Integrating knowledge extracted from biomedical literature: normalization and evidence statements for interactions

Graciela Gonzalez[1] graciela.gonzalez@asu.edu

Luis Tari[2] luis.tari@asu.edu

Anthony Gitter[2] anthony.gitter@asu.edu

Robert Leaman[2] bob.leaman@asu.edu

Shawn Nikkila[2] shawn.nikkila@asu.edu

Ryan Wendt[2] ryan.wendt@asu.edu

Amanda Zeigler[2] amanda.zeigler@asu.edu

Chitta Baral[2] chitta@asu.edu

[1] Department of Biomedical Informatics, [2] Department of Computer Science and Eng, School of Computing and Informatics, Fulton School of Engineering, Arizona State University, Tempe, AZ 85281 USA,

Abstract

This paper reports our approach to three specific tasks of the BioCreAtIvE II challenge: protein interaction sentences (PPI-ISS), protein interaction pairs (PPI-IPS) and gene normalization (GN). Our approach to software engineering and implementation decisions was based on addressing first and foremost the core problem of integrating knowledge extracted from the literature: thus, we saw PPI-ISS as pairing statements of certain characteristics with core facts extracted elsewhere in the document and GN as mapping extracted entities to some standard names. This allows us to focus on generic solutions that can then be gradually refined to solving specific problems. In this same spirit, we developed a text-extraction XML format, a query language for the extraction of information constructs from a parse tree, a prototype extraction system, and a prototype web-based generic evaluation system that were then adapted to BioCreAtIvE. Our approach to the three tasks as well as analysis of results and a brief description of the related technologies developed are included in this report.

Keywords: normalization, protein-protein extraction, NLP, ranking, evaluation, data mining

1 Introduction

Numerous efforts to extract and annotate data from biomedical articles have resulted in over 200 databases and other resources [1] that allow scientists to access (in most cases, free of charge) structured biological information. However, it is estimated that between 300,000 and 500,000 [2] articles are added each year to the millions already in PubMed. The constantly increasing number of articles and the complexity inherent to its annotation results in data sources that are continuously outdated. For example, GeneRIF (Gene Reference Into Function), was started in 2002, yet it covers only about 1.7% of all the genes in Entrez [3] and 25% of human genes.

Automatic extraction and annotation seems a natural way to overcome the limitations of manual curation, and a lot of work has been done in this area, including the automatic extraction of genes and gene products [4], protein-protein interactions [5-9], relationships between genes or proteins and biological functions[10], genes and diseases[11-13], and genes and drugs[14], among others. However, the reliability of the extracted information varies greatly, and thus discourages the biologists from using it for their research.

The BioCreAtIvE II challenge with its different tasks addresses core areas in automatic extraction from biological texts: gene mention, gene normalization, and protein-protein interaction extraction. A particularly challenging aspect of the later is that only interactions that were supported by evidence of experimental methods in the same article were of interest[1]. The KDD Cup 2002 Information Extraction challenge [15] was

[1] Quoting from the 1st paragraph of the IPS Evaluation Process readme file, "… interaction pairs were only annotated by the database curators from the full text articles of the test set in case there was an experimental confirmation for this interaction mentioned in the article."

among the first to propose extracting interactions accompanied by sentences describing the experimental evidences. The logic behind this requirement is very important and often overlooked by PPI extraction systems: in practice, only interactions which are confirmed using experimental techniques are useful for high quality interaction annotations for biologists, and such sentences are often used by human curators as a deciding factor when annotating protein-protein interactions from text. Two of the most important manually annotated PPI databases, IntAct [16] and MINT [17], use this criteria and include the sentences in their databases. Usually, automated interaction extraction systems [5-9, 18] deploy techniques to determine if sentences are about interactions, but do not particularly address the more semantically refined concept of whether the given sentences provide *evidence* of the interaction. We hypothesize that the disparity in performance of the systems participating in BioCreAtIvE with respect to what is reported in the literature for PPI extraction systems (for example, reaching 92% f-measure in [18]) can be attributed in part to this requirement, as well as to the fact that such reported performance measures might in reality not be comparable, given the disparity in evaluation methods and gold standards used to generate them.

This paper reports our approach to three specific tasks of the BioCreAtIvE II challenge: interaction support statements (PPI-ISS), protein-protein interaction extraction (PPI-IPS) and gene normalization (GN) that share a number of pre and post processing techniques. Our approach to software engineering and implementation decisions was based on addressing the core problem of integrating knowledge extracted from the literature: thus, we saw PPI-ISS as pairing statements of certain characteristics to core facts extracted elsewhere in the document and GN as mapping extracted entities to some standard names. This allows us to focus on generic solutions that can then be refined to specific problems. Such refinements include, for example, the use of specific ontologies (like the MeSH category "Investigative Techniques" for locating evidence statements) and filtering and ranking techniques (like those applied to extracted interactions to find the most likely true positives).

