

Information Extraction from Biomedical Literature: Methodology, Evaluation and an Application

L. Venkata Subramaniam, Sougata Mukherjea, Pankaj Kankar, Biplav Srivastava,
Vishal S. Batra, Pasumarti V. Kamesam, Ravi Kothari
IBM India Research Lab,
New Delhi, India
E-mail: (lvs,smukherj,kpankaj,sbiplav,bvishal,pvk,rkothari)@in.ibm.com

ABSTRACT

Journals and conference proceedings represent the dominant mechanisms of reporting new biomedical results. The unstructured nature of such publications makes it difficult to utilize data mining or automated knowledge discovery techniques. Annotation (or markup) of these unstructured documents represents the first step in making these documents machine analyzable. In this paper we first present a system called BioAnnotator for identifying and annotating biological terms in documents. BioAnnotator uses domain based dictionary look-up for recognizing known terms and a rule engine for discovering new terms. The combination and dictionary look-up and rules result in good performance (87% precision and 94% recall on the GENIA 1.1 corpus for extracting general biological terms based on an approximate matching criterion). To demonstrate the subsequent mining and knowledge discovery activities that are made feasible by BioAnnotator, we also present a system called MedSummarizer that uses the extracted terms to identify the common concepts in a given group of genes.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;
J.3.3 [Computer Applications]: Life and Medical Sciences—*Medical Information Systems*

General Terms

Design, Experimentation

Keywords

Information Extraction, Biological Document Processing,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.
Copyright 2003 ACM 1-58113-723-0/03/0011 ...\$5.00.

1. INTRODUCTION

Biomedical information is growing explosively and new and useful results are appearing everyday in research publications. Many of these publications are available online, for example, in PubMed's MedLine database[13]. However, automatic extraction of useful information from these online sources remains a challenge because these documents are unstructured and expressed in a natural language form. To enable data mining and knowledge discovery from such documents it is necessary to make this data available in a structured format. Because of the very large amounts of data being generated, it is difficult to have human curators extract all these information and present them in a form useful to researchers. Information Extraction (IE) from such sources is becoming crucial for the timely dissemination of information. However, information extraction in the biomedical domain is a challenging task. Some problems include, long multiword terms, inconsistent naming conventions, new coinages and the use of abbreviations and acronyms.

In this paper we present a system called **BioAnnotator** for identifying biological terms in scientific literature and annotating the terms with their semantic classes. BioAnnotator first identifies already known terms by doing a lookup on different dictionaries. It then tries to identify new and unknown terms by using character and word level properties of biological terms in addition to contextual clues. Evaluating BioAnnotator using a publicly available corpus indicates that the system can identify biological terms with 87% precision and 94% recall using an approximate matching criterion.

Extracting terms from biomedical literature is useful in various applications. For example, in the extraction of relations between biological entities (for example, protein-protein interactions), it is first necessary to recognize and classify the entities taking part in the interactions [20]. Term extraction is also useful for automatically updating biomedical databases like SwissProt[2] and KEGG[10]. At present these databases are largely hand curated.

To demonstrate the subsequent data mining and knowledge discovery that is possible with the output produced by BioAnnotator, we also present a system called *MedSummarizer* which uses the terms extracted from biomedical literature in a unique way. Analysis of data from Microarrays [16] produces a cluster of genes whose relation to each other is derived solely from the microarray experimental data. In order to explain the underlying biological mechanisms associated with a cluster of genes obtained by analytical meth-

ods, it is necessary to cross-reference the gene cluster with previously known biological facts and results. The MedSummarizer system is geared for this task of helping a biologist understand the common concepts or “biological meaning” in a gene cluster. It identifies the biological terms co-occurring with the given group of genes in MedLine papers. The system uses a Java Swing based user interface to visualize the similarity between the genes and show the biological terms associated with the group.

The paper is organized as follows. The next section cites related work. Section 3 explains the system architecture of BioAnnotator and how it extracts biological terms. Section 4 discusses our evaluation of the BioAnnotator. Section 5 discusses the MedSummarizer system followed by the conclusion in section 6.

2. RELATED WORK

The task of extracting biological terms from scientific documents can be considered similar to the *named entity* task in the Message Understanding Conference evaluation exercises [14]. Many biomedical information extraction methods thus represent adaptations of methods originally proposed for MUC [9].

Biological term extraction systems can be broadly divided into two types: those with a rule base and those with a learning method. In [5] protein names are identified in biological papers using hand-coded rules. A rule based approach combined with dictionary lookup for term recognition and classification is given in [6]. In [4] supervised learning methods based on Hidden Markov Models are used. In [1] statistical approaches based on word distributions in a large corpus are used to find biological terms. In [15] an Entropy-based approach combined with morphological rules is used for finding terms. [8] gives a very good overview of the field.

