Identifying Protein-Protein interactions in Biomedical publications

Alejandro Figueroa `figueroa@dfki.de` **Günter Neumann** `neumann@dfki.de`

DFKI - LT Lab, Stuhlsatzenhausweg 3, D - 66123, Saarbrücken, Germany

Abstract

The paper describes the approaches and the results of our participation in the protein-protein interaction (PPI) extraction task (sub-tasks 1 to 3) of the BioCreative II challenge.[1] The core of our approach is to analyse the logical forms of those sentences which contain the mentioning of relevant protein names, and to rank the sentences from which the relations where extracted using the class descriptors computed in the sub-task 1 and interaction sentences from the Christine Brun corpus.

Keywords: Protein-Protein interactions identification, Predicate Analysis

1 Introduction

One of the goals of the Question Answering group at the DFKI LT-Lab is taking part in standard evaluations such as TREC or CLEF. During the last three years, our group has focused on the Cross-Lingual German-English, English-German and monolingual German tracks of the CLEF campaign. Results have been strongly encouraging, obtaining the best results for these tracks [13, 14, 15].

In QA the current research focus is still on domain–open QA in order to answer term-based questions like *Where was the "killer smog" of 1952 which resulted in 4,000 deaths?* from newspaper articles. However, there is an increasing interest to explore also domain–specific QA, i.e., to answer domain–specific questions from domain–specific sources. Here, event specific questions are of interest, which require the identification of relevant relation instances, e.g., in order to answer a question like *How does GUKH interacts with DLG?* from scientific articles.

Our approach is to consider domain–specific QA as a kind of *on-demand information extraction* where the NL question describes important constraints for the relation instances that have to be extracted from the answer sources. This perspective actually motivated our interest in the BioCreative challenge, especially in the Protein-Protein interaction subtask. Of course, the focus in the BioCreative challenge is on off-line information extraction in the sense that the information request (i.e., the question) is pre-specified and that all possible valid relation instances have to be extracted (i.e., the answer candidates). For researchers in question answering like us, there are important subtasks in common for on-demand and off-line information extraction, like named entity recognition, relation mining, co-reference detection, concept name disambiguation, etc.

Since BioCreative II was our first excursion into Information Extraction in Biology, our objectives were: (a) learn about the inherent challenges and share our experience, and (b) discriminate key components of systems that deal with natural language texts in the biological domain. Here, the main motivation raises from the way that biological texts are written: a plenty of technical words and

[1]The work presented here was partially supported by a research grant from the German Federal Ministry of Education, Science, Research and Technology (BMBF) to the DFKI project `HyLaP` (FKZ: 01 IW F02) and the EC-funded project QALL-ME.

complex sentence structures as well as a high term variation, especially gene names. Assessing several Natural Language Processing techniques is hence positively encouraging, and by the same token, our group focused essentially on covering the sub-tasks (a) Protein Interaction Article Sub-task (IAS), (b) Protein Interaction Pairs Sub-task (IPS), and (c) Protein Interaction Sentences Sub-task (ISS).

In the next section, we firstly describe our principle approach and then focus on particular solutions for the different sub-tasks in the subsequent sections. In section 6 we briefly discuss our results, which – of course – we interpret as "the glass is half full".

2 Predicate Analysis

Predication computes the semantic representation of a sentence. This representation distinguishes relationships or semantic roles played by its different constituents within a semantic frame[10]. To neatly illustrate this, consider the sentence "*GUKH interacts with DLG in vivo*", its corresponding predicate representation is given by:

interact("GUKH", "with DLG", "in Vivo")

In this representation, the verb is the predicate and the remaining constituents are arguments. Labels are then assigned to each argument according to their role in the predicate. The level of specification can be abstract such as `VERB, SUBJECT, OBJECT`, or specific to the different framesets of a particular verb. Good examples are the two framesets for the verb "*inhibit*" (see [11] for examples in the PropBank[16]):

1. **inhibit**(preventor entity, thing prevented from happening), i. e. "*Influenza virus NS1 protein inhibits pre-mRNA splicing* ".

