TXM programme, which aims to develop tools to help with the curation of biomedical papers.

After reviewing related work in the following section, a detailed description of how the annotated corpus was created and its descriptive statistics is provided in section 3. The methods used to extract the properties and attributes are explained in section 4, and then evaluated and discussed in section 5. Some conclusions and suggestions for further work are offered in section 6.

BioNLP 2007: Biological, translational, and clinical language processing, pages 145–152,
Prague, June 2007.

2 Related Work

There has been much recent interest in extracting PPIs from abstracts and full text papers (Bunescu and Mooney, 2006; Giuliano et al., 2006; Plake et al., 2005; Blaschke and Valencia, 2002; Donaldson et al., 2003). In these systems however, the focus has been on extracting just the PPIs without attempts to enrich the PPIs with further information. Enriched PPIs can be seen as a type of biological event extraction (Alphonse et al., 2004; Wattarujeekrit et al., 2004), a technique for mapping entities found in text to roles in predefined templates which was made popular in the MUC tasks (Marsh and Perzanowski, 1998). There has also been work to enrich sentences with semantic categories (Shah and Bork, 2006) and qualitative dimensions such as polarity (Wilbur et al., 2006).

Using NLP to aid in curation was addressed in the KDD 2002 Cup (Yeh et al., 2002), where participants attempted to extract records curatable with respect to the FlyBase database, and has been further studied by many groups (Xu et al., 2006; Karamanis et al., 2007; Ursing et al., 2001).

The Protein-Protein Interaction task of the recent BioCreAtIvE challenge (Krallinger et al., 2007) was concerned with selecting papers and extracting information suitable for curation. The PPI detection subtask (IPS) required participants not simply to detect PPI mentions, but to detect curatable PPI mentions, in other words to enrich the PPI mentions with extra information. Furthermore, another of the subtasks (IMS) required participants to add information about experimental methods to the curatable PPIs.

3 Data Collection and Corpus

3.1 Annotation of the Corpus

A total of 217 papers were selected for annotation from PubMed and PubMedCentral as having experimentally proven protein-protein interactions (PPIs). The papers were annotated by a team of nine annotators, all qualified in biology to at least PhD level, over a period of approximately five months.

The XML versions of the papers were used wherever possible, otherwise the HTML versions were used and converted to XML using an in-house tool. The full-text of each paper, including figure captions, was annotated, although the materials and methods sections were not included in the annotation.

From the 217 annotated papers, a total of 65 were selected randomly for double annotation and 27 for triple annotation. These multiply-annotated papers were used to measure inter-annotator agreement (IAA), by taking each pair of annotations on the same paper, and scoring one annotation against the other using the same algorithm as for scoring the system against the annotated data (see Section 5). Each doubly annotated paper contributed one pair of annotations, whilst the triply annotated papers contributed three pairs of annotations. The overall IAA score is the micro-average of the F_1 scores on each pair of corresponding annotations, where it should be emphasised that the F_1 does not depend on the order in which the annotated papers were combined. The multiply annotated papers were not reconciled to produce a single gold version, rather the multiple versions were left in the corpus.

The papers were annotated for entities and relations, and the relations were enriched with properties and attributes. The entities chosen for annotation were those involved in PPIs (Protein, Complex, Fusion, Mutant and Fragment) and those which could be attributes of PPIs (CellLine, DrugCompound, ExperimentalMethod and ModificationType). A description of the properties and attributes, as well as counts and IAA scores are shown in Tables 1 and 2.

Once annotated, the corpus was split randomly into three sections, TRAIN (66%), DEVTEST (17%) and TEST (17%). TRAIN and DEVTEST were to be used during the development of the system, for feature exploration, parameter tuning etc., whilst TEST was reserved for scoring the final system. The splits were organised so that multiply annotated versions of the same paper were placed into the same section.

3.2 Descriptive Statistics of Corpus

The total number of distinct PPIs annotated in the 336 papers was 11523, and the PPI IAA, measured using F_1, was 64.77. The following are examples of enriched PPIs, with the entities in bold face:

(1) **Tat** may also increase initiation of HIV-1 transcription by enhancing **phosphorylation** of **SP1**, a transcription factor involved in the basal HIV-1 transcription [14].

