BioText Team Report for the TREC 2006 Genomics Track

Anna Divoli, Marti A. Hearst, Preslav I. Nakov, Ariel Schwartz
University of California, Berkeley
{divoli@ischool,hearst@ischool,nakov@cs,sariel@cs}.berkeley.edu

Alex Ksikes
University of Cambridge
alex.ksikes@gmail.com

1 Introduction

The paper reports on the work conducted by the BioText team at UC Berkeley for the TREC 2006 Genomics track.

Our approach had three main focal points: First, based on our successful results in the TREC 2003 Genomics track [1], we emphasized gene name recall. Second, given the structured nature of the Generic Topic Types (GTTs), we attempted to design queries that covered every part of the topics, including synonym expansion. Third, inspired by having access to the full text of documents, we experimented with identifying and weighting information depending on which section (Introduction, Results, etc.) it appeared in. Our emphasis on covering the different pieces of the query may have helped with the aspects ranking portion of the task, as we performed best on that evaluation measure.

We submitted three runs: Biotext1, BiotextWeb, and Biotext3. All runs were fully automatic. The Biotext1 run performed best, achieving MAP scores of .24 on aspects, .35 on documents, and .035 on passages.

2 Biotext1: Main Run

Our first run produced results that were reranked by the other two runs. We used the open source Lucene search engine (lucene.apache.org) for indexing, querying, and producing the initial ranking of the full text.

2.1 Pre-Processing of the Full Text

The pre-processing step stripped out all of the HTML markup, and identified the boundaries of the legal spans. Although we realized it might be important to convert the various Greek letters and other markup characters to plain text, we did not have time to do this conversion.

The pre-processing also attempted to identify the sections and their boundaries, using the markup in the section headings and defining a set of 16 different section types:

> *title, references, abstract, abbreviations, conclusions, results, introduction, methods, footnotes, acknowledgments, appendix, future, cases, grants, main, not-categorized*

We have not evaluated the accuracy of this analysis, and without training data it was not possible to assess how best to weight the different sections. As described below, we did make some attempts to weight spans depending on which section they came from.

2.2 Analyzing the Topic Description

As mentioned above, our main strategy was to try to maximize recall for gene and protein names, and to try to be sure that every content word in the query was represented in the retrieved documents.[1]

Thus, for each topic description, we identified the terms within it that might be a gene name, and developed a set of synonyms for those names. We made a distinction between strongly matched and weakly matched gene names, described below.

For each term that was not a gene name or a stop word, we attempted to match it against a term from MeSH, and produced a set of synonyms for that term as well.

Given a topic description, we created two sets of synonyms. S contained all possible variants of any gene names found in the topic description, using EntrezGene, UniProt and OMIM as resources. M contained the variants of any term in the topic description that was found to match a MeSH term.

For example, for a topic like "*How do HMG and HMGB1 interact in hepatitis?*", the code recognizes two gene names, *HMG* and *HMGB1*, and one MeSH term, *hepatitis*.[2]

[1]Since gene names and protein names are often interchangeable, below when we refer to *gene names* we implicitly mean *gene and/or protein names*.

[2]We developed a small number of topics of our own and used them during system development.

Looking up the corresponding variants in the different data sources yields the following sets of synonym candidates (here converted to lowercase):

S = {*ac2 008, clb, columbus, cg10367, dkfzp686a04236, dmhmg coar, fb23c02, hmgcr, hmg, hmg1, hmg2, hmg coar, hmg alpha, hmg coa reductase, hmg3, hmgb1, hmgcoar, hmgcoarr, mgc103168, mgc103169, mgc117896, mgc117897, mgc64255, mgc93598, mgc93599, nfd1, sbp 1, hbp1, hmg 11, hmg 12, hmg 3, hmg 4, hmg 5*}

M = {*hepatitis, hepatitides*}

Gene terms and MeSH labels were indexed in Lucene as separate fields and were queried using Lucene's fielded query facility. Below we describe in detail the gene name recognition and normalization, the MeSH term recognition and the query generation process.

2.3 Gene Name Recognition and Normalization

In our earlier Genomics Track work [1], we developed an in-house gene recognizer and normalizer tool. This original tool looked for gene names in raw text and mapped each identified instance to one or more possible corresponding unique identifiers from the then LocusLink (which has subsequently been superseded by EntrezGene). Multiple LocusLink identifiers can be found for a given gene name, because (a) the same gene has different IDs in different organisms, and (b) the gene name itself may be ambiguous, especially for short 3-4 character names.

The original tool used a set of normaliza-

tion and expansion rules in order to allow for some variations in form, including token rearrangement, and removal of whitespace, commas, parentheses, and numerals. All possible normalizations and expansions of all known LocusLink gene names and their synonyms were generated offline and then matched against a normalized version of the input text using an exact, first-longest-string-matching measure. The matches were then mapped back to the original unnormalized text, and the corresponding IDs were assigned.