We use rules and dictionary lookup for identifying biological terms. Some of the previous rule based systems have tuned their rules for identifying a small class of terms. For example, [5] has created rules for only finding proteins. On the other hand, BioAnnotator tries to identify all possible biological terms. Moreover, the system is designed so that the rules can be easily modified to identify a different class of entities. Unlike most of the previous systems we have also evaluated BioAnnotator using a publicly available corpus. As stated in [8], good evaluation of the existing systems is one of the main challenges in this domain.

Our MedSummarizer system is similar to the system described in [18] that tries to determine gene similarity by determining co-occurrences of gene-names in MedLine abstracts. However, the system cannot determine why the genes are similar because they do not associate biological terms with the genes. Another similar system is MedMiner[19], which aims at providing summarized literature information on genes. However, MedMiner is limited to finding relations between two genes only and needs a relevant keyword list, requiring the user to have prior knowledge of the possible interactions between the two genes. [17] presents a system, which attempts to find functional relations among genes on genome-wide scale, but it requires the users to specify a representative document for each gene, which describes the gene very well. Looking for the representative document may need a lot of time, effort and knowledge on part of the user.

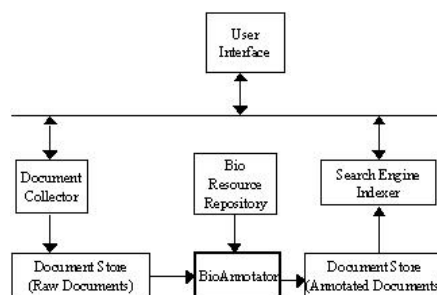


Figure 1: Unstructured Information Management Middleware Architecture

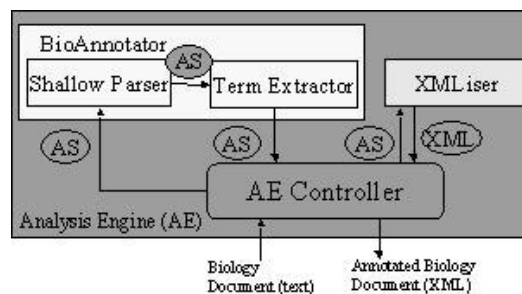


Figure 2: BioAnnotator: Component and Flow Details

3. BIOANNOTATOR

BioAnnotator takes raw documents and annotates the biological terms present in them. In this section we provide an overview of the system.

3.1 System Architecture

At IBM Research there is a division wide effort to define and implement an architecture called *Unstructured Information Management Middleware Architecture (UIMA)*. The goal of UIMA is to provide a common framework that allows an easy deployment and integration of text analysis tools that are developed under different platforms and programming models. The BioAnnotator system conforms to (and benefits from) this architecture and middleware.

The high level components of the UIMA framework are shown in Figure 1. The Document Collector fetches locally available documents or those from external data sources like PubMed and populates them in the Document Store. The Analysis Engine is a generic interface to house text analysis tools and integrate components developed on heterogeneous platforms. The BioAnnotator is deployed in the Analysis Engine to annotate biological documents. To analyze documents, the BioAnnotator uses the external resources deployed in the Bio-resource Repository. These resources include dictionaries, a rule base and stop word lists. These are described in sections 3.2 and 3.3. BioAnnotator writes the annotated documents to the Document Store which provides scalable persistence and efficient search over the documents.

Figure 2 shows the processing steps and component details inside the Analysis Engine. The Analysis Engine (AE) is submitted text documents for biological term annotation. The AE Controller initializes the Analysis Structure (AS) with the input document and processes the document with the Shallow Parser and the Term Extractor that are de-

ployed within the engine. The Shallow Parser identifies noun phrases in the document and uploads these annotations into AS. The AS is then routed to the Term Extractor for identifying biological terms.

The updated AS with the biological terms annotated is then passed on to the XML-iser that translates it to XML. The XML tag set consists of *BIO_CONCEPT* tags that mark out biological entities. In addition every *BIO_CONCEPT* is assigned features giving their baseform and class information. Baseform refers to the canonical form of the concept. For example, *caspase-3* has the baseform *CPP32 protein*. A biological concept can be referred to by various synonyms. For example, *caspase 3* is variously referred as *apopain*, *Yama protein*, *CPP32 protein*, etc. A consistent baseform tag allows us to recognize every reference to the biological concept even if it is called by different names. The class feature assigns each biological concept to its correct semantic class. For example, *caspase-3* has the class *Amino Acid, Peptide or Protein*. The complete annotation for this example looks as follows:

```
<BIO_CONCEPT>
  <baseform>CPP32 protein</baseform>
  <class>Amino Acid, Peptide, or Protein</class>
  caspase-3
</BIO_CONCEPT>
```

(In section 3.4 we describe how the class and baseform of a biological term is determined). The annotated documents are stored in Document Store and can be searched using a XML search engine [3].