Name	Explanation	Values	Counts	Pct	IAA
IsPositive	The polarity of the statement about the PPI.	Positive	10718	93.01	99.57
		Negative	836	7.26	90.12
IsDirect	Whether the PPI is direct or not.	Direct	7599	65.95	86.59
		NotDirect	3977	34.51	61.38
IsProven	Whether the PPI is proven in the paper or not.	Proven	7562	65.63	87.75
		Referenced	2894	25.11	88.61
		Unspecified	1096	9.51	34.38

Table 1: The properties that were attached to PPIs, their possible values, counts and IAA

Name	Entity type	Explanation	Count	IAA
InteractionDetectionMethod	ExperimentalMethod	Method used to detect the PPI.	2085	59.96
ParticipantIdentificationMethod	ExperimentalMethod	Method used to detect the participant.	1250	36.83
ModificationBefore	Modification	Modification of participant before interaction.	240	68.13
ModificationAfter	Modification	Modification of participant after interaction.	1198	86.47
DrugTreatment	DrugCompound	Treatment applied to participant.	844	49.00
CellLine	CellLine	Cell-line from which participant was drawn.	2000	64.38

Table 2: The attributes that could be attached to the PPIs, with their entity type, counts and IAA

(2) To confirm that **LIS1** and **Tat** interact in vivo, we used **yeast two-hybrid system**, in which **Tat** was expressed as a bait and **LIS1** as a prey. Again, we found that **LIS1** and **Tat** interacted in this system.

In Example 1, the properties attached to the PPI between "Tat" and "SP1" are Referenced, Direct and Positive, and "phosphorylated" is attached as a ModificationAfter attribute. Example 2 shows a PPI between "Tat" and "LIS1" (in the second sentence) which is given the properties Proven, Direct and Positive, and has the InteractionDetectionMethod attribute "yeast two-hybrid system". This second example indicates that attributes do not have to occur in the same sentence.

Statistics on the occurrence of properties are shown in Table 1. For most of the property values, there are significant numbers of PPIs, except for Unspecified and Negative, which are used in less than 10% of cases. Note that annotators were permitted to mark more than one PPI between a given pair of entities if, for example, they wished to mark both Positive and Negative PPIs because the author is making a statement that proteins interact under one condition and not under another condition. For the purposes of data analysis and to make modelling easier, such PPIs have been collapsed to give a single PPI which may have multiple values for each property and attribute.

Table 2 shows occurrence statistics for attributes, where, as for properties, there can be multiple values for the same attribute. A notable feature of the attribute attachment counts is that certain attributes (ModificationBefore and DrugTreatment especially) are quite rarely attached, making it difficult to use statistical techniques.

Also shown in Tables 1 and 2 are the IAA figures for all properties and attributes. The IAA for properties is generally high, excepted for the Unspecified value of the IsProven property. This being something of a "none of the above" category means that the annotators probably have different standards regarding the uncertainty required before the PPI is placed in this class. The IAA for attributes is, on the whole, lower, with some attributes showing particularly low IAA (ParticipantIdentificationMethod). A closer investigation shows that the bulk of the disagreement is about when to attach, in other words if both annotators decide to attach an attribute to a particular PPI, they generally agree about which one, scoring a micro-averaged overall F_1 of 95.10 in this case.

4 Methods

4.1 Pipeline Processing

The property and attribute assignment modules were implemented as part of an NLP pipeline based on the LT-XML2 architecture[1]. The pipeline consists of tokenisation, lemmatisation, part-of-speech tagging, species word identification, abbreviation detection and chunking, named entiry recognition (NER) and relation extraction. The part-of-speech tagging uses the Curran and Clark POS tagger (Curran and Clark, 2003) trained on MedPost data (Smith et al., 2004), whilst the other preprocessing stages are all rule based. Tokenisation, species word identification and chunking were implemented in-house using the LT-XML2 tools (Grover and Tobin, 2006), whilst abbreviation extraction used the Schwartz and Hearst abbreviation extractor (Schwartz and Hearst, 2003) and lemmatisation used morpha (Minnen et al., 2000).