2. **inhibit**(preventor entity, thing prevented from happening, medium), for instance: "*ArhGAP9 inhibits Erk and p38 activation through WW domain binding Boon K Ang1 ,2*".

Each frameset is seen as a different semantic frame. The motivation behind applying predication to discriminate protein interaction is two-fold: (a) since proteins interactions are likely to be expressed by complex semantic constructions at the sentence level [4, 6], and (b) the existence of tools, like Montylingua[17], which compute a semantic representation of a raw text in English. Montylingua specifically extracts tuples *verb(subject, objects)*, which are an abstract predicate-argument representation of sentences in a given text.

3 Document Classification

In this sub-task, documents containing relevant protein interaction information must be accurately identified. This identification must be performed by accounting solely for their headlines and abstracts. For this purpose, systems were allowed to submit three different runs, and in our case, to test three different strategies. Two out of these three strategies started stepwisely pre-processing the training and testing sets as follows:

1. **Protein name removal** Since protein and gene names are the most obvious source of classification bias[1], they are distinguished by Abner[18] and replaced with the word "*Protein*" afterwards.

2. **Lemmatization** In this step, words are lemmatized by means of MontyLingua[17], in order to avoid counting several morphological inflections of the same term as occurrence of different words.

3. **Sentence normalization** Abstracts are split into sentences by means of JavaRap[19] and normalized afterwards. This normalization consists chiefly in inserting spaces between punctuation and words, this way our methods avoid also misinterpreting words followed by their punctuation as occurrence of different words. By the same token, all words are lowercased.

4. **Bag of words** Each abstract is represented as a bag of words. These words are distinguished by means of spaces and every word is linked to their frequency on the corresponding abstract. Stop-words[2] are removed from each bag.

While our strategies were dealing with this task, we found that the unbalanced training data, caused by the strong bias in favour of positive samples, was a major problem. Consequently, strategies aiming specifically for dealing with unbalanced data were explored. The first two runs (`RUN I` and `RUN II`) were based on the binary Bayes classifier presented in [2]. In these runs, we trained two classifiers: one with abstracts and the other with headlines. Documents in the test set were eventually ranked by weighting the output of both classifiers in the following way:

$$r_d(D) = \begin{cases} r_h(D) * r_a(D) & \text{if } r_h(D) <> 0 \text{ and } r_a(D) <> 0. \\ r_h(D) & \text{if } r_a(D) == 0. \\ r_a(D) & \text{if } r_h(D) == 0. \end{cases}$$

Where $r_h(D)$ and $r_a(D)$ are the output (corresponding to a document D) of the Maximum Entropy classifier trained with headlines and abstracts respectively. A new document D was considered containing relevant protein interaction information, if $r_d(D) > 1$, otherwise irrelevant. The training tuples were chosen by means of a 10-fold validation and due to three reasons, they were deliberately selected only from negatives and noisy positives samples: (a) we found that positive samples did not improve results, (b) markedly reduce the size of the training set, and (c) given the fact that the test set belongs solely to the positive and negative class, we clearly intended to increase the robustness of our classifiers by decreasing their dependence upon positive samples. These first two runs differ fundamentally in the training model obtained by the 10-fold cross validation.

`RUN III` was based on the approach presented in [9]. In this approach, documents and categories are seen as sets of independent words. For each category, this classifier creates two data structures: semantics-oriented topic words and surface focused index words with a high discrimination value. Documents are classified by means of two category rankings (each for index and topic words) which are combined to one ranking (m-ary classifier) afterwards. This classifier was trained with non pre–processed negative and positives samples only.