3.2 Term Extraction - Dictionaries

The noun phrases that are identified by the shallow parser are examined by the Term Extractor to identify biological terms. The Term Extractor uses dictionaries and a rule engine. At present we are using 3 dictionaries:

- **Unified Medical Language System (UMLS)** [21]: UMLS is a consolidated repository of medical terms and their relationships, spread across multiple languages and disciplines (chemistry, biology, etc.), and is meant for ready use in knowledge-based systems. In the 2002AB release of UMLS there are approximately 875,255 concepts and 2.14 million concept names which are organized into 134 semantic classes and 54 semantic relationships. Each biological concept in UMLS is associated with semantic classes such as *Gene or Genome* and *Amino Acid, Peptide, or Protein*. Certain classes of UMLS do not refer to biological concepts; examples include *Geographic area* and *Governmental or Regulatory activity*. We have compiled an **UMLS stop class list** to prevent terms from these classes being extracted.
- **LocusLink** [12]: This is our primary knowledge source for gene names. It contains the list of genes of several organisms such as human, fruit fly, rat, cow etc. LocusLink contains over 80,000 distinct genes. Along with their aliases it contains over 200,000 gene names.
- **GeneAlias**: This is a locally compiled list of aliases for some of the gene names not present in LocusLink. In total, the Gene Alias list contains about 16,000 gene names.

More dictionaries can be added by specifying them in a configuration file. For each dictionary, the configuration file also indicates whether it is to be used for single-word terms, multi-word terms or both. For example, since most gene names are single word, it does not make sense to lookup multiword terms in *GeneAlias* or *LocusLink*.

The term extractor first looks up the entire noun phrase in the multi-word dictionaries. It then removes stop words from the beginning and end of the phrase using a stop word list. This list contains standard English stopwords as well as some additional words like *abstracts*, *title*, *pubmed* that occur frequently in the documents of interest. The stripped phrase is then sent for lookup in the multi-word dictionaries. If no match is found for the phrase, each single word from the phrase are sent for lookup in the single-word dictionaries. Note that the order in which the dictionaries are looked up are specified in the configuration file. It can also be specified that a dictionary should be looked up only after the rule engine fails.

3.3 Term Extraction - Rule Engine

We have seen that many biological concepts are missing from UMLS and the other knowledge sources that we use. It is important to recognize these terms as biological concepts also. For this purpose we use a rule engine. The rules of the engine are encoded in a rule base which is a XML file containing **regular expressions** and **regular expression groups**.

3.3.1 Regular Expressions

Though the naming of biological concepts does not follow any convention, many biological terms have some specific patterns. We use regular expressions to specify positive and negative examples of biological term patterns. Let us illustrate these with some examples¹:

- Many biological terms contain upper case letters, numerical figures and non-alphabetical characters (for example, *PTEN*, *c-N-ras*, *CD4-Positive T-Lymphocytes*). To identify these terms we use the following two regular expressions:

```
<RegularExpression name="DigitCapSpecialChar">
  [\p{Upper}\d\p{Punct}]
</RegularExpression>
<RegularExpression name="NotDigitCapSpecialChar">
  (^[\d\p{Punct}]+$)|(^[\p{Upper}?[\p{Lower}-]+$]
  |(^[\p{Alpha}\.-]+\.[\p{Alpha}\.-]+$)
</RegularExpression>
```

The first expression will match any word, which has upper case alphabets, numbers or special characters. The second regular expression can be used to filter out some non-biological words like:

- Words with only digits and special characters (for example, 10,000)
- Hyphenated words with only lower case alphabets (for example, *bio-informatics*)
- Words containing only alphabets and . (for example, *H.D.Smith*)

¹Our regular expressions follow the *java.util.regex* convention

- Words in which an upper case alphabet is followed only by lower-case alphabets (for example, the first word of a sentence or proper nouns).

- Terms that contain Greek words as part of the multi-word term (for example, *pancreatic alpha cells*, *beta-hemolysis*, *tau interferon*) are generally biologically relevant. These can be determined by using a regular expression that matches any Greek letter.
- Biological concept names often contain prefixes, suffixes and root forms that give an indication of their class [22]. For example, many proteins end with *ase* (for example *amylase*) and many cell names have *blast*, *cyt* or *phore* (for example *leucocytes*). Therefore, the following regular expressions are useful:

```
<RegularExpression name="ProteinSuffix">
  .+ase$
</RegularExpression>
<RegularExpression name="CellRoot">
  (?i)cyt|blast|phore|plast
</RegularExpression>
```