2 Method and Results

2.1 Protein Interaction Sentences (PPI-ISS)

In this section, we describe our approach for the PPI-ISS task to extract passages that contain experimental confirmation for extracted. The system takes as input extracted protein-protein interactions and their corresponding PubMed ids, and outputs a ranked list of passages which describe the experimental evidence for the interactions. As part of the requirement of the PPI-ISS task, a maximum of 5 passages per interaction is returned and each passage cannot be longer than 3 sentences.

2.1.1 PPI-ISS Architecture

The system architecture for passage extraction is illustrated in Figure 1. The system uses Lucene [19] to index an XML version of the articles, which are converted in-house from the BioCreAtIvE HTML corpus. For each interaction extracted by our extraction systems (described in Section 2.2), a query is formed to retrieve potentially relevant paragraphs from the corresponding article. Passages are then extracted from within the relevant paragraphs. Each passage is scored based on the proteins and experimental methods they contain, to produce a final ranked list of passages. The details of each of the major components follows.

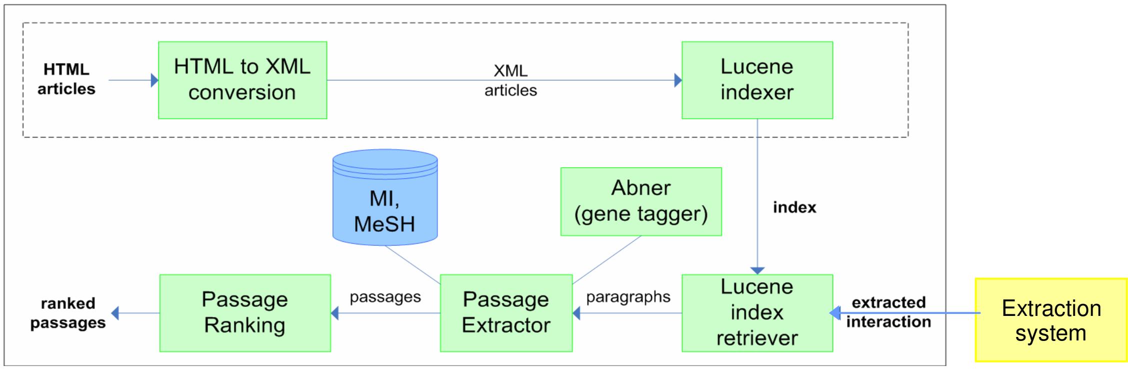

Figure 1. System architecture for extracting passages with experimental evidences. All articles are pre-processed by converting them to XML and indexing the resulting files in Lucene. Given an interaction pair, a Lucene query results in relevant paragraphs from the corresponding article. The extraction systems used are considered separately (Section 2.2).

*Retrieval of relevant paragraphs***.** We used the BioCreAtIvE HTML corpus as our initial input, converting it into an XML format. Each paragraph is stored in the XML format as an element of generic sections which include abstract, introduction, methods, results, conclusion, references and captions. Note that not all articles explicitly title their sections as such, so the mapping of paragraphs to sections is done using a heuristic algorithm. The XML file moves through the different system components as a universal input/output format, since all relevant information is added to it. For example, the acronym resolution algorithm described in [20] is run on the whole article, and occurrences of the acronyms are stored as elements in the XML file.

The XML articles are indexed using Lucene [19]. Given an interaction pair, a Lucene query returns paragraphs that have mentions of both of the entities in the interaction, and the section to which they belong. All relevant paragraphs are processed to extract valid passages, as detailed next.

Extraction of passages. The passage extraction component takes an interaction of interest and the relevant paragraphs as input, and produces a ranked list of passages as output. A *passage* is defined as a contiguous list of up to 3 sentences. To find passages, the sentences in the relevant paragraph are scanned and its genes and proteins are tagged using ABNER [21] (trained based on BioCreAtIvE I corpus). A sentence with one or both of the interactors serves as *seed* for a passage. If relevant keywords are found in the neighboring sentences, they are added to the passage. Keywords of interest include the protein interactors and terms associated with experimental evidence.

To recognize experimental methods within a passage, a dictionary of stemmed experimental method terms was compiled from the Molecular Interaction ontology (MI) [22] and MeSH terms under the categories “Investigative Techniques”, “Diagnosis” and “Therapeutics”. In each of the sentences in the passages, words are stemmed using the Porter stemmer [23] and exact string-matching is used in for recognizing them.