The NER module uses the Curran and Clark NER tagger (Curran and Clark, 2003), augmented with extra features tailored to the biomedical domain. Finally, a relation extractor based on a maximum entropy model and a set of shallow linguistic features is employed, as described in (Nielsen, 2006).

4.2 Properties

To assign properties to each PPI extracted by the relation extraction component, a machine learning based property tagger was trained on a set of features extracted from the context of the PPI. The property tagger used a separate classifier for each property, but with the same feature set, and both Maximum Entropy (implemented using Zhang Le's maxent[2]) and Support Vector Machines (implemented using svmlight[3]) were tested. To choose an optimal feature set, an iterative greedy optimisation procedure was employed. A set of potential features were implemented, with options to turn parts of the feature set on or off. The full feature set was then tested on the DEVTEST data with each of the feature options knocked out in turn. After examining the scores on all possible feature knockouts, the one which offered the largest gain in performance was selected and removed permanently. The whole procedure was then repeated until knockouts produced no further gains in performance. The resulting optimised feature set contains the following features:

ngram Both unigrams and bigrams were implemented, although, after optimisation, unigrams were switched off. The ngram feature uses *vlw backoff*, which means that words are replaced by their verb stems, backed off to lemmas and then to the word itself if not available. Furthermore, all digits in the words are replaced with "0". Ngrams are extracted from the sentences containing the participants in the PPI, and all sentences in between. Ngrams occurring before, between and after the participants of the PPI are treated as separate features.

entity The entity feature includes the text and type of the entities in the PPI.

headword This feature is essentially constructed in the same way as the ngram feature, except that only head verbs of chunks in the context are included, and the vlw backoff is not used.

entity-context In the entity context feature, the vlw backoffs of the two words on either side of each of the entities in the PPI are included, with their positions marked.

4.3 Attributes

For attribute assignment, experiments were performed with both rule-based and machine-learning approaches. The following sections summarise the methods used for each approach.

4.3.1 Rule-based

In the rule-based approach, hand-written rules were written for each attribute, using part-of-speech tags, lemmas, chunk tags, head words and the NER tags. In all, 20 rules were written. Each rule is

[1] http://www.ltg.ed.ac.uk/software/xml/

[2] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

[3] http://svmlight.joachims.org/

Rule	Protein	Prec	Count
P1 ATT P2	P2	100	13
P1 is *ATT* by *P2*	P1	100	1
ATT of *P2*	P2	86.1	112
ATT of *P1*	P1	74.5	80
P1 * *ATT* site	P1	72.2	13
P1 * *ATT* by * *P2*	P2	70.0	100
P1 * (*ATT* pass) * *P1*	P2	64.0	16
P1 * *ATT* * *P2*	P2	67.5	187
P2 ATT	P2	75.0	100
P2 - any-word ATT	P1	73.7	14

Table 3: The rules used to assign ModificationAfter attributes. The protein column indicates whether the attribute attaches to the 1st or 2nd protein, the prec field indicates the precision of the rule on the training set and the count indicates the number of times the rule applied correctly in training. In the rules, **P1** refers to the first protein, **P2** refers to the second protein, **ATT** refers to the attribute, * refers to any number of words, *any-word* refers to any single word, and pass refers to the passive voice. For example, the rule "*P2 - any-word ATT*" applied to the sentence "protein 1 is regulated by protein 2-dependent phosphorylation" would result in the attribute *phosphorylation* being assigned as the ModificationAfter attribute to *protein 1*.

ranked according to its precision as determined on the TRAIN set, and the rules are applied in order of their precision. This is particularly important with modification attributes which are constrained so that a given modification entity can only be attached once per interaction. Table 3 lists the rules used to assign the ModificationAfter attribute.