- In the documents many biological concept names are preceded or followed by keywords or signals that give an indication of their class (for example, *p16 tumor suppressor gene*, *pancreatic alpha cells*, *proteins Rac1 and Cdc42*, etc.). We have formed regular expressions for such signal words:

```
<RegularExpression name="CellSignal">
  (?i)apoptoti(s|c)|cell(s)?|clone(d|s)?|
  culture(d|s)?|neuron(s)?|strain(s)?
</RegularExpression>
<RegularExpression name="GeneSignal">
  (?i)mutate(d|s)?|(onco)?gene(s)?
</RegularExpression>
<RegularExpression name="ProteinSignal">
  (?i)amino|amine(s)?|enzyme(s)?|kinase(d|s)?|
  ligand(s)?|motif(s)?|peptid(e|ase)|
  protein(s|ase)?|protease
</RegularExpression>
```

- In order to filter out units like *124 nM* and *30 degrees* we use the following regular expression:

```
<RegularExpression name="Units">
  [0-9\.]+\.(nM|microM|UV|(?i)gram(s)?|
  min(ute)?s?|degree(s)?|bp|aa|kg)
</RegularExpression>
```

3.3.2 Regular Expression Groups

A Regular Expression group (*RegExpGroup*), as the name suggests, consists of a group of regular expressions. An example is shown below:

```
<RegExpGroup name="SingleWord">
  <RegExps positiveRE="CellSignal"
    label="Cell" score="1.2"/>
  <RegExps positiveRE="GeneSignal"
    label="Gene" score="1.2"/>
  <RegExps positiveRE="ProteinSignal"
    label="Protein" score="1.2"/>
```

```
<RegExps positiveRE="DigitCapSpecialChar"
  negativeRE="NotDigitCapSpecialChar"
  score="1.0"/>
```

```
.....
</RegExpGroup>
```

When a string is being matched against a Regular Expression Group, it is matched against the group's regular expressions in the order in which the expressions are arranged in the group. If the string matches the *positiveRE* and does not match the *negativeRE* of a pair, it matches the *RegExpGroup* with the score specified for the pair. Thus, since the string *Ca2+* matches *DigitCapSpecialChar* and does not match *NotDigitCapSpecialChar*, it matches the *RegExpGroup SingleWord* with a score of 1.0.

At present we use four Regular Expression Groups. The group *SingleWord* is used to determine single words that are biologically relevant. Some of the regular expressions in the group are shown in the example above. The *RegExpGroup MultipleWord* is used to determine phrases that are biologically relevant. At present we do not have expressions in the group. However, if an expert feels that certain important phrases are not getting extracted, she can add the required regular expressions to this *RegExpGroup*. The Regular Expression groups *NotSingleWord* and *NotMultipleWord* are used to provide negative patterns to filter out non-relevant biological terms. For example, the group *NotMultipleWord* contains the Regular Expression *Units*.

3.3.3 Rule Engine Methodology

The rule engine identifies biological terms as follows:

```
boolean ruleEngine(String phrase)
{
  if (matchRegExpGroup("NotMultiWord",phrase))
    return false;
  if (matchRegExpGroup("MultiWord",phrase))
    return true;

  double score = 0;
  foreach word in phrase
    if (!stopWord(word)) {
      score+=(matchRegExpGroup("NotSingleWord",word));
      score+=(matchRegExpGroup("SingleWord",word));
    }

  if (score >= threshold)
    return true;
  else return false;
}
```

The input to the rule engine is a noun phrase with all the leading and trailing stop words removed. The input phrase is first matched with the Regular Expression groups *MultiWord* and *NotMultiWord*. If there is no match, the individual words of the phrase are matched with the Regular Expression groups *SingleWord* and *NotSingleWord*. The scores of the individual words are accumulated to determine the final score of the phrase. If it is greater than the threshold specified in the configuration file, it is considered to be a biological term.

For example, the phrase *124 nM* will match the Regular Expression group *NotMultiWord* (which contains the Regular Expression *Units*). So the procedure returns *false*. On

	Precision	Recall	F-score
Approx	0.8652	0.9425	0.9022
Exact	0.6029	0.6859	0.6401

Table 1: Precision and Recall of the BioAnnotator

the other hand, the word *Ca2+* matches the Regular Expression group *SingleWord* and if the corresponding score 1.0 is greater than the threshold, *true* is returned. Now consider the phrase *protein-kinase proto oncogene*; *protein-kinase* is a protein signal and *oncogene* is a gene signal. So the overall score of the phrase is 2.4.

Note that all the biological knowledge is captured in the rule base XML file. A domain expert may easily modify this file to enable other patterns to be annotated as biological terms. Moreover, the rule engine can be tuned for other domains by writing appropriate regular expressions for handing single-word and multi-word terms without any change to the underlying code.

3.4 Term Classification

For each identified biological term we also try to determine its class. If a term is identified using UMLS, the UMLS semantic class and the UMLS specified baseform is associated with the term. If a term is identified using LocusLink, its semantic class is *Gene or Genome: LocusLink* and its baseform is the primary name of the gene. Gene names identified by GeneAlias are classified similarly.

