Rule-based Protein Term Identification with Help from Automatic Species Tagging

Xinglong Wang

School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW, Scotland
xwang@inf.ed.ac.uk

Abstract. In biomedical articles, terms often refer to different protein entities. For example, an arbitrary occurrence of term *p53* might denote thousands of proteins across a number of species. A human annotator is able to resolve this ambiguity relatively easily, by looking at its context and if necessary, by searching an appropriate protein database. However, this phenomenon may cause much trouble to a text mining system, which does not understand human languages and hence can not identify the correct protein that the term refers to. In this paper, we present a Term Identification system which automatically assigns unique identifiers, as found in a protein database, to ambiguous protein mentions in texts. Unlike other solutions described in literature, which only work on gene/protein mentions on a specific model organism, our system is able to tackle protein mentions across many species, by integrating a machine-learning based species tagger. We have compared the performance of our automatic system to that of human annotators, with very promising results.

1 Introduction

Biomedical literature provide a wealth of information on genes, proteins and their interactions. To make this vast quantity of data manageable to biologists and to utilise them in conjunction with bioinformatics methods, it is desirable to automatically organise the free text information into machine-readable, well-defined form. A growing body of work has been devoted to recognition of protein and gene names, and to extraction of their interactions. In this paper, we report our work on another fundamental task of identification of "ambiguous" mentions of biological entities in documents, which we believe has not been adequately addressed in the literature.

We call the task of grounding a biological term in text to a specific identifier in a referent database as *Term Identification* (TI) [1]. TI is crucial for the automated processing of the biomedical literature [2, 3]. For example, a system that extracts protein-protein interactions would ideally collapse interactions involving the same proteins, which might appear in different word forms in articles. This

paper describes our system for identification of protein entities.[1] We summarise the sources of ambiguity and the corresponding disambiguation tasks that need to be carried out as follows:[2]

1. **Term Normalisation** [4] A protein term may appear in text in various forms, such as orthographic variants (e.g., *IL-5* and *IL5*), acronyms or abbreviations (e.g., *IL5* for *Interleukin-5*), etc. Term normalisation is to "normalise" such variants to their canonical form, as recorded in a protein database.
2. **Term Disambiguation** [1] A protein term may refer to different protein entities across different model organisms (e.g., *IL5* can be *IL5 Homo sapiens* or *IL5 Rattus norvegicus*). Also, it may refer to different protein entities within the same model organism (e.g., *IL-5* for *interleukin 5* ***precursor*** or *interleukin 5* ***receptor*** of *Homo sapiens*). A term disambiguation module resolves the ambiguity and associates the term to a unique identifier.

Our TI system addresses both tasks. Specifically, the TI system approaches the first challenge by a rule-based fuzzy matching algorithm. For the second task, we studied several solutions and compared their performances. The best approach utilises a machine learning species-tagger, which was trained on human-annotated data and then automatically assigns a model organism to a protein mention. If the mention is still ambiguous, a heuristic rule is applied to resolve the remaining ambiguity. Experimental results show that our best term identification system achieved an F1 score that exceeded 85% of the inter-annotator agreement (IAA).

This paper is organised as follows: Section 2 provides an overview on related work on TI. Section 3 describes the data and the protein database that we have worked on. We also explain the evaluation metrics for measuring inner-annotator agreement and our system. Section 4 details our solutions to term identification where we tackle both term normalisation and term disambiguation. We emphasise one approach that integrates a species tagger to help resolve ambiguity, as it performed best in our evaluation. We finally draw conclusions and propose future research directions in Section 5.

2 Related Work

The identification of terminology in the biomedical literature is one of the most challenging research topics in the last few years both in Natural Language Processing and in biomedical research communities. Krauthammer and Nenadic [1] provides an excellent overview to the task and state-of-the-art solutions to it. They summarise three main steps to successful identification of terms from literature: term recognition, term classification and term mapping. As the names

[1] Our experiments focus on protein entities, but our techniques should be applicable to other biological entities such as genes or mRNAs.