4.3.2 Machine Learning

For this approach, attributes are modelled as relations between PPIs and other entities. For each PPI in a document, a set of candidate relations is created between each of the entities in the PPI and each of the attribute entities contained in the same sentence(s) as the PPI[4]. If there are no entities of the appropriate type for a given attribute in the same sentence as the PPI, the sentences before and after the PPI are also scanned for candidate entities. Each of the candidate relations that correspond to attributes annotated in the gold standard are considered positive examples, whilst those that were not annotated are considered negative examples. For example, given the following sentence:

```
Protein A phosphorylates protein B
[Protein] [Modification] [Protein]
```

If the gold standard indicates a PPI between Protein A and Protein B with phosphorylates assigned as a ModificationAfter attribute to Protein B, four candidate relations will be created as shown in Table 4

Type	Entity 1	Entity 2	Label
Mod Before	Prot A	phosphorylates	neg
Mod Before	Prot B	phosphorylates	neg
Mod After	Prot A	phosphorylates	neg
Mod After	Prot B	phosphorylates	pos

Table 4: Candidate Attribute Relations for Protein A phosphorylates Protein B

A set of features is extracted for each of the examples and a maximum entropy (ME) model is trained using Zhang Le's maxent toolkit. The features used are listed below:

entity The text and part-of-speech of the attribute, as used for properties.

entity-context The entity context feature used for properties, except that the context size was increased to 4, and parts-of-speech of the context words were also included.

ngram This is the same as the ngram feature used for properties, except that unigrams were switched on.

entities-between The entities that appear between the two entities involved in the candidate relation.

parent-relation-feature Indicates the position of the attribute entity with respect to parent PPI (i.e. before, after, or in between). For attributes that are in between the two entities involved in the PPI, also indicates if the sentence is active or passive.

5 Evaluation

5.1 Properties

To score the property tagger, precision, recall and F_1 are calculated for each of the seven possible

[4] PPIs spanning more than 2 sentences were ignored

Name	Value	Baseline		Maximum Entropy		SVM	
		Gold	Predicted	Gold	Predicted	Gold	Predicted
IsPositive	Positive	96.87	97.33	97.10	98.22	97.08	98.27
	Negative	0.00	0.00	38.46	48.39	45.45	57.53
IsDirect	Direct	78.66	81.90	82.05	85.54	81.94	86.87
	NotDirect	0.00	0.00	58.92	54.33	60.80	63.44
IsProven	Proven	78.21	78.85	87.86	82.73	88.08	88.51
	Referenced	0.00	0.00	81.46	69.65	82.83	81.97
	Unspecified	0.00	0.00	25.74	29.41	22.77	28.00
Overall		74.20	76.24	83.87	83.33	84.09	86.79

Table 5: The performance of the property tagger, measured by training on TRAIN and DEVTEST combined, then testing on TEST. The two scores given for each system are for testing on gold PPIs, and testing on predicted PPIs. An F_1 score is shown for each property value, as well as a microaveraged overall score.

property values and then the F_1 scores are microaveraged to give an overall score. As mentioned in Section 3.1, all versions of the annotation for each multiply-annotated document were included in the training and test sets, taking care that all versions of the same document were included in the same set. This has the disadvantage that the system can never achieve 100% in cases where the annotators differ, but the advantage of giving partial credit where there is genuine ambiguity and the system agrees with one of the options chosen by the annotators.

The scores for all property values, tested on TEST, are shown in Table 5, both using the model (with Maximum Entropy and SVM) and using a baseline where the most popular value is assigned. Two scores are shown, the performance as measured when the test set has the gold PPIs, and the performance when the test set has the predicted PPIs, scored only on those PPIs where both system and gold agree. The relation extractor used to predict the PPIs is trained on the same documents as were used to train the property tagger.

To see which features were most effective, a knockout (lesion) test was conducted in which features were knocked out one by one and performance was measured on the DEVTEST set. In each feature knockout, one of the features from the list in Section 4.2 was removed. Table 6 shows how the overall performance is affected by the different knockouts. From the knockout experiment it is clear that the ngram (actually bigram) feature is by far the most effective, with the other features only contributing marginally to the results.

Feature	Knockout score	Difference
vanilla	86.08	0.00
ngram	81.86	-4.22
entity	85.30	-0.77
headword	84.38	-0.50
entity-context	85.54	-0.54

Table 6: The effect of knocking out features on the property score. Tests are conducted by training on TRAIN and testing on DEVTEST, on predicted PPIs. “vanilla” refers to the case where the optimal features set is employed.