4 Protein protein interaction identification

This sub-task aims at recognising protein interactions from full text articles. The underlying assumption of our methods is that interacting proteins are expected to co-occur in many sentences along the respective article, and therefore, in several semantic frames. Some of these semantic frames are accordingly more likely to indicate whether they interact or not. The flow of our strategy is as follows:

1. **Pre-processing** starts by extracting the content from the PDF2TXT version of the article and splitting it into sentences by means of JavaRap[19] afterwards. The higher frequent sentence was interpreted as the title or headline of the article, since it is seldom directly recognised from the text and it is usually repeated. Like [3], citations were permanently removed by means of purpose-built regular expressions, this way the quality of the predicate analysis noticeably improves. Another key issue is that sections within documents are identified by searching for special

[2]The stop-list from [20] is used. It contains 319 highly frequent closed class forms.

tags such as "*MATERIALS*", "*REFERENCES*", "*ACKNOWLEDGMENTS*". In case that no section was correctly identified, the article is seen as containing only one section. Sentences are then associated with their corresponding sections afterwards.

2. **Protein detection** is performed by Abner across the whole document. Since our system works with predicates at the sentence level, protein references across sentences must be unveiled. For this specific purpose, we took advantage of the full implementation of [5] provided by JavaRap, instead of its partial implementation presented in [4].

3. **Predicate Analysis** takes all sentences containing at least two recognised proteins and identifies its predicate and arguments. This semantic structure is a crucial aspect of our strategy (also in [4, 6]), because the role of proteins within sentences signals their relation and verbs whether this relation a protein-protein interaction is or not [4, 3, 7]. Arguments with no protein mentions were for this reason also completely discarded. Another thing is, headlines of articles are usually ungrammatical, MontyLingua could not then distinguish their structure. Our system keeps hence track of co-occurring proteins within headlines, because they are likely to signal a relevant relation.

4. **Gene name normalisation** maps protein names, which occur in at least one predicate, to their corresponding `UniProt Accession Numbers`. This mapping consists of the next steps:

   (a) The `UniProt light Knowledge Base` was indexed by normalized terms extracted from the following columns: description and gene name lines, gene synonyms, locus and ORF names, keywords. These terms indexed their corresponding accession numbers and their normalization consisted in leaving only letters and numbers [8].

   (b) Candidate protein keys are extracted by looking for matches across this index. Firstly, our system attempts to find exact string matches, if it does not succeed, it looks for inexact matches. The first matching considers only the exact gene name identified in the text, and the second accounts solely for the letters and number in the distinguished gene name.

   (c) Our system searches for co-occurring pairs organism-protein within sentences. If any highly co-occurring pair exists, the organism is used for disambiguating the key.

   (d) If key ambiguity still exists, our system tries to discover known interacting key pairs in the Expasy Knowledge Base[21].

   (e) If our system cannot disambiguate the key, the first key in alphabetical order is selected.

   Protein names were eventually replaced in predicates with their mapped accession numbers. Each predicate provided accordingly the following interacting pairs:

   (a) The subject was paired with each argument.

   (b) Each argument was paired with the other arguments.

5. **Ranking predicates and protein pairs** Let S be the set of $1 \leq s \leq \mid S \mid$ sentences extracted from a given article D and S_s the s-th sentence in S, $1 \leq s \leq \mid S \mid$. Each sentence $S_s \in S$ is then ranked according to the potential of its words for expressing protein interactions. The computation of this potential is based mainly on the following equation:

$$word\_potential(S_s) = \sum_{\forall w_i \in S} P^{ISS}(w_i) + W^{IAS}(w_i)$$

Where $P^{ISS}(w_i)$ is the probability that the word w_i occurs within interaction sentences across abstracts in the Christine Brun corpus. $W^{IAS}(w_i)$ is given by:

$$W^{IAS}(w_i) = W^{+}(w_i) - W^{-}(w_i)$$

Where $W^+(w_i)$ and $W^-(w_i)$ are the likelihood of w_i to the noisy positive and negative class respectively (previously computed in sub-task I (see section 3)). Additionally, we define the potential of a verb for expressing protein interactions as P^{IAS}_{verb}, the probability that a protein and a particular verb co-occur in the same sentence across positive and noisy positive abstracts given in sub-task I. The rank of a sentence is eventually defined as follows:

$$rank(S_s) = \Gamma * (1 + word\_potential(S_s)) * (1 + \sum_{\forall \vartheta_r \in \vartheta(S_s)} P^{IAS}_{verb}(verb(\vartheta_r))) \quad (1)$$