[2] Our TI system is designed for term identification rather than term recognition. We used a separate Named Entity Recognition system to generate a list of protein mentions for our system to identify.

suggest, term recognition "picks up" single or several adjacent words that indicate the presence of domain concepts; term classification categorises the terms into biomedical classes, such as proteins, genes or mRNAs; and term mapping links terms to well-defined concepts in referent data sources, such as controlled vocabularies or databases. The first two steps are normally covered by Named Entity Recognition, which has been relatively better studied. The third step is essentially term identification, which is arguably more challenging because it involves resolving language ambiguity, where simple pattern matching and machine learning approaches are often not adequate.

Chen et al [5] collected gene information from 21 organisms and quantified naming ambiguities within species, across species, with English words and with medical terms. Their study shows that intra-species ambiguity in gene names was negligible at 0.02%, whereas across-species ambiguity was high at 14.2%. It suggests that resolving species ambiguity is an effective step towards gene name identification. Fang et al [6] reported their identification system based on automatically built synonym dictionaries and string matching techniques. However, their system restricts itself to identification of only human genes.

Recently, the BioCreAtIvE workshop [7] task 1B provided an excellent forum for research in term identification. Participating systems were required to produce lists of gene identifiers for occurrences of genes and gene products, in three model organisms (Yeast, Fly and Mouse), mentioned in sets of biomedical abstracts. Most systems [8–14] presented in the workshop followed a three-step procedure of term recognition, approximate search in lexicon and term disambiguation. However, they are different in details and a wide range of rule-based and machine learning techniques were applied.

Note that the BioCreAtIvE task and other previous work are different from ours in two ways. First, most of them identify gene names, whereas our task requires protein term identification, which is in general equally important for biomedical text mining applications. In specific applications such as extraction of protein-protein interactions, identification of protein names is even more important. In addition, protein name identification could be more challenging, as researchers observed that protein names tend to be more ambiguous [15] than genes, because protein names a) are inclined to contain multiple words than gene names and b) their naming convention is more diverse.

Second, in the BioCreative 1B task, the gene names to identify were species specific.[3] According to our experience and reports in previous work [5, 8], this largely reduces their ambiguity and made the task easier. Our term identification system, on the other hand, tackles protein terms across multiple species, which is more likely to happen in real world text mining applications, where species of biological entities are often not explicitly expressed in biomedical articles.

[3] Some researchers did use results from species identification as a feature to help perform species-specific term normalisation (e.g., [8]), although systematical study on species identification has not been reported.

3 Data and Ontology

Our TI system is a hybrid of rule-based and machine learning techniques, some of which require a protein database and manually annotated data. We used a commercial protein ontology, the *Cognia Molecular* (CM), as our referent protein database. It is derived from an early version of *RefSeq*[4] and similar to *RefSeq*, it comprises of protein records covering many species. The TI system assigns unique CM identifiers to ambiguous terms in texts.

We then hired a group of biologists and asked them to manually assign CM IDs to mentions of proteins in a collection of 584 biomedical articles taken from *PubMed Central*.[5] The TI annotation[6] involves linking a protein mention in text to a unique CM ID, where the annotators were asked to resolve any lexical ambiguity that might exist, based on contextual information and CM. They were also advised to pay attention to the species that a protein mention belongs to during the manual identification.

When the annotation process finished, we split the annotated data into 3 folds: training data (64%), development test (devtest) data (16%) and blind test data (20%)[7]. We analysed the manually annotated training data as follows:

1. Correct normalisation (24.3%): Terms are linked to their unique identifiers in CM.
2. Unknown (1.63%): Identification of these protein mentions could not be determined, and therefore were not assigned CM identifiers.
3. Not available in the ontology (2.48%): The protein mentions and their species could be identified but they were not included in CM;
4. Species overriding (68.5%): The annotators recognised the protein names and found them in CM, but they could not find the correct species for them; in which case they were advised to assign CM IDs of the same proteins but in *Homo sapiens* to the mentions and then assign the correct species to it.
5. Experimental proteins but not real proteins (3%) were not normalised.
6. Protein complexes (0.05%) were not normalised.