A passage is *valid* if it includes both of the proteins in the interaction. Valid passages are scored based on two criteria: (1) origin of the passages, and (2) frequency of terms of interest. The intuitive basis for criteria (1) is that experimental evidence for protein-protein interactions is usually mentioned in the methods and/or results sections as well as in captions more often than in other sections. Thus, a passage p_i that originated from one of these sections is scored higher, as follows:

$$score\_origin(p_i) = \begin{cases} 1 & \text{if } p_i \text{ is originated from method, results, captions of an article} \\ 0.5 & \text{if } p_i \text{ is originated from abstract, introduction, conclusion of an article} \\ 0 & \text{if } p_i \text{ is originated from the referencessection of an article} \end{cases}$$

Criteria (2) is based on the number of experimental methods and gene/protein names of interest appearing in the passages. Let $freq(p_i)$ be the number of occurrences of experimental methods and gene/protein names of interest (interactors and their synonyms) in passage p_i, where $p_1, \ldots, p_n$ are valid passages extracted from an article. Let $F = \{ freq(p_1), \ldots, freq(p_n) \}$. Then criteria (2) is computed as follows:

$$score\_evidence(p_i) = freq(p_i) \,/\, \max F$$

The final score of passage p_i is the sum of $score\_origin(p_i)$ and $score\_evidence(p_i)$. This single score is associated with each valid passage. The top 5 passages from all relevant paragraphs are returned.

2.1.2 PPI-ISS Analysis

We submitted 3 runs for the BioCreAtIvE PPI-ISS task, each one resulting from identical processing of a different input set of interactions. Thus, the same approach was used to extract passages for the 3 runs, but the extracted interactions were obtained from different runs of our PPI-IPS task, as described in Section 2.2. The results of each of the runs are presented in Table 1. Some passages were judged as false positives when in fact the passages could be alternative to the passages used in the gold standard for evaluation, as noted by the BioCreAtIvE organizers in the readme file of the ISS subtask. The inclusion of such alternative statements will impact our performance positively by reducing the number of false positives.

Table 1. PPI-ISS results. Different sets of interactions were obtained from different runs of our PPI-IPS task. The “Mean” column represents the average performance of all of the BioCreAtIvE PPI-ISS runs.

	Mean	Run 1	Run 2	Run 3
Fraction correct (best) from predicted passages	0.0473	0.0514	0.0483	0.0605
Fraction correct (best) from unique passages	0.0473	0.0496	0.0456	0.0533
Mean reciprocal rank of correct passages	0.5574	0.5731	0.5813	0.5476

We further analyzed 35 out of the 169 true positive passages with respect to their paragraphs of origin. A total of 26 out of the 35 originated from the results section, while 7 passages were from figure captions. This suggests that the intuition behind criteria (1) of passage scoring is reasonable. For criteria (2), the length of the passages was not considered so that it gives higher preferences to long passages over short passages.

Recall that paragraphs stored in the XML format are not necessarily assigned to the actual sections in the original format (due to variations in the section names). The deficiencies of the conversion can affect the scoring of the passages, since scoring is partly based on their origin. To get an approximation of the performance impact of the conversion step, we quantified the converted articles that were incorrectly converted into XML as follows: if (1) there was no text in any of the sections or (2) there were fewer than 5 paragraphs in the references section, the article was flagged as incorrectly converted. Either condition points to a conversion error, since all paragraphs should belong to a section, and articles usually cite more than 5 papers. Of the 358 articles provided as the PPI testing dataset, 48 of the converted articles failed the first condition, and 82 failed the second, indicating a potential "infiltration" of references as regular paragraphs.

Thorough quantification of these problems and their impact in the overall performance of the system is ongoing. Other limitations of our approach reflect the categories identified in [24] as common challenges: (a) discriminating the polarity of passages (b) evaluating the certainty of passages, briefly discussed next.