Where $verb(\vartheta_r)$ is a function which returns the verb in the predicate ϑ_r, $\vartheta(S_s)$ a function which returns the identified predicates for S_s, and Γ is a weight according to the section in which S_s occurs. $\Gamma = 1$ for all sections, apart from "*MATERIALS*", "*MATERIALS AND METHODS*", "*RESULTS AND DISCUSSION*", "*RESULTS*", "*EXPERIMENTAL*", "*DISCUSSION*", "*EXPERIMENTAL PROCEDURES*", which their value for Γ was set to two. The rank of the interaction of two proteins p_1 and p_2 is given by:

$$rank(p_1, p_2) = \tau(g_1, g_2)\gamma \sum_{\forall S_s \in S} \lambda(p_1, p_2, S_s) * rank(S_s)$$

Where $\lambda(p_1, p_2, S_s)$ is the number of predicates $\vartheta_r \in \vartheta(S_s)$ in which p_1 and p_2 occur. The weight γ favours pairs occurring in the title. $\tau(g_1, g_2)$ favours interaction pairs that can be found in the Expasy Knowledge Base (step 4.d).

6. **The three runs** were generated according to the following criteria:
   (a) **RUN I**: All identified ranked pairs.
   (b) **RUN II**: All ranked pairs that satisfactorily fulfil the next rule:

   $$rank(p_1, p_2) > 0.1 * rank^*$$

   Where $rank^*$ is the rank value of the higher ranked pair.
   (c) **RUN III**: Top five ranked pairs.

5 Protein protein interaction sentence Ranking

This sub-task asks participants to provide, for each protein interaction pair, a ranked list of at most five text passages (maximal three sentences per passage) describing their interaction. For this sub-task, we submitted only one run. Our system took advantage of the ranking provided by sub-task II (eq. 1) and selected the top five ranked sentences for each protein interaction pair. Each sentence was aligned with the source HTML document as follows:

1. The first word in the sentence was used as an anchor. This anchor signals the start of a window of two times the length of the ranked sentence.
2. Words were placed in each window according to their relative position within the ranked sentence. When a word could not be accurately located within the window, it was marked with a "*". The window with less "*" was eventually selected.
3. If the last word in the selected sentence was properly aligned, the window is cut off at the end of this word.

6 Results

The following section describes the results obtained by our system in details.

6.1 Document Classification

Table 1 and 2 provide the results obtained by each run for the document classification sub-task:

Table 1: Results overview.

	Precision	Recall	Accuracy	F-Score	AUC	Error Rate
RUN I	0.527	0.986	0.550	**0.687**	0.795	0.44
RUN II	0.518	**0.992**	0.536	0.681	0.797	0.46
RUN III	**0.577**	0.725	**0.597**	0.643	0.589	**0.40**

RUN I and RUN II finished with a F-score about the mean of all systems (0.6868). Conversely, RUN III achieved a slightly worse F-score, but a higher accuracy. Table 2 shows the confusion matrices for each run:

Table 2: Confusion matrices.

	TP	FP	TN	FN
RUN I	370	332	43	5
RUN II	372	345	30	3
RUN III	272	199	176	103

The number of FP gives the reason for the high recall and low precision of RUN I and RUN II, caused by the assignment of many negative test documents to the positive class. Table 2 also shows that RUN III improved the recall of the negative class at expenses of its precision, which is a consequence of the few number of negative training samples used for our classifiers. Table 3 provides greater details about the results achieved by the three runs:

Table 3: Comparisson of the three runs.

	RUN I	RUN II	RUN III
RUN I	-	19	116
RUN II	6	-	109
RUN III	153	158	-

This table compares two runs by taking documents, for which their prediction differ, and counting the number of correct forecast for each run. For instance, RUN I and RUN II obtained different predictions for 25 documents and six cases were correctly labelled by RUN II, while 19 cases by RUN I. This result envisages that the combination of the output of several classifiers can improve results.

6.2 Protein-protein interaction identification

Protein-protein interaction prediction

Tables 4 and 5 supply our per document and overall results respectively. In these tables, EVAL stands for all articles and SP_EVAL for the subset containing exclusively SwissProt interaction pairs.