In the experiments reported in this paper, we only made use of the portion of the data that were correctly normalised (ie., category (1)), because essentially, only protein mentions in this portion can be correctly identified with respect to CM ontology. We noticed that the majority of the data belong to the "species overriding" category, which might be due to incompleteness of CM.[8] It also reflects the fact that protein mentions in biomedical articles belong to a wide range

[4] See http://www.ncbi.nlm.nih.gov/RefSeq/.

[5] See http://www.pubmedcentral.nih.gov/. The collection of papers used were a combination of abstracts and full-length papers.

[6] The annotation process was aiming to provide high-quality data not only for TI, but also for other text mining systems such as Named Entity Recognition and Relationship Extraction.

[7] Training data were used to train machine learning systems, which then tune their parameters on the devtest data. Evaluation was carried out on blind test data, which were unseen by the machine learn system, and therefore would reflect an unbiased performance.

[8] The CM ontology contains proteins across 22 species.

of species, which further confirms our observation that a species identifier would be very important for real world text mining systems.

We had 5% of the training data double-annotated for calculation of inter-annotator agreement (IAA).[9] In detail, we arbitrarily took one annotation as gold standard and the second as system output, and calculated F1 score for the second annotation.[10] The IAA on this task is 69.55%, which we think is reaonable, given the fact that the IAA on the task of English Word Sense Disambiguation is only about 67.0% [16], where native speaking annotators were asked to disambiguate the meaning of uses of common polysemous English words such as *interest*.[11] We measure the performance of TI in the same way with precision, recall and F1, which are then compared to IAA.

4 Hybrid Approachs to TI

The target of our TI system is to associate a CM ID to every mention of a protein in a document. In general, we approached the target following the two-step procedure of term normalisation and term disambiguation. We utilise a rule-based term normaliser which matches protein mentions in text to entries in CM. If there is a match, then the CM ID of the entry is assigned to the mention. Having multiple matches indicates that the protein mention in question is ambiguous, in which case the term disambiguation module is invoked. We have experimented with a few disambiguation methods and the best performing one takes advantage of a machine-learning-based species tagger and a heuristic rule.

More specifically, our final TI system repeats the following steps until all protein mentions are identified:

1. Associate candidate identifiers to a protein mention, by performing an approximate search in CM. If a single candidate is returned, then the protein mention is monosemous and assign this identifier to it; otherwise, go to Step (2).
2. Identify the species of the protein mention, using an automatic species tagger. Then compare the predicted species to the species associated with the candidate identifiers and filter out all identifiers whose species do not match the predicted one. If there is only one candidate left, assign it to the protein mention; otherwise, go to Step (3).
3. Apply a heuristic rule to rank the remaining candidate identifiers and assign the top-ranked one to the protein mention;

The first step (term normalisation) is described in Section 4.1. Steps 2 and 3 together perform term disambiguation and are detailed in Section 4.2. The same section also describes other disambiguation approaches that we tried but performed less well.

[9] Due to constraints on time and resources, we only had 5% data doubly annotated.

[10] F1 score is $\frac{2 \times precision \times recall}{precision + recall}$, where *precision* is the number of correctly identified terms divided by the total number of terms identified, and *recall* is the number of correctly identified terms divided by the total number of identified terms in the gold standard dataset.

[11] The word *interest* can be used for the "excitement of feeling" sense, or the "fee paid for use of money" sense, among others, according to context.

4.1 Assigning Potential CM Identifiers to Protein Mentions

The first step of TI is to assign one or many potential CM identifiers to a protein mention. It is achieved by looking up the CM ontology and matching the protein mention in text, to its potential "synonyms" in CM. The CM IDs of its synonyms are then assigned to the protein mention in question, as its candidate identifiers. Note that this is not a task of exact string matching, because, as we mentioned, names of proteins occur in articles in a variety of forms, including orthographic variants, abbreviations, acronyms, etc., which may not be the same as what they appear to be in CM.