Discriminating the polarity of passages. Our current approach cannot distinguish if interactions are confirmed or not from the extracted passages. Consider for example the following sentence from PMID 16234233, which should not have been provided as evidence of an interaction:

> Passage 1: *"We have not been able to confirm the specificity of the commercially available antibodies against ASIC3 on DRG tissues isolated from ASIC3-inactivated mice."*

Evaluating the certainty of passages. Some of the passages extracted by our system are mere speculation of hypotheses, and should not have been regarded as correct evidence passages. Consider the following sentences extracted from PMID 16278218:

> Passage 2: *"Forced expression of MAPKAP kinase 2 (MK2) appears to lead to phosphorylation of free Heat shock transcription factor 1 (HSF1) on serine 121, and this is associated with HSP90 binding and inhibition of heat shock elements (HSE) binding."*
>
> Passage 3: *"We have shown that MAPKAP kinase 2 (MK2) directly phosphorylates Heat shock transcription factor 1 (HSF1) and inhibits activity by decreasing its ability to bind the heat shock elements (HSE) found in the promoters of target genes encoding the HSP molecular chaperones and cytokine genes."*

A human reader can easily distinguish the "we have shown" in Passage 3 as much stronger than the "appears to lead" in Passage 2, but the distinction is not obvious using the scoring criteria of our system.

2.2 Protein Interaction Pairs (PPI-IPS)

The PPI-IPS runs by our group were completed using two natural language processing (NLP) extraction systems, IntEx [25] and Phoenix, that differ in their extraction method but share a number of pre- and post-processing techniques. For both, each paragraph in the source article is broken into individual sentences, which are processed individually. Each sentence is first cleaned by the Jericho HTML Parser [26] that transforms HTML character references into the corresponding ASCII characters. ABNER [21] is then used to identify protein name mentions in the sentence. If at least two protein names and an interaction word from the IEPA corpus [7] are detected, the sentence is parsed by Link Grammar[27], a deep syntactic parser that generates constituent trees and grammatical linkages between words. The differences in the architecture are detailed next, followed by an analysis of our results in this task.

2.2.1 PPI-IPS Architecture

IntEx uses complex combinations of Link Grammar[27] word-to-word linkages to identify subjects -*S*-, objects -*O*-, verbs -*V*-, and modifiers -*M*- in a sentence, and extracts interactions based on patterns of these roles. IntEx has been described in detail in [25]. Phoenix, still under development, is our follow-up system. The main motivating factors for writing a new system were flexibility and extensibility: Phoenix is modular in design, and will be easy to upgrade and fine-tune.

Extracting triplets of interest. An ad-hoc query language was developed to express the rules that detect syntactic roles of words in parsed sentences. The extraction rules use the constituent tree representation provided by Link Grammar to detect subjects, verbs phrases, and objects in each clause of the sentence (rather than the word-to-word linkages used by IntEx). Using the constituent trees facilitates the construction of potentially useful grammatical combinations that result in triplets of the form <subject, verb phrase, object>. These are then filtered to include only protein-protein triplets of interest. As seen in the sample rules in Figure 2, the extraction rules examine the relationships (child, descendent, or sibling) between tree nodes and are used to match patterns of constituents in the tree.

Selecting triplets. In both Phoenix and IntEx, the subject and object are first normalized to their UniProt identifiers using the algorithm described in Section 2.3, attempting to map them first to the most common organisms (humans, yeast, and mouse). If a high-confidence match is not found, then the entire list of UniProt identifiers provided by BioCreAtIvE is used. The triplet-filtering step also uses a list of protein types [5] to strip the type from subjects and objects to prepare protein names for normalization.

Once interactions have been normalized, all the triplets produced by IntEx are used in the final output, whereas Phoenix filters them as follows:

- Remove interactions where both entities are identical
- Keep only one copy of interactions detected multiple times in the same sentence
- Score interactions based on different factors, such as the section where it appears, the number of times the entities and the interaction itself appear, and the confidence level of the normalization step.

The interactions are then sorted by their scores, which are used to decide which interactions to include in Phoenix's output. High precision runs can be created by only outputting interactions with a score greater than a certain threshold.

Evaluation. To aid in our development, we modified our existing prototype web-based evaluation system to support the BioCreAtIvE IPS and ISS submission formats, adding features to aid in rapid evaluation of Phoenix. Like many simple evaluation scripts, the online evaluation system automatically calculates the precision and recall of an uploaded run based on a set of gold standard facts. In addition, it allows a document by document view of each interaction, with the system score for each, plus the source sentence and the extracted protein names before normalization. Throughout development, we could quickly locate incorrectly extracted facts and identify the general source of the error by examining how the protein names were normalized and the grammatical structure of the source sentence. This significantly reduced the time required to assess the effects of changes to the extraction algorithm and was a great aid in determining which areas of Phoenix required improvement.