5.2 Attributes

The attributes are scored in the same manner as the properties. Table 7 summarises the results for both the rule-based and machine learning attribute systems. These are compared to a baseline system that simply attaches the nearest entity of the appropriate type for each attribute.

5.3 Discussion

The results for the more common property values are generally close to human performance (as measured by IAA), however performance on both IsNegative and Unspecified is fairly low. In the case of Unspecified, the IAA is also low, making it likely that the training and test data is inconsistent, compounding the problem of the low occurrence rate of this value. The Negative value also suffers from a low occurrence rate, leading to an imbalance between Negative and Positive which makes life hard for the

Attribute	Baseline		Rule-based		Machine Learning	
	Gold	Predicted	Gold	Predicted	Gold	Predicted
InteractionDetectionMethod	36.02	39.71	39.22	41.38	37.02	46.81
ParticipantIdentificationMethod	08.68	09.27	12.32	12.87	03.37	05.97
ModificationBefore	13.10	16.00	42.22	43.84	04.88	08.33
ModificationAfter	43.37	46.00	64.93	73.04	62.32	69.64
DrugTreatment	49.57	51.11	51.29	53.33	13.90	24.52
CellLine	50.19	45.90	54.47	50.47	45.13	42.28
Overall	29.68	30.32	45.26	48.32	32.08	43.11

Table 7: The performance of the attribute tagger, on TEST. The two scores given for each system are for testing on gold PPIs, and testing on predicted PPIs. Performance on each attribute value is measured using F_1, and then microaveraged to give an overall figure.

machine learners. However it is also possible that the shallow linguistic features used in these experiments are not sufficient to make the sometimes subtle distinction between a negative statement about an interaction and a positive one, and that models based on a deeper linguistic analysis (e.g. parse trees as in (Moschitti, 2004)) would be more successful. Note also that the feature set was optimised for maximum performance across all property values, with all given equal weight, but if some values are more important than others then this could be taken into account in the optimisation, with possibly different feature sets used for different property names.

The results for the attributes using the rule-based system are approximately 75% of human performance and are higher than results for the machine learning system. However, for the ModificationAfter, CellLine, and InteractionDetectionMethod attributes, which occur more frequently than the other attributes and have higher IAA, the machine learning system is competitive and even slightly outperforms in the case of the InteractionDetectionMethod. The scores are directly correlated with the IAA and both the scores and the IAA are higher for the attributes that tend to occur in the same sentence as the PPI. On a practical level, this suggests that those who hope to create similar systems would be advised to start with local attributes and pay particular attention to IAA on non-local attributes.

5.4 Further work

As regards properties, good results were obtained using shallow linguistic features, but it would be interesting to learn whether machine learning techniques based on a deeper linguistic analysis would be more effective. Also, properties were treated as additional information added on to the PPIs after the relation extractor had run, but perhaps it would be more effective to combine relation extraction and property tagging to, for example, consider positive and negative PPIs as different types of relations.

For attributes, it would be interesting to combine the rule-based and machine learning systems. This has the advantage of having a system that can both learn from annotated data when it exists, but can be potentially improved by rules when necessary or when annotated data is not available. Another issue may be that some attributes might not be represented explicitly by a single entity in a document. For example, an experimental method may be described rather than explicitly stated. Attributes that are not local to the PPI caused difficulty for both the annotators and the system. It would be interesting to see if it is easier to attach attributes to a single PPI that has been derived from the text, rather than attempting to assign attributes to each specific mention of a PPI within the text. This could be accomplished by attempting to merge the information gathered from each relation along the lines described in (Hobbs, 2002)

Since the main motivation for developing the system to extract enriched PPIs was to develop a tool to aid curators, it would be useful to know how effective the system is in this task. Aside from (Karamanis et al., 2007), there has been little work published to date on the effect that NLP could have on the curation process. In the most recent BioCreAtIvE evaluation, the PPI subtasks were concerned with automating information extraction tasks typically performed by curators such as distinguishing between curatable and non-curatable PPI mentions and specifying the details of how the PPI was detected.