We devised a set of rules for this matching process, based on our observations and previous work in literature [8]. Rules are divided into two sets. The first set were used to expand CM ontology: they were applied to every entry and the generated terms were added to CM. This resulted in an enriched CM with $186,863$ entries, in contrast to the original one with $153,997$ entries. The rules are:

1. Lowercase the item;
2. Remove/Add space between w and x, eg. "TEG 27" $\Rightarrow$ "TEG27";
3. Remove/Add hyphen between w and x, eg. "TEG-27" $\Rightarrow$ "TEG27";
4. Replace space between w and x with hyphen and vice versa, eg. "TEG 27" $\Rightarrow$ "TEG-27";

Where w denotes a token with multiple letters, and $x \in \mathcal{D} \cup \mathcal{L} \cup \mathcal{G}$, where $\mathcal{D}$ are tokens containing digits only, $\mathcal{L}$ are tokens containing a single letter only and $\mathcal{G}$ denote the set of English spelling equivalents to Greek letters (eg. alpha, beta, etc). We can see that this set of rules are employed to capture the orthographic variants of protein mentions.

The other set of rules are applied to the protein mentions on-the-fly during term identification. Each rule generates a variant of the mention and then it is used to query the enriched ontology. The matched entries are then retrieved. Repeat this process until all the rules are attempted. Note that these rules are ordered: if there are more-than-one matches, the matches are ranked according to the order of the rules that generated them. In detail, search the enriched CM using the following queries:

1. The original term as in text.
2. Lowercased form of the term.
3. The abbreviation/definition form of the term, acquired by searching a list of pairs of definition and abbreviation/acronym extracted from the document being processed, using an algorithm developed by Schwartz and Hearst [17].
4. If a word starts with a lower-case letter, followed by an upper-case letter, remove the preceding lower-case letter (eg. "hTAK1" $\Rightarrow$ "TAK1").

The rationale of the last rule is that the preceding small letter might be added by the authors to denote species of a protein mention, whereas the ontology may only contain the original form of the protein without the species indicating prefix.

At the end of this step, for each protein mention appearing in the text, one or many CM identifiers are retrieved from the expanded CM ontology. If a mention has only one match, the matching ID is assigned to it. Otherwise, proceed to the next step where term disambiguation is carried out.

4.2 Term Disambiguation

For every protein mention, the term disambiguation module selects a unique identifier from the pool of candidates generated for this mention in the previous step. We experimented with four disambiguation systems. We first describe the approach that performed best in our evaluation and then the alternatives.

Disambiguation with Help from Species Tagging As mentioned, knowing the host species of a protein mention can largely reduce its ambiguity. Therefore, we split the disambiguation task into two stages: we first predict its species to reduce the "cross-species" ambiguity. If a mention still maps to multiple identifiers, we resolve the "intra-species" ambiguity using a heuristic rule.

Species tagging can be treated as a text classification problem: a species tagger attempts to classify the piece of context surrounding a protein mention to the predefined categories of species, where a context is often represented by a set of features [18]. Following this idea, we developed two species taggers. The first one is rule-based. We first compile a list of 'species' words, which indicate specific species. For example, *mouse* is a 'species' word indicating *mus_musculus*, and *Escherichia coli* is a 'species' term for *escherichia_coli*. Intuitively, if a 'species' word appears in nearby context, a protein mention can be assumed to belong to the species that this 'species' word indicates[12].