2.2.2 PPI-IPS Analysis

For the first run, we used Phoenix and tuned the interaction score threshold to try to optimize the *f*-score of the extracted interactions. The second run was also Phoenix, but with a lower threshold to generate more interactions. These interactions were then post-processed to leave only those for which supporting

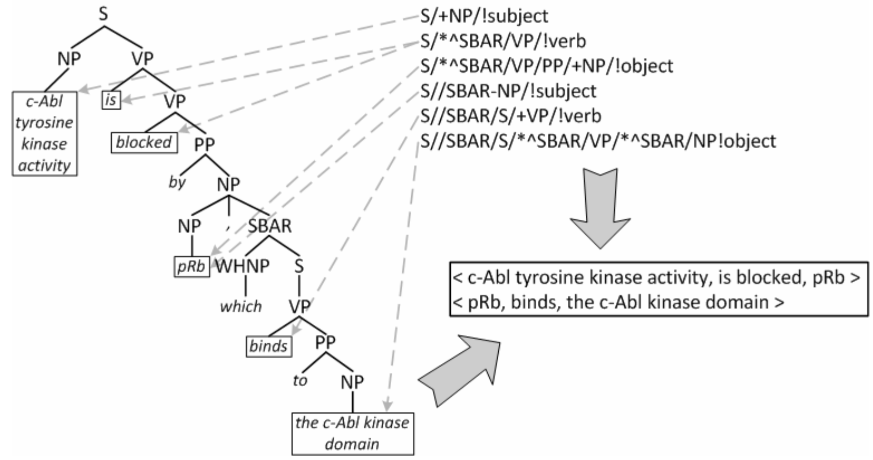
S
NP
VP
c-Abl tyrosine kinase activity
is
VP
blocked
PP
by
NP
NP
,
SBAR
pRb
WHNP
S
which
VP
binds
PP
to
NP
the c-Abl kinase domain
S/+NP/!subject
S/*^SBAR/VP/!verb
S/*^SBAR/VP/PP/+NP/!object
S//SBAR-NP/!subject
S//SBAR/S/+VP/!verb
S//SBAR/S/*^SBAR/VP/*^SBAR/NP!object
< c-Abl tyrosine kinase activity, is blocked, pRb >
< pRb, binds, the c-Abl kinase domain >

Figure 2. A partial list of extraction rules expressed in the ad-hoc grammar query language developed for our NLP text extraction system (Phoenix), as they apply to a Link Grammar constituent tree.

experimental evidence was found (using the output from the PPI-ISS subtask, described in Section 2.1). Interactions without supporting passages indicating the experimental techniques were pruned. Run 3 was extracted by IntEx without any experimental evidence post-processing. As seen in Table 2, the official BioCreAtIvE evaluation results of our submission, Runs 1 and 3 outperformed Run 2, with Run 3 as the best run overall. Although Run 3 (IntEx) did slightly better, the difference with respect to Run 1 (Phoenix high precision) is not statistically significant, having an effect size of less than 0.02 (negligible).

Table 2. Official scores by run

	Run 1 Phoenix	Run 2 Phoenix	Run 3 IntEx
Mean Precision	0.0456	0.020172	0.056049
Mean Recall	0.124279	0.099706	0.136227
Mean F-score	0.055964	0.029517	0.068575
Overall Precision	0.036957	0.020233	0.052997
Overall Recall	0.080189	0.069575	0.071934
Overall F-score	0.050595	0.03135	0.061031

Protein name normalization was a significant source of error across all three runs. Even with a flawless NLP extraction technique, Table 3 gives the BioCreAtIvE evaluation of the normalization of our predicted interactor proteins. This data shows that even if all pre-normalization extraction modules hypothetically performed flawlessly, our extraction systems' results would still be limited by our ability to map protein name mentions to UniProt[28] identifiers. We further discuss this problem in Section 3.

The blind two-tiered approach used for normalization within this task, where normalization to common organisms is done first, proved problematic. It helped give greater weight to the most common cases, but it introduced errors in others. For example, in one case, a correctly extracted interaction pair was normalized to human proteins instead of yeast, even though his article's title alone, "The Cap-binding protein eIF4E promotes folding of a functional domain of yeast translation initiation factor eIF4G1", shows that IF4E, IF4G1 should be mapped to yeast proteins. Thus, contextual clues need to be examined when selecting the correct organism.

Phoenix relied on ABNER [21] for protein name mentions for sentence classification and triplet filtering. Using the model trained on the BioCreAtIvE corpus, which is what was used, ABNER reports 65.9% recall. Therefore, assuming independence of protein name recognition and ignoring the possibility that a false positive is identified, there is a 56.6% (100% – 65.9% * 65.9%) chance that the sentence will be ignored because both protein names in are not recognized. In addition, a single false positive from ABNER could cause multiple false positives in the extracted interaction pairs if the incorrect protein name was present in multiple interaction pairs.