The total recall of our system was about the mean respecting the 45 runs submitted by all systems. In case of EVAL, our system achieved 0.09 (0.1064 overall) and in case of SP_EVAL, it finished with 0.094 (0.1150 overall). In contrast to recall, results concerning precision are unconvincing. Given this sharp difference, it can be concluded that our system discovers interacting pairs of proteins along with a large amount of incorrect pairs. Looking closer upon table 5, we additionally observe that the decrease in recall from RUN I to RUN II and RUN III leads us to conclude that interaction pairs tend

Table 4: Mean values for the three different runs (per document).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.01	0.11	0.018	0.011	0.11	0.019
RUN II	0.029	0.056	0.035	0.025	0.056	0.032
RUN III	0.026	0.087	0.036	0.023	0.087	0.034

Table 5: Overall result for the three different runs.

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.01	**0.09**	0.018	0.01	**0.094**	0.019
RUN II	0.029	0.030	0.034	0.025	0.026	0.026
RUN III	0.018	0.05	0.027	0.019	0.05	0.027

to be ranked low (RUN II and RUN III consider only a subset of the highest ranked pairs of RUN I). These conclusions motivate the usage of Montylingua for distinguishing protein interactions, but a strategy that can filter out misleading interactions along with a better ranking strategy is necessary, this way the noise could be reduced and the precision similarly increased.

Interactor proteins Normalisation.

Tables 6, 7 and 8 gives our results for the normalisation of interactors.

Table 6: Mean values for interactor proteins normalization (all evaluated articles).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.06	**0.29**	0.095	0.066	**0.32**	0.11
RUN II	0.11	0.18	0.13	0.11	0.19	0.135
RUN III	0.09	0.20	0.11	0.095	0.22	0.123

Table 7: Mean values for interactor proteins normalization (all evaluated articles with predictions).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.06	0.31	0.1	0.072	0.34	0.11
RUN II	0.14	0.23	0.17	0.15	0.26	0.18
RUN III	0.11	0.27	0.15	0.13	0.30	0.17

Our gene normalisation strategy achieves a slightly better recall than the mean considering all evaluated documents and a slightly worse recall taking into account only articles with predictions. In the three cases, RUN II was the best, because of its higher precision and F-Score. The higher recall of RUN I is a logical consequence of accounting for an unfiltered set of pairs.

Table 9 provides the performance of our gene mormalisation strategy: 361 out of 1306 protein names were correctly identified and correctly mapped to their database entries, and 268 out of 896 taking into account only SwisssProt entries. The difference in the number of correctly identified protein names shows that our ranking strategy ranks many relevant interacting proteins low. This could be

Table 8: Mean values for interactor proteins normalization (Overall SwissProt interactor pairs).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.06	**0.28**	0.09	0.04	**0.3**	0.074
RUN II	0.13	0.15	0.14	0.097	0.158	0.12
RUN III	0.09	0.18	0.12	0.064	0.19	0.096

Table 9: Number of interactor protein-article associations.

	EVAL				SP_EVAL			
	Correct	Wrong	Missed	Predicted	Correct	Wrong	Missed	Predicted
RUN I	361	6011	945	6372	268	6104	628	6372
RUN II	197	1273	1109	1470	142	1328	754	1470
RUN III	238	2421	1068	2659	171	2488	725	2659

due to the detection of sentences, some relevant sentences could not be parsed, therefore, the relation between proteins could not be properly determined. Results show that this is the most critical module in our system.

6.3 Protein-protein interaction sentence ranking

Our system found out 590 sentences that matched the gold standard (manually selected passages), 285 out of these 590 were unique. Since our system returned a long list of interacting proteins in sub-task II, it returned a huge list of 21431 sentences for this sub-task (10422 unique), which caused an MMR of 0.3785.