6 Conclusions

A system was implemented for enriching protein-protein interactions (PPIs) with properties and attributes providing additional information useful to biologists. It was found that a machine learning approach to property tagging, using simple contextual features, was very effective in most cases, but less effective for values that occurred rarely, or for which annotators found difficulty in assigning values. For the attributes, sparsity of data meant that rule-based approaches worked best, using fairly simple rules that could be quickly developed, although machine learning approaches could be competitive when there was sufficient data.

7 Acknowledgements

The authors are very grateful to the annotation team, and to Cognia (`http://www.cognia.com`) for their collaboration on the TXM project. This work is supported by the Text Mining Programme of ITI Life Sciences Scotland (`http://www.itilifesciences.com`).

References

Erick Alphonse, Sophie Aubin, Philippe Bessieres, Gilles Bisson, Thierry Hamon, Sandrine Lagarrigue, Adeline Nazarenko, Alain-Pierre Manine, Claire Nedellec, Mohamed Ould Abdel Vetah, Thierry Poibeau, and Davy Weissenbacher. 2004. Event-based information extraction for the biomedical domain: the Caderige project.

C. Blaschke and A. Valencia. 2002. The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, (17):14–20.

Razvan Bunescu and Raymond Mooney. 2006. Subsequence kernels for relation extraction. In Y. Weiss, B. Schlkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. Cambridge, MA.

James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-2003*.

Ian Donaldson, Joel Martin, Berry de Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D. Bader, Katerina Michalickova, Tony Pawson, and Christopher W. V. Hogue. 2003. PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4:11.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the EACL*.

Claire Grover and Richard Tobin. 2006. Rule-Based Chunking and Reusability. In *Proceedings of LREC 2006*.

Jerry R. Hobbs. 2002. Information extraction from biomedical text. *Journal of Biomedical Informatics*, 35(4):260–264.

N. Karamanis, I. Lewin, R. Seal, R. Drysdale, and E. J. Briscoe. 2007. Integrating natural language processing with flybase curation. In *Proceedings of PSB 2007*.

Martin Krallinger, Florian Leitner, and Alfonso Valencia. 2007. Assessment of the Second BioCreative PPI Task: Automatic Extraction of Protein-Protein Interactions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

E. Marsh and D. Perzanowski. 1998. MUC-7 evaluation of IE technology: Overview of results. In *Proceedings of MUC-7*.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of INLG 2000*.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the ACL*.

Leif Arda Nielsen. 2006. Extracting protein-protein interactions using simple contextual features. In *Proceedings of the BioNLP 2006 at HLT/NAACL 2006*.

Conrad Plake, Jörg Hakenberg, and Ulf Leser. 2005. Optimizing syntax-patterns for discovering protein-protein-interactions. In *Proc ACM Symposium on Applied Computing, SAC, Bioinformatics Track*, volume 1, March.

A.S. Schwartz and M.A. Hearst. 2003. Identifying abbreviation definitions in biomedical text. In *Proceedings of PSB 2003*.

Parantu K. Shah and Peer Bork. 2006. Lsat: learning about alternative transcripts in medline. *Bioinformatics*, 22(7):857–865.

L. Smith, T. Rindflesch, and W. J. Wilbur. 2004. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321.

Björn M. Ursing, Frank H. J. van Enckevort, Jack A. M. Leunissen, and Roland J. Siezen. 2001. Exprot - a database for experimentally verified protein functions. *In Silico Biology*, 2:1.

Tuangthong Wattarujeekrit, Parantu K. Shah, and Nigel Collier. 2004. PASBio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics*, 5:155.

John W. Wilbur, Andrey Rzhetsky, and Hagit Shatkay. 2006. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC Bioinformatics*, 7:356+, July.

H. Xu, D. Krupke, J. Blake, and C. Friedman. 2006. A natural language processing (nlp) tool to assist in the curation of the laboratory mouse tumor biology database. *AMIA Annu Symp Proc*.

Alexander Yeh, Lynette Hirschman, and Alexander Morgan. 2002. Background and overview for KDD cup 2002 task 1: information extraction from biomedical articles. *SIGKDD Explor. Newsl.*, 4(2):87–89.