We traced most of the remaining errors to Link Grammar and the rules used to extract interaction pairs from its constituent tree output. At the time of submission, Link Grammar split multiword protein names when building a constituent tree. This made normalization of the interaction pairs much more difficult, but has since been corrected. Moreover, Link Grammar produces many possible linkages and constituent trees for each sentence, but the first linkage and constituent tree returned by Link Grammar was always used for the extraction. Upon manual examination, it was found that the first linkage and tree returned were not always the best representation of the sentence structure. In addition, much of the information to be gained by using a deep parse instead of a shallow POS tagging was not exploited. In Phoenix, subjects, verb phrases, and objects were grouped into sets and combined based on the clause of the sentence that contained them, rather than the tree structure. The rules themselves covered only the most general sentence constructs, which led Phoenix to overlook protein interactions expressed in less common grammatical forms. These issues are presently being addressed in the refinement of the Phoenix extraction system.

2.3 Gene Normalization

The gene normalization system we implemented was a lightweight implementation which mixed well-known systems with the implementation of new, relatively nonstandard, ideas. Overall, the system relied heavily on orthographic and syntactic information rather than semantic knowledge, including biological domain knowledge. Its architecture and analysis of results follow.

2.3.1 Architecture

The Gene Normalization Task receives as input an abstract to process and produces a list of normalized gene mentions from the text. The system completes 4 distinct execution phases: extraction, filtering, normalization

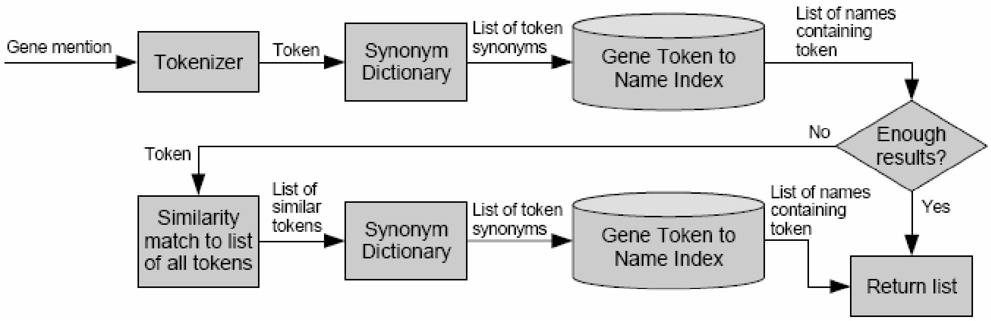

Figure 4. Gene normalization system. Gene mentions are compared first as a complete instance, and then at the token (word) level if not enough matches are found.

and disambiguation, with most of the complexity residing in the normalization phase. There, each gene mention is tokenized and compared against the standard gene names and a similarity score is computed for each. A list of the most similar standard gene names is then returned. We describe details of each phase next.

Extraction. We intended the system to primarily test gene normalization ideas and therefore employed the same ABNER [21] system for tagging gene mentions in each abstract, and as for the other tasks, used the model trained on the BioCreAtIvE 1a task. After gene mentions are tagged and extracted, acronyms are resolved using the Stanford Biomedical Abbreviation database, described in [29], and their provided Java code. The list of gene mentions found is the only data passed from the abstract to the next phase.

Filtering. In the filtering phase, mentions of generic words (such as “gene” and “protein”) are dropped. Specifically, gene mentions which consist entirely of generic words are removed; all other mentions are retained. The list of generic words contains about 100 entries of the following types:

- Organism names such as “yeast”, “human”, and “E. coli”
- General protein types and descriptors like “enzyme”, “amyloid”, and “protein”
- Other terms related to molecular biology, but not gene names, such as “DNA” or “alpha”

Normalization. Each gene mention which passes filtering is capitalized and separated into tokens. The system then compares the mention with each of the standard gene names and computes a similarity score for each comparison. This score is based on the Dice coefficient [30], and therefore reflects the number of tokens contained in both the gene mention and the standard gene name, scaled to reflect the lengths of both, and gives twice the weight to agreements. A perfect match has a similarity score of 1.0 while the similarity score for an attempted match with no tokens in common is 0. The equation for the standard Dice coefficient is $dice(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|}$.