The second species tagger uses the Support Vector Machines (SVM) classifier,[13] whose idea is to map the set of training data into a higher dimensional feature space $\mathcal{F}$ via a mapping function ϕ and then construct a separating hyperplane with maximum margin. Recall that the protein mentions in our manually annotated data are linked to their CM IDs, which are species specific. Therefore, they can be used as training data for our SVM based species tagger. The features we used are contextual word lemmas within a window size of 50 around the target protein entity, where the lemmas are TFIDF weighted. Table 1 shows 10-fold cross-validation performances of our machine learning and rule-based species taggers, respectively. The machine learning approach outperformed rule-based approach by 6.6% on average, and therefore we adopted the SVM based species tagger in our final system.

It is possible that protein names are still ambiguous within the same species, in which case we use a heuristic rule to resolve the remaining ambiguity. After species tagging, if a protein mention (p) still maps to multiple candidate identifiers, we use an algorithm to score every occurrence of a candidate identifier and then the scores for the same identifiers are accumulated. The identifier bearing the highest accumulated score is then assigned to the protein mention.

More formally, suppose our approximate matching algorithm retrieved n synonyms for a protein mention p, from CM. Let's denote the set of synonyms as

[12] This rarely happens but, when two 'species' words appear in equal distance at the left-hand side and the right-hand side, we assign the protein mention the species indicated by the 'species' word on the left.

[13] We use the Weka implementation of this machine-learning algorithm. See: http://www.cs.waikato.ac.nz/~ml/weka/

Table 1. Comparison of performance on species-tagging, with machine learning (ML) or Rule-based (R) species taggers (ST). All figures are in percentage (%).

Experiments	1	2	3	4	5	6	7	8	9	10	avg
ML-ST	41.0	69.5	66.4	53.9	47.8	36.8	48.6	68.8	71.9	55.0	**56.0**
R-ST	50.2	40.6	64.2	67.4	52.1	22.0	44.8	49.3	35.6	67.5	**49.4**

$S = \{s_1, s_2, ..., s_i, ...s_n\}$, where each synonym s_i maps to a set of CM identifiers: $ID_{s_i} = \{id_{s_{i1}}, id_{s_{i2}}, ..., id_{s_{ij}}, ...id_{s_{im}}\}$. m is the number of identifiers that a synonym s_i has. Therefore, $\mathcal{ID} = \bigcup_{i=1}^{i=n} ID_{s_i}$ is the set of candidate identifiers that p may link to. Note an identifier in $\mathcal{ID}$ may occur in multiple ID_{s_i} sets. An occurrence of id_i ($i \in [1, |\mathcal{ID}|]$) in ID_{s_i} is scored in a way that, if it is the lowest numbered identifier in ID_{s_i}, we assign it a score 3; otherwise we assign it a score 1. This weighting rewards the lowest numbered identifier in an arbitrary set ID_{s_i}. Then scores for all occurrences of id_i are accumulated. Repeat this procedure for every id_i in $\mathcal{ID}$, where $i \in [1, |\mathcal{ID}|]$, and the identifier id_i that bears the highest accumulated score is assigned to the protein mention p.

The heuristic behind the weight assignment (ie., weight 3 to the lowest numbered ID and 1 to others) is that CM IDs are formed with an uppercase P and digits (e.g., *P00678045*). We observed that the lower numbered IDs tend to occur more often than the higher numbered ones, and therefore the lower numbered IDs are more likely to become the correct identifiers for a protein mention. The next section describes another disambiguation method which empirically proved this observation.

Other Disambiguation Methods We also implemented three other disambiguation methods. First, as a baseline approach, we assign to a protein mention an arbitrary identifier taken from the pool of candidate identifiers associated to it. The second method is also straightforward. As we mentioned, we observed that the CM IDs are formed with an uppercase P and digits (e.g., *P00678045*). We sort the candidate IDs in numerical order with respect to the numerical part in the IDs and then assign the lowest numbered ID to the protein mention. If this system outperforms the first one, it means that the ordering of CM IDs are not arbitrary and lower numbered IDs are more likely to be the correct identifiers.