For our participation this year, we significantly modified this tool. First, we made a clear separation between the normalization and the expansion rules. We further revised the expansion rules and split them into two subgroups: *strong rules* and *weak rules*, where the terms indicate the confidence that the resulting transformation reflects the original terms. The strong rules allow only minor changes such as:

- removal of white space
  (e.g., "*BCL 2*" → "*BCL2*")
- substitution of non-alpha-numerical characters with a space
  (e.g. "*BCL-2*" → "*BCL2*")
- concatenation of numbers to the preceding token (e.g., "*BCL 2*" → "*BCL2*").

The weak rules remove at least one alphanumeric token from the string. An example weak rule is the removal of trailing numbers e.g., "*BCL 2*" → "*BCL*".

As another example, treating a "/" as a disjunction produces two new strings:

"*aspartyl/asparaginyl beta-hydroxylase*" →
"*aspartyl beta-hydroxylase*" or
"*asparaginyl beta-hydroxylase*".

Another weak rule handles parenthesized expressions, removing text before, within and/or after the parentheses. For example,

"*mitogen-activated protein (MAP) kinase*" →
"*mitogen-activated protein MAP*", or
"*mitogen-activated protein kinase*", or
"*MAP kinase*", or
"*mitogen-activated protein*", or
"*MAP*", or
"*kinase*".

Unlike in the original tool, the new rules have no priorities and are applied in parallel and recursively, trying all feasible sequences. For each resulting expanded variant, we record the ID of the source gene synonym and whether a weak rule was used at least once in the derivation. For a given variant, there are multiple possible IDs, some of which used strong rules only and others that used at least one weak rule. The strong variants are meant to be very accurate and used for gene name matching, while the weak ones are suitable for query expansion, as they conflate related genes.

In our experiments, we downloaded and used the latest versions of EntrezGene, UniProt and OMIM. Because mapping among their IDs is complicated, we extracted the sets of expansion variants for each of them separately and applied each rule set in isolation.

In order to save time and storage space, we indexed information only about those genes that occurred in the TREC topics. We first ran the recognizer/normalizer over the topics and we then determined which of the IDs they mapped to for each database, using only strong transformation rules for the matching. Each gene

name found in a topic was assigned an identifier, which associated it with the corresponding IDs from the different databases (e.g., *bcl2* might be the identifier for BCL-2). We did the same separately for the variants found with the weak transformation rules.

2.4 Analysis of the Gene Transformation Strategy

Upon reflection, we think that the aggressive expansion of gene names may have negatively impacted our results.

For example, looking back at the example topic "*How do HMG and HMGB1 interact in hepatitis?*", we can see that the set S contains a number of terms that introduce noise. Some candidates, although listed in databases as synonyms of either *HMG* or *HMGB1*, come from organisms that cannot possibly be associated with *hepatitis* directly. For instance HMG2 is found in *Arabidopsis thaliana*, a plant, and fb23c02 is found in *Danio rerio* (zebrafish), a fish.

We should have addressed this issue by checking which organisms are associated with the terms in set M, and then used these organisms to filter out the unrelated terms from set S. We did something similar to this in the 2003 Genomics track. This time however, we decided to keep all gene homologs, since they often offer valuable information on the gene's function across all species, but that might have hurt more than helped.

We could also have used the various fields of EntrezGene, UniProt and OMIM more carefully. In S we can see the term *columbus*, which comes from the description field of UniProt for the entry *HBG2_HUMAN* and refers to a position variant. The whole description field is: *D → N (in Columbus-Ga) /FTId=VAR_003167.*

Since we were aiming for high recall, we did not consider problems generated by homonyms. For instance, SBP-1 (*Sterol regulatory element Binding Protein*) is listed in EntrezGene as an alias for HMGB1 (*High-Mobility Group Box 1*), but it also stands for *Sulfate-Binding Protein.*

2.5 Strong vs. Weak Mapping Rules

We ran the gene recognition tool against the entire text collection, using EntrezGene, LocusLink and OMIM, retaining only the matches corresponding to an ID that was recognized in some of the TREC topics. When a match was found, we included the corresponding identifier in the gene field for that document; the gene field could contain a bag of identifiers.

Each identifier name was prefixed with one of four codes: *ss, sw, ws,* or *ww.* The first letter corresponds to the gene found in the TREC topic, where *s* means it was derived by strong transformation rules, and *w* means weak rules. The second letter corresponds to what was found in the full text of the documents.

Suppose for example, that the TREC topic contained the string *BCL.* Strong rules map this only to *BCL*, but weak rules will also map it to *BCL 1* (numeral removal). The IDs corresponding to *BCL* will be considered strong IDs and the ones mapping it to *BCL 1* will be weak IDs. When indexing the full text documents, *BCL* would be found using strong rules, and and so this term would be marked as *ssbcl.*

Weak and strong labels were combined depending on the origin of the term. Finding *BCL-1* in the text via a weak rule for the topic transformation, and a strong rule converting from *BCL 1* to *BCL-1*, would result in assigning it the label *wsbcl.* BCL-2 will be marked as *swbcl* as it is derived from a weak rule for BCL.