If a phrase is identified using the rule engine, we try to guess its class. For example, if a word of the phrase matches the regular expressions for *GeneSignal*, *CellRoot* or *Protein-Suffix*, it is classified as *Gene*, *Cell* or *Protein* respectively. If two words in the phrase give different clues, the last matched regular expression gets preference. For example, the phrase *protein-kinase proto oncogene* has contextual clues for both protein and gene. Since the last match is for *GeneSignal*, it is classified as a gene. Note that if we identify a term because it only has words containing special characters, numbers or common biological prefixes or suffixes, we cannot guess its class.

4. BIOANNOTATOR EVALUATION

We did a formal evaluation of BioAnnotator’s term extraction using the publicly available GENIA 1.1 corpus [7]. This corpus has abstracts of 670 research papers as well as a list of the biological terms identified in them by human experts manually.

The BioAnnotator results are compared with the manual annotations. When a term from BioAnnotator is matched with a human annotated term, one can look for exact or approximate match. For exact match, the annotations from BioAnnotator and experts should match exactly. For approximate match, one of the annotations should be a substring of the other. Table 1 summarizes the results. Note that we have also shown *F-score*, which is the harmonic mean of precision and recall, to facilitate comparison. It is calculated as $\frac{2 * Precision * Recall}{Precision + Recall}$.

The table shows that for approximate match the precision and recall are 87% and 94% respectively while for exact match they are 60% and 69% respectively. The precision and recall of BioAnnotator are affected by a variety of reasons. On closer analysis of the results, we found that:

		Precision	Recall	F-score
NP & VG	Approx	0.8467	0.9499	0.8453
	Exact	0.5773	0.686	0.627
NP & NPP	Approx	0.8784	0.9505	0.913
	Exact	0.4433	0.5177	0.4776
Using LocusLink	Approx	0.8555	0.9554	0.9027
	Exact	0.5826	0.7084	0.6326
Only Dictionaries	Approx	0.8767	0.3975	0.547
	Exact	0.4458	0.2027	0.2787
No Dictionaries	Approx	0.8647	0.9208	0.8919
	Exact	0.6	0.6615	0.6293
Rule Engine threshold 1.5	Approx	0.9211	0.5399	0.5393
	Exact	0.6562	0.3813	0.4823
Rule Engine modification	Approx	0.8645	0.9207	0.8917
	Exact	0.4827	0.5328	0.5065

Table 2: Precision and Recall of the BioAnnotator in different configurations

- Experts differ on whether certain terms are biologically relevant in the context of a document. Examples include *temperature*, *pathway* and *regulation*. The BioAnnotator identified them as biological terms in all cases. However, the experts consider them to be biologically relevant in certain contexts only.
- Usually it is very hard to get an agreement even between two experts about the extent of a biological term. For example, for the phrase *human cancer tissue*, one expert may consider the whole phrase to be a biological term while another may only mark *cancer tissue* to be the biological term. Because of the differences in the extent of biological terms between the BioAnnotator and the human experts, our exact match precision and recall are much lower than the approximate match scores. However, in certain cases even the approximate match scores will be affected because of differences in the extent of the biological terms. For example, for the phrase *splice site pairing*, experts identified *splice site* as the biological term while the BioAnnotator had identified *site pairing*. Since one is not a subset of the other, we would have a false positive and a false negative using both the exact match and the approximate match criteria.
- Experts have their bias about how a term is written. We found variations between the actual terms in the document from which BioAnnotator identified its annotations and the way it was manually annotated. An example is *G(2)* and *G2*.
- Errors of the underlying shallow parser also affect the precision and recall. For example, for the sentence *Calcineurin acts in synergy with PMA*, the parser identified *Calcineurin acts* as a noun phrase. Therefore the rule engine marked both the words as a biological term whereas only the first word is biologically relevant.

4.1 Testing Different Configurations

The results shown in Table 1 were obtained when the Term Extractor identifies biological terms from noun phrases by using UMLS as the multi-word dictionary followed by the

rule engine followed by Gene Alias as the single-word dictionary. We have also tested with other configurations some of which are shown in Table 2. Note that the results are worse or at most comparable to the results shown in Table 1. Let us now discuss these other configurations.

4.1.1 Considering verb groups and noun prepositional phrases

If we consider verb groups in addition to noun phrases, the recall goes up marginally but the precision comes down. The F-score is lower.

If we consider noun prepositional phrases ($\langle NP \rangle$ $\langle Preposition \rangle$ $\langle NP \rangle$) in addition to noun phrases, although the approximate precision and recall both go up, the exact precision and recall go down dramatically. This indicates that the experts generally identify noun phrases as biological terms and may annotate the underlying noun phrases of a NPP as two biological terms. Note that if we consider a NPP, the underlying noun phrases are not examined by the Term Extractor. Therefore, if the underlying noun phrases are the biological terms and the Term Extractor incorrectly annotates the whole NPP, both the exact precision and recall will go down.