We applied a Vector Space Model (VSM) in the third system. In detail, in order to disambiguate a protein mention (p), we represent the textual context that p appears in as a vector of N word features, which we call a 'context' vector, where each feature has an 1 or 0 value to indicate occurrence or absence of a non-functional word. Similarly , we build n 'definition' vectors for all of the candidate identifiers, where 'definition' means description (ie., synonyms, species, etc) of a candidate identifier in CM. The 'context' vector is then compared

to the 'definition' vectors using the cosine similarity measure.[14] The identifier with a 'definition' vector that is most similar to the 'context' vector is assigned to the protein mention.

Table 2. Performance (%) of the four disambiguation systems in TI as evaluated on devtest data, ranked by F1.

System	Precision	Recall	F1
Species tagging+Heuristic ranking	64.1	55.5	59.5
Lowest id	52.1	46.4	49.1
VSM	48.9	43.5	46.0
Random id	47.3	41.1	44.1

The performance of the 4 systems are compared as shown in Table 2. The first system with a species-tagger is in lead for a large margin. Interestingly, the second system that selects the lowest CM ID significantly outperformed the one that assigns random ID. This indicates that the lowest numbered ID in a mention's candidate ID set has more chances to be its identification. This heuristic is used in the first system and empirically worked. The third system that compares 'context' vectors and 'description' vectors did not perform as well as we expected. One of the reasons might be that the glosses of identifiers in CM are too short, which causes the 'definition' vectors too sparse to be representative. There are two possible solutions that we could try in the future: one is to use a better ontology that have more extensive descriptions for its protein entries, and the other is to use smoothing techniques to alleviate the data sparseness problem.

4.3 Results

The best result of TI was achieved by combining machine-learning based species tagging and rule-based disambiguation. Table 3 shows the precision, recall and F1 of our system, as evaluated on devtest data and blind test data,[15] together with IAA. Recall that IAA indicates the performance by human experts on the same task. Our TI system has achieved a very promising performance that exceeded 85% of IAA. Also note that the machine-learning species tagger achieved an

[14] Cosine similarity: $corr(\boldsymbol{v}, \boldsymbol{w}) = \frac{\sum_{i=1}^{N} \boldsymbol{v_i w_i}}{\sqrt{\sum_{i=1}^{N} \boldsymbol{v_i}^2 \sum_{i=1}^{N} \boldsymbol{w_i}^2}}$, where $\boldsymbol{v}$ and $\boldsymbol{w}$ are vectors and N is the dimension of the vector space.

[15] Evaluation on blind test data was carried out independently by a third-party organisation who only evaluated the TI system as a whole. This explains why the performance of species tagging on the blind test data is unknown.

accuracy of 75.60% on the development test data, which is much higher than its performance of 10-fold cross-validation on the training data.[16]

Table 3. TI performance on Devtest data. ST denotes 'species tagging'. All figures are in percentage (%).

Dataset	IAA	ST Accuracy	Precision	Recall	F1	% to IAA
devtest	69.55	75.60	64.14	55.51	**59.51**	85.56
test	69.55	-	65.10	56.42	**60.44**	86.90

5 Conclusions

Our TI system automatically links mentions of proteins in biomedical texts to IDs in a referent protein database. It achieves this in two steps of term normalisation and term disambiguation. The first step involves collection of all potential IDs that can be associated to the mention in question, using fuzzy-matching rules. This approximate searching process found corresponding entries to protein mentions in our devtest data 86.53% of the time. It is highly possible that multiple CM IDS are retrieved for a single protein mention (over 73% cases, as estimated on devtest data). Our disambiguation module resolves the ambiguity by using machine-learning species tagging and a heuristic rule.

One of the distinctive features of our system is that it integrates assignment of the species as an indispensable part, which makes it capable of tackling identification of protein mentions across a number of species. Experimental results have shown our TI system achieved promising results. Note that our species tagger can also be used independently in text mining systems that require identification of model organism,

We carried out our work using a commercial protein database and manually annotated data. In the future, we will investigate the possibility of using publicly available protein databases, such as *RefSeq*. We will also study the feasibility of training the species tagger using automatically acquired training data, hence to make a completely unsupervised system.