The standard calculation was modified in the following ways:

- Each token was given a weight based on the frequency with which it appears in the list of gene names. Tokens appearing more frequently have a lower weight than tokens appearing less frequently, according to the following function $w(x) = 1 - \left(\frac{f(x)}{a \times m}\right)^{\frac{1}{a}}$, where f(x) is the frequency of the token in the list of gene names, m is the maximum frequency of any token and a is an empirically-determined tuning parameter greater than 1. Note that for any token x, 0 ≤ w(x) ≤ 1. This weighting scheme was designed to decrease more slowly than simply using the inverse of the token frequency.
- The Dice coefficient is further modified to give tokens from the gene mention a higher weight than tokens from the gene name. This reflects the fact that the gene mentions have, on average, fewer tokens than the standard gene names.

These modifications result in the equation $dice_w(X,Y) = \frac{2 \times \sum_{z \in X \cap Y} w(z)}{a \times \sum_{x \in X} w(X) + (1-a) \times \sum_{y \in Y} w(y)}$.

Tokens are initially considered a match if they contain exactly the same series of characters or represent synonymous ordinal values, such as Arabic and Roman numerals and the letters of the Greek alphabet.

To boost precision, thresholding is applied so that matches with a low score are dropped from further consideration. A list of candidate gene names taken from the top matches is then associated with each gene mention as it moves into the disambiguation phase.

*Disambiguation***.** Since the normalization phase returns a set of candidate gene names from the standard list, it is necessary to determine which of the candidates is the most likely to be correct. Disambiguation proceeds in a short series of automated steps based on simple rules as follows:

1. Gene mentions where the similarity margin – the difference between the similarity of the best match and the similarity of the second best match – is above a threshold are considered unambiguous. For these, the genes to which the best-matching gene name refers are added to the final output. The margin threshold used is preset and was determined empirically using the training set.
2. Gene mentions which remain after step 1 are reviewed to determine if their list of potential matches contains a name which refers to a gene already accepted as unambiguous. The intuition is that the abstract is most likely referring to the same gene by different names. The gene mention is removed.
3. Finally, for any remaining gene mentions, the best-matching gene name is accepted and the gene to which it refers is added to the final output.

2.3.2 Gene Normalization Analysis

The system achieved a recall of 0.713 and a precision of 0.520 on the test set, for an f-measure of 0.602. We believe that these results demonstrate that metric-based methods are insufficient, even when coupled with orthographic similarity between two tokens. Table 3 shows the evaluation of several variants of the system, showing the respective contribution of the various phases.

Table 3. Adjusted performance measures on GN system variations.

Variation	Precision	Recall	F-Measure
As evaluated for the competition	0.462	0.667	0.546
Without filtering phase	0.440	0.670	0.531
Standard Dice coefficient instead of weighted	0.461	0.669	0.546
No threshold-based removal of low similarity matches	0.339	0.713	0.460
Return best match instead of using disambiguation rules	0.439	0.692	0.537

Using acronym resolution to substitute the original text of the gene mention introduces a problem when the standard gene names also contain abbreviations.

The simple disambiguation rules used to eliminate generic mentions perform reasonably well in practice, and their failures are generally due to failures in the normalization to correctly identify semantic equivalence. However, the current method of relying on a small dictionary is brittle and ought to be based on a wide sampling of molecular biology terms. A more flexible method may be to perform filtering after the normalization step by noting that generic mentions are going to match a wide variety of standard gene names at a low level of similarity, but match none of them well.

3 Discussion

Three important developments from our participation include the development of the overall architecture that allows a more flexible incorporation of the different components using a standard input/output XML format, the development of a new extraction system flexible enough to sustain generic extractions of relationships in biomedical text, and the development of a flexible evaluation platform. Given the reliance of the overall knowledge extraction and integration approach on solid gene mention and gene normalization modules, these two subsystems will occupy a good part of our efforts.

For the extraction of related statements (evidence of interaction being one of them), we will expand on the issues of polarity and certainty of passages, as they are critical to the problem of finding passages with experimental evidences.

As for the extraction system, future development will initially focus on improving the manner in which the extraction rules are used to identify potential interactions. The algorithm that combines the subjects, verbs, and objects will be modified to utilize the relationships between these syntactic roles by analyzing their common ancestors in the constituent tree. Furthermore, we have learned that organism identification is a nontrivial component of successful protein name normalization. Before normalizing, we will search for context clues regarding the organisms and provide this information to the normalization process.

References

[1] "Pathguide: The Pathway Resource List."

[2] E. S. Soteriades and M. E. Falagas, "Comparison of amount of biomedical research originating from the European Union and the United States," *BMJ: British Medical Journal.* , vol. 331 pp. 192-194, 2005.

[3] Z. Lu, K. B. Cohen, and L. Hunter, "Finding GeneRIFs via Gene ONtology Annotations," presented at Pacific Symposium on Biocomputing, Maui, Hawaii, USA, 2006.