4.1.2 Experimenting with Dictionaries

If we use LocusLink in addition to GeneAlias for identifying gene names, the recall goes up but the precision goes down. The F-scores are comparable. Note that the precision is decreased because many of the gene names in Locus Link are also common English words (for example, *we*, *high*, *star*).

Only using the three dictionaries provides really bad recall scores for both approximate and exact match. This shows that the dictionaries that we have used are not comprehensive. Note that if a gene name has surrounding words that are biologically relevant, the experts will generally annotate all the words as a single biological entity. The rule engine may have been able to correctly identify the entity but Gene Alias and Locus Link will only annotate the gene name. The gene dictionaries may also annotate some common words as gene names. Because of these reasons, the exact precision also goes down by not using the rule engine.

Not using the dictionaries at all also reduces the recall. This shows that the rule engine cannot correctly identify some terms available in the dictionaries. Moreover, the dictionaries will do a much better job of classifying the terms. Note that the precision also goes down slightly by not using UMLS since the rule engine may annotate some non-relevant terms that would have been filtered out by the UMLS stop class list.

4.1.3 Experimenting with the Rule Engine

The results of Table 1 were obtained with a rule engine threshold of 1.0. Increasing it to 1.5 increases the precision but reduces the recall (as expected). The F-scores are much lower.

Suppose the rule engine is passed a phrase with words $w_1w_2w_3w_4$. Also suppose only the words w_2 and w_3 are biologically relevant (matches the *SingleWord* Regular Expression group). If the overall score of the two matches is greater than the threshold, the whole phrase is considered to be a biological term. Another alternative is to remove the leading and trailing non-relevant words from the biological

term. In that case only w_2w_3 will be considered to be the biological term. The last row of Table 2 shows the evaluation of the system using the modified Rule engine. Both the exact precision and recall has gone down. This seems to indicate the experts generally do not annotate subparts of a noun phrase as a biological term. (Therefore, it is essential that the shallow parser correctly identifies the noun phrases). Surprisingly, the approximate recall has also gone down. This seems to indicate that some words that we have excluded from phrases are actually biologically relevant.

5. MEDSUMMARIZER

We have developed a system called MedSummarizer that showcases an application of BioAnnotator. MedSummarizer uses the biological terms extracted from biomedical literature by BioAnnotator to produce a biologically relevant summary of a gene cluster. The input to the system is a group of genes. This group can be obtained from clustering of gene expression data or any other group of genes of interest to the user. The system follows a series of steps to create the summary of these genes and calculate the similarity between them.

5.1 Methodology

5.1.1 Information extraction

The first step is to retrieve the relevant documents for each of the genes from PubMed. For each gene in the gene list, a PubMed query is formulated using the given gene name and its aliases. Using the aliases prevents us from missing some relevant documents for the gene where it was not being referred by the name given by the user. For example, if the given gene name is *OLE1*, then the PubMed query will be *OLE1 OR MDM2 OR YGL055W*. The gene aliases are obtained from the gene dictionaries.

Once the relevant documents for the gene are obtained, they are submitted to the BioAnnotator for annotation. The annotated XML document returned by BioAnnotator is parsed to extract the baseforms of the biological terms.

Some of the biological terms, rather than being associated with the given cluster of genes, are associated with genes in general. We have created a MedSummarizer stop word list to remove these terms. We obtained the list by querying PubMed with a large random collection of genes and running BioAnnotator on the resulting documents. The biological terms with the highest frequency were added to the stop word list. A few of them, which were determined to be biologically important by a domain expert, were removed. The final list has about 40 terms; some examples are *Molecular Sequence Data* and *Amino Acid Sequence*.

5.1.2 Summarization

The system generates a summary of the given gene cluster by determining a set of important biological terms which gives an “optimal” summary of the biological properties and functions of the entire cluster. Just determining the terms with the highest frequency would not give good results. The core idea of our approach is that terms are considered relevant in describing a cluster if they fall into one of the following three categories:

- **Cluster Topics (Major):** These are terms that are commonly associated with almost all the genes in a cluster.