7 Conclusions

In this work, we presented our first participation in an evaluation of Information Extraction Systems in Biology. For a future participation, we envisage the following improvements:

1. Combining the output of several classifiers in order to enhance the accuracy of our predictions and the robustness of our classifier.
2. The usage of language models that consider more contextual information, like bi-grams.
3. A bootrapping strategy can also take advantage of recognised pairs, this way undetected sentences by Montylingua can be identified, bringing about an improvement in the ranking of sentences and interacting protein pairs.
4. The usage of LSA[12] and the Web for discriminating the source organism of a protein.

References

[1] Marcotte, E. M., Xenarios I. and Eisenberg, D, *Mining literature for protein-protein interactions*, Bioinformatics, 17:4, pp. 359–363, 2001.

[2] Rennie, J. D. M., Shih, L., Teevan, J. and Karger, D. R., *Tackling the Poor Assumptions of Naive Bayes Text Classifiers*, in Proceedings of ICML-2003, Washington DC, 2003.

[3] Huang, M., Zhu, X., Hao, Y., Payan, D. G., Qu, K. and Li, M., *Discovering patterns to extract protein-protein interactions from full texts*, Bioinformatics, 20:18, pp. 3604–3612, 2004.

[4] Sekimizu, T., Park, H. and Tsujii, J, *Identifying the interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts*, In Genome Informatics Series: Proceedings of the Workshop on Genome Informatics, Vol. 9, pp. 62-71, 1998.

[5] Lappin, S. and Leass, H. J., *An algorithm for pronominal anaphora resolution*, Computational Linguistics, 20:4, pp. 535–561, 1994.

[6] Ahmed, S., Chidambaram, D., Davulcu H. and Baral C., *IntEx: A Syntactic Role Driven Protein-Protein Interaction extractor for Bio-Medical Text*, in Proceedings ACL-05/ISMB-05, pp. 54–61, 2005.

[7] Hatzivassiloglou V. and Weng W., *Learning Anchor Verbs for Biological Interaction Patterns from Published Text Articles*, Int J Med Inf., 67, pp. 19–32, 2002.

[8] Wellner, B., *Weakly Supervised Learning Methods for Improving the Quality of Gene Name Normalization Data*, in Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pp. 1–8, Detroit, June, 2005.

[9] Neumann G. and Kappes M., *A simple base-line text-categorizer for evaluating the effect of feature extraction in text mining applications*, in abstract booklet accompanying the 26th Annual Conference of the German Classification Society (GfKl 2002), July 22-24, 2002, University of Mannheim, Germany.

[10] Gildea D. and Jurafsky D., *Automatic Labeling of Semantic Roles. Computational Linguistics*, Computational Linguistics, 28:3, pages 245–288, 2002.

[11] Palmer, M., Gildea D. and Kingsbury P., *The Proposition Bank: An Annotated Corpus of Semantic Roles*, Computational Linguistics, 31:1, pages 71–106, 2005.

[12] Deerwester, S., Dumais, S., T., Furnas, G., W., Landauer, T., K. and Harshman R., *Indexing By Latent Semantic Analysis*, Journal of the American Society For Information Science, 41, 391–407, 1990.

[13] Sacaleanu B. and Neumann G. *DFKI-LT at the CLEF 2006 Multiple Language Question Answering Track.* In Working Notes for the CLEF 2006 Workshop, August, Alicante, Spain, 2006

[14] Neumann G. and Sacaleanu B. *DFKI's LT-lab at the CLEF 2005 Multiple Language Question Answering Track.* In Working Notes for the CLEF 2005 Workshop, 21-23 September, Vienna, Austria, 2005.

[15] Neumann G. and Sacaleanu B. *Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering System.* In: C. Peters et al. (Eds): Clef 2004, LNCS 3491, pp. 411-422, 2005, Springer Berlin Heidelberg.

[16] `http://www.cs.rochester.edu/~gildea/PropBank/Sort/`

[17] `http://web.media.mit.edu/~hugo/montylingua/`

[18] `http://www.cs.wisc.edu/~bsettles/abner/`

[19] `http://www.comp.nus.edu.sg/~qiul/NLPTools/JavaRAP.html`

[20] `htpp://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words`

[21] `http://www.expasy.org/sprot/`