[4] L. Tanabe and W. J. Wilbur, "Tagging gene and protein names in biomedical text," *Bioinformatics*, vol. 18, pp. 1124-1132, 2002.

[5] G. Leroy, Chen, H. , et al., "Genescene: biomedical text and data mining," presented at The third ACM/IEEE-CS joint conference on Digital libraries, 2003.

[6] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature," *Bioinformatics*, vol. 17, pp. 155 - 161, 2001.

[7] J. Ding, Berleant, D., Xu, J., Fulmer, A., "Extracting biochemical interactions from MEDLINE using a link grammar parser," *IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*, pp. 467, 2003.

[8] S. T. Ahmed, D. Chidambaram, H. Davulcu, and C. Baral, "IntEx: A Syntactic Role Driven Protein-Protein Interaction Extractor for Bio-Medical Text. ," in *BioLINK SIG: Linking Literature, Information and Knowledge for Biology, a Joint Meeting of The ISMB BioLINK Special Interest Group on Text Data Mining and The ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics (Biolink'2005).* Detroit, Michigan, 2005.

[9] T. M. Phuong, Lee, D., Lee, K. H., "Learning Rules to Extract Protein Interactions from Biomedical Text," *PAKDD 2003*, pp. 148-158, 2003.

[10] A. Koike, Y. Niwa, and T. Takagi, "Automatic extraction of gene/protein biological functions from biomedical text," *Bioinformatics*, vol. 21, pp. 1227-1236, 2005.

[11] H.-W. Chun, Y. Tsuruoka, J.-D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. i. Tsujii, "Extraction of Gene-Disease Relations from Medline Using Domain Dictionaries and Machine Learning," presented at Pacific Symposium on Biocomputing, 2006.

[12] C. Perez-Iratxeta, P. Bork, and M. Andrade, "Association of genes to genetically inherited diseases using data mining," *Nature Genetics*, vol. 31, pp. 316-319, 2002.

[13] D. Hristovski, B. Peterlin, J. Mitchell, and S. Humphrey, "Improving literature based discovery support by genetic knowledge integration," *Stud Health Technol Inform 2003*, vol. 95, pp. 68-73, 2003.

[14] T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter, "EDGAR: extraction of drugs, genes and relations from the biomedical literature," *Pac Symp Biocomput*, pp. 517 - 528, 2000.

[15] A. S. Yeh, L. Hirschman, and A. A. Morgan, "Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup," *Bioinformatics*, vol. 19, pp. i331-339, 2003.

[16] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler, "IntAct: an open source molecular interaction database," *Nucl. Acids Res.*, vol. 32, pp. D452-455, 2004.

[17] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: a Molecular INTeraction database," *FEBS Letters*, vol. 513, pp. 135-140, 2002.

[18] I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. Bader, K. Michalickova, T. Pawson, and C. Hogue, "PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine," *BMC Bioinformatics*, vol. 4, pp. 11, 2003.

[19] "Lucene."

[20] A. Schwartz and M. Hearst, "A simple algorithm for identifying abbreviation definitions in biomedical texts," *In Proceedings of the Pacific Symposium on Biocomputing (PSB 2003)*, vol. 8, pp. 451-462, 2003.

[21] B. Settles, "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text," *Bioinformatics*, vol. 21, pp. 3191-3192, 2005.

[22] "Molecular Interaction (MI) ontology."

[23] M. F. Porter, "An algorithm for suffix stripping.," *Pro-gam*, vol. 14, pp. 313--316, 1980.

[24] W. J. Wilbur, A. Rzhetsky, and H. Shatkay, "New directions in biomedical text annotation: definitions, guidelines and corpus construction," *BMC Bioinformatics*, vol. 7, pp. 356, 2006.

[25] S. Ahmed, D. Chidambaram, H. Davulcu, and C. Baral, "Intex: A syntactic role driven protein-protein interaction extractor for bio-medical text," *Proceedings ISMB/ACL Biolink*, pp. 54-61, 2005.

[26] "Jericho HTML Parser."

[27] D. Sleator and D. Temperley, "Parsing English with a Link Grammar," *Third International Workshop on Parsing Technologies*, 1993.

[28] "UniProt."

[29] J. D. Wren, J. T. Chang, J. Pustejovsky, E. Adar, H. R. Garner, and R. B. Altman, "Biomedical term mapping databases," *Nucl. Acids Res.*, vol. 33, pp. D289-293, 2005.

[30] L. Egghe and C. Michel, "Strong similarity measures for ordered sets of documents in information retrieval," *Information Processing and Management*, vol. 38, pp. 823-848, 2002.