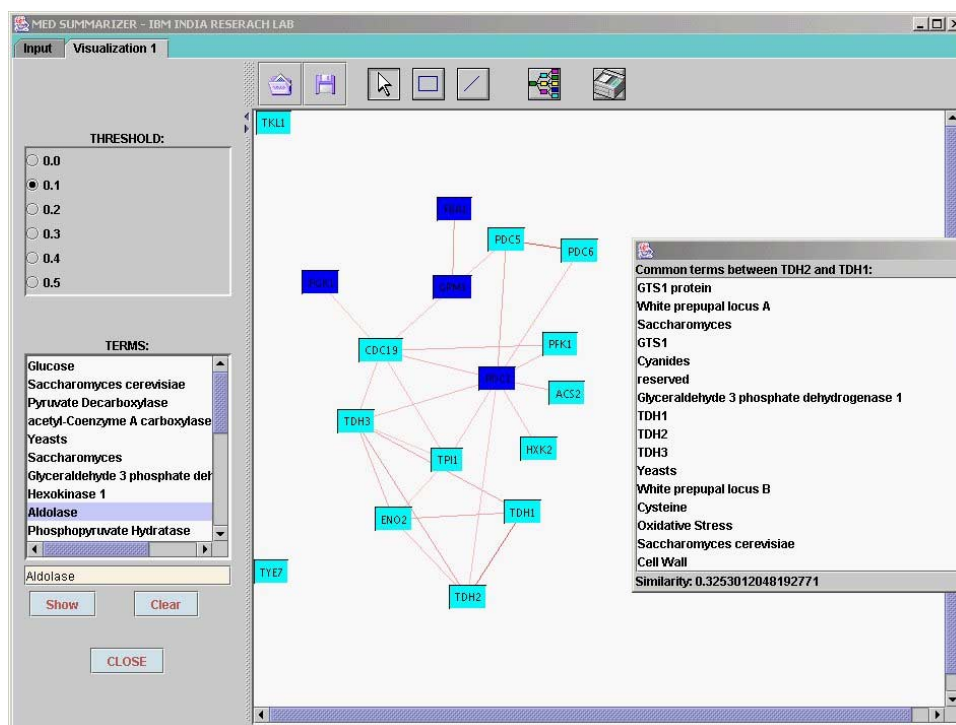


Figure 3: The MedSummarizer User Interface showing some genes involved in Glycolysis

- **Cluster Topics (Minor):** These are terms that are less common than the major topics but still appear with most of the genes and are considered important for the cluster as a whole.
- **Particular Topics:** There are terms that are not cluster topics but have *high particularity*, that is, are frequently associated with a few genes in the cluster. Many times they describe an important part of the cluster and help in deriving the complete functionality of the cluster.

We selected several statistical attributes to capture the required properties and then based on these statistical properties determined the set of terms providing the relevant summary. The details of the algorithm are provided in [11]. Note that the results reported in that paper uses the MeSH (Medical Subject Heading) keywords that are associated with the MedLine papers. Generally each paper is assigned only 10-15 MeSH keywords. Therefore, by augmenting the MeSH terms with the terms extracted by the BioAnnotator, we will generally get better results.

5.1.3 Determining Gene Similarity

Finally, the similarity between each pair of genes are determined from the biological terms common between them. Using the **Dice Coefficient**, the similarity between genes g_a and g_b is calculated to be $\frac{|a \cap b|}{|a| + |b|}$ where $|a|$ is the number of terms associated with g_a and $|a \cap b|$ is the number of biological terms common between the two genes.

5.2 User Interface

The user interface for the MedSummarizer system is shown in Figure 3. The input genes are shown as the nodes of a

graph. The input cluster for this example is a group of yeast genes encoding the enzymes involved in *glycolysis* (the process by which glucose is broken down in the cells of all higher animals). If the similarity between two genes is greater than the threshold an edge is drawn between the genes. The user can change the threshold as required (by default it is 0.1). The figure shows that all except two genes have similarity greater than 0.1 with at least one other gene. Obviously, if the threshold is increased, many of the edges will disappear. The brightness of an edge is proportional to the similarity between the two genes. Clicking on an edge shows the terms common among the genes. For example, the terms common between *TDH1* and *TDH2* are shown in the popup window.

The main terms associated with the given group of genes is shown in a scrolled list on the left. Thus, Figure 3 shows that terms like *glucose*, *yeast* and *hexokinase* are associated with the given set of genes. All of these terms are very relevant to the input gene cluster and will give a biologist an indication that these genes are involved in *glycolysis*. The user can select a term to see the genes that are associated with it. For example, in the figure the genes that are associated with *adolase* are highlighted.

6. CONCLUSION

In this paper we presented a biological term identification system which uses a variety of knowledge sources along with syntactic information, term properties and contextual clues to identify and classify known and new terms. BioAnnotator is implemented in a framework that allows easy deployment and integration of the components, and its rule engine can be modified to extract information from other domains. Our evaluation shows that the system has good precision and recall. Extensive evaluation of the system has also given

some insights on how experts identify biological terms in the literature.

We also presented a system called the MedSummarizer that uses the extracted terms to determine the common concepts in a cluster of gene. The user interface of the system allows the biologists to easily determine the similarity between the genes and see the biological terms associated with them. MedSummarizer represents an instance of how to use the output of BioAnnotator and we believe that the *semi-structured* documents that result from BioAnnotator provide opportunities for additional types of data mining and knowledge discovery from biomedical document corpuses.