References

1. Krauthammer, M., Nenadic, G.: Term identification in the biomedical literature. Journal of Biomedical Informatics (Special Issue on Named Entity Recogntion in Biomedicine) **37(6)** (2004) 512–526

[16] The evaluation was performed by an independent third party. Therefore the test data was unaccessable to us and we could only give the overall score of the TI system but not evaluate the species-tagger, which is a subsystem to TI.

2. Hirschman, L., Morgan, A.A., Yeh, A.S.: Rutabaga by any other name: extracting biological names. J Biomed Inform **35(4)** (2002) 247–259
3. Tuason, O., Chen, L., Liu, H., Blake, J.A., Friedman, C.: Biological nomenclature: A source of lexical knowledge and ambiguity. In: Proceedings of Pac Symp Biocomput. (2004) 238–249
4. Nenadic, G., Ananiadou, S., McNaught, J.: Enhancing automatic term recognition through term variation. In: Proceedings of 20th Int. Conference on Computational Linguistics (Coling 2004), Geneva, Switzerland (2004)
5. Chen, L., Liu, H., Friedman, C.: Gene name ambiguity of eukaryotic nomenclatures. Bioinformatics (2005) 248–256
6. Fang, H., Murphy, K., Jin, Y., Kim, J.S., White, P.S.: Human gene name normalization using text matching with automatically extracted synonym dictionaries. In: Proceedings of BioNLP'06, New York, USA (2006)
7. Hirschman, L., Colosimo, M., Morgan, A., Columbe, J., Yeh, A.: Task 1B: Gene list task BioCreAtIve workshop. In: BioCreative: Critical Assessment for Information Extraction in Biology. (2004)
8. Hanisch, D., Fundel, K., Mevissen, H.T., Zimmer, R., Fluck, J.: ProMiner: Organism-specific protein name detection using approximate string matching. BMC Bioinformatics **6(Suppl 1):S14** (2005)
9. Crim, J., McDonald, R., Pereira, F.: Automatically annotating documents with normalized gene lists. BMC Bioinformatics **6(Suppl 1):S13** (2005)
10. Fundel, K., Güttler, D., Zimmer, R., Apostolakis, J.: A simple approach for protein name identification: prospects and limits. BMC Bioinformatics **6(Suppl 1):S15** (2005)
11. Tamames, J.: Text detective: A rule-based system for gene annotation. BMC Bioinformatics **6(Suppl 1):S10** (2005)
12. Hackey, B., Nguyen, H., Nissim, M., Alex, B., Grover, C.: Grounding gene mentions with respect to gene database idntifiers. In: BioCreAtIvE Workshop Handouts. (2004) Granada, Spain.
13. Liu, H.: BioTagger: A biological entity tagging system. In: BioCreAtIvE Workshop Handouts. (2004) Granada, Spain.
14. Morgan, A., Hirschman, L., Colosimo, M., Yeh, A., Colombe, J.: Gene name identification and normalization using a model organism database. J Biomedical Informatics **37** (2004) 396–410
15. Hanisch, D., Fluck, J., Mevissen, H., Zimmer, R.: Playing biology's name game: identifying protein names in scientific text. Pac Symp Biocomput **403-14** (2003)
16. Mihalcea, R., Chklovski, T., Killgariff, A.: The Senseval-3 English lexical sample task. In: Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3). (2004)
17. Schwartz, A., Hearst, M.: A simople algorithm for identifying abbreviation definitions in biomedical texts. In: Proceedings of the Pacific Symposium on Biocomputing. (2003)
18. Ghanem, M., Guo, Y., Lodhi, H., Zhang, Y.: Automatic scientific text classification using local patterns: KDD Cup 2002. In: ACM SIGKDD Explorations Newsletter. Volume 4(2). (2003) 95–96