Finally, *cyclin* will be marked as *wwbcl* as it is obtained from *cyclin D1*, a synonym of *BCL 1*, using a weak rule, and *BCL 1* is derived from the topic using a weak rule.

2.6 Recognizing MeSH Terms

For marking MeSH terms, we used the same recognizer/normalizer tool as for gene names, but used MeSH terms and their synonyms rather than gene databases, and limited transformations to strong rules only. Again, only those MeSH term IDs recognized in the TREC topics were identified and indexed in Lucene for the documents of the collection. MeSH terms were stored in a separate field in the Lucene index but did not use the strength prefixes.

2.7 Generating Queries

Our strategy was to issue a series of queries, starting with stricter constraints to achieve high precision, and then loosening constraints and issuing additional queries until the target number of documents (2000) was retrieved. The order of retrieved documents was retained; that is, the first set of results were kept at the top of the list, the next set appended on to the end (after removing duplicates), and so on. Only the top 1000 documents were returned to NIST for the Biotext1 run, as required by TREC rules, but the deeper set was retained for the Biotext3 run described below. (The Biotextweb run used the top 1000 documents, but reordered them.) For every query, we added the Boolean restriction "NOT section:references" to remove reference sections from consideration, as the Genomics Track guidelines stated that references were not valid results. The standard Lucene ranking algorithm was used as the rank order for the results of each query.

For the first query, for a given TREC topic, we first removed stopwords and then identified the gene names and the MeSH terms that we could extract from the topic, either directly or through synonym expansion as described above. If no gene names were found, the first query was simply a query on an OR of the MeSH terms (plus the modifier removing reference sections from consideration). If gene names were found, the first query was an OR of the gene names ANDed with an OR of the MeSH terms. Some gene names are part of MeSH, so we took care not to double-count gene names within the MeSH terms.

When analyzing the TREC topic, we checked to see if more than one gene name was mentioned in the topic. If so, we found a different set of synonyms for each gene name. For each gene in the topic, we weighted its synonyms depending on what kind of gene synonym it was. We boosted genes labeled *ss* by 100, those labeled *sw* by 20, those labeled *ws* by 10 and those labeled *ww* by 5. (These weights were entirely ad hoc; a training set would have been a great aid for setting these weights.) The weighted synonyms for each gene were combined into an OR; if the original topic contained more than one gene, we combined the disjuncts for each gene with an AND.

The second query took into account the fact that some non-gene and non-MeSH terms in the query might be useful for ranking. It built one part which was an OR of all of the topic terms that were neither stopwords nor gene names. It then ANDed these terms with the original gene part of the query, if gene names were found.

The third query started with the original gene part of the query, and if it contained an AND (meaning we detected more than one gene name

in the original TREC topic) we replaced this AND with an OR, thus relaxing the requirement that every gene from the topic appear in the retrieved documents. Note that for most topics, this query would not be run since we assume most contained only one query.

The fourth query was an OR of the prior parts: the gene part of the query, the MeSH part of the query, and the other-query-words part of the query.

3 BiotextWeb: Using Web Statistics

The BiotextWeb run was inspired by [3], who use the Web as an external thesaurus in order to supplement the topic descriptions, thus achieving notable improvements on the TREC 2004 Robust Track.

The BiotextWeb run took the output of the previous run (up to 1000 documents) and reranked the documents using statistics derived from the Web. It tried to identify which verbs and noun compounds were associated with the important entities in the target question.

So for the example topic "*How do HMG and HMGB1 interact in hepatitis?*" for each pair (s, m), $s \in S$, $m \in M$, we generate a query using the Google search engine to determine which terms are associated with these query terms. Sample generated queries are:

```
"ac2 008" AND "hepatitis"
"clb" AND "hepatitis"
...
```

If the topic is missing a gene or a MeSH term, we generated Google queries for only those entities that were present.

Non-verbs	**2-grams**	**3-grams**
hepatitis	hepatitis b	hepatitis b virus
virus	hepatitis c	coa reductase inhibitor
hmg	coa reductase	hepatitis c virus
coa	b virus	high mobility group
reductase	reductase inhibitor	b virus x
protein	c virus	mobility group box
inhibitor	mobility group	hmg coa reductase
clb	high mobility	group box 1
hmgb1	virus x	virus cellular receptor
high	hmg coa	mobility group protein
group	group box	cellular receptor 1
mobility	chronic hepatitis	virus x interact
hmg1	box 1	c virus core
columbus	virus cellular	hepatitis b surface
box	cellular receptor	hepatitis delta antigen
chronic	hepatitis delta	box transcription factor
antigen	group protein	chronic hepatitis c

Table 1: The most frequent non-verb unigrams, bigrams and trigrams found from Web queries for the example topic.

We collected all returned snippets for all queries (Google returns up to 1000 results per query), sentence-split and POS-tagged them using OpenNLP tools (opennlp.sourceforge.net), and collected all word n-grams ($n = 1, 2, \ldots, 6$). For the unigrams, we collected separately the verbs