Besides trying to further improve the precision and recall of the BioAnnotator, future work is planned along various directions:

- At present BioAnnotator can process about 10 documents per second on a 2GHz machine with 256Mb RAM. Since there are no dependencies among the documents being annotated, using Grid Computing to improve the performance further is an interesting research direction.
- At present all the regular expressions have to be manually determined. We also plan to use machine learning techniques to automatically learn the expressions based on an annotated corpus or biological ontologies.
- We have to improve our method of determining the semantic classes of the biological terms. Evaluation of our classification technique is also required.
- As stated before, many of the gene names are also common English words. An effective method of disambiguating between the two based on the surrounding context is required.

7. REFERENCES

- [1] M. Andrade and A. Valencia. Automatic extraction of keywords from Scientific text: Application to the knowledge domain of Protein families. *BioInformatics*, 4(7):600–607, 1998.
- [2] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence databank and its new supplement TrEMBL. *Nucleic Acids Research*, 25:31–36, 1997.
- [3] D. Carmel, E. Amitay, M. Hersovici, Y. Maarek, Y. Petruschka, and A. Soffer. Juru at TREC-10 experiments with index pruning. In *the Proceedings of the 10th Text Retrieval Conference*, pages 228–237, 2001.
- [4] N. Collier, C. Nobata, and J. Tsujii. Extracting the names of Genes and Gene products with a Hidden Markov Model. In *the Proceedings of the 18th International Conference on Computational Linguistics*, pages 201–207, Saarbrücken, Germany, 2000.
- [5] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward Information Extraction: Identifying Protein Names from Biological papers. In *the Proceedings of the Pacific Symposium on Biocomputing*, pages 707–718, Hawaii, 1998.
- [6] R. Gaizauskas, G. Demetriou, and K. Humphreys. Term Recognition and Classification in Biological Science journal articles. In *the Proceedings of the Computational Terminology for Medical and Biological Applications: Workshop of the 2nd International Conference on Natural Language Processing*, pages 37–44, Patras, Greece, 1998.
- [7] Genia Corpus. <http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/>.
- [8] L. Hirshman, J. Park, J. Tsujii, L. Wong, and C. Wu. Accomplishments and challenges in Literature Data Mining for Biology. *BioInformatics Review*, 18(12):1553–1561, 2002.
- [9] K. Humphreys, G. Demetriou, and R. Gaizauskas. Two applications of Information Extraction to Biological Science: Enzyme interactions and Protein structures. In *the Proceedings of the Pacific Symposium on Biocomputing*, pages 502–513, Hawaii, 2000.
- [10] M. Kanehisa et al. The KEGG database at GenomeNet. *Nucleic Acids Research*, 30:42–46, 2002.
- [11] P. Kankar, S. Adak, A. Sarkar, K. Murari, and G. Sharma. MedMeSH Summarizer: Text Mining for Gene Clusters. In *the Proceedings of the Second SIAM International Conference on Data Mining*, Arlington, VA, 2002.
- [12] Locuslink. <http://www.ncbi.nlm.nih.gov/locuslink/>.
- [13] MedLine. <http://www.ncbi.nlm.nih.gov/PubMed/>.
- [14] *Proceedings of the Sixth Message Understanding Conference*, Columbia, Md, 1995. Morgan Kaufman.
- [15] Y. Park. Identification of probable real words: An Entropy-based approach. In *the Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 1–8, Saarbrücken, Germany, 2002.
- [16] M. Schena, D. Shallon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complimentary DNA microarray. *Science*, 2700:467–470, 1995.
- [17] H. Shaktay, S. Edwards, J. Wilbur, and M. Boguski. Genes, Themes and Microarrays: Using Information Retrieval for large-scale Gene analysis. In *the Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 1999.
- [18] B. Stapley and G. Benoit. Bibliometrics: Information Retrieval and Visualization from Co-occurrences of Gene Names in Medline Abstracts. In *the Proceedings of the Pacific Symposium on Biocomputing*, pages 529–540, Hawaii, 2000.
- [19] L. Tanabe, U. Schref, L. Smith, and J. Lee et al. MedMiner: An Internet Text-Mining tool for Biomedical information with application to Gene Expression Profiling. *BioTechniques*, 27:1210–1217, 1999.
- [20] J. Thomas, D. Milnard, C. Ouzounis, S. Pulman, and M. Carroll. Automatic Extraction of Protein Interactions from Scientific Abstracts. In *the Proceedings of the Pacific Symposium on Biocomputing*, pages 541–551, Hawaii, 2000.
- [21] UMLS. <http://umlsks.nlm.nih.gov>.
- [22] P. Zweigenbaum and N. Grabar. A contribution of medical terminology to medical language processing resources: Experiments in morphological knowledge acquisition from thesauri. In *the Proceedings of the IMIA-WG6 Conference*, pages 131–141, Arizona, USA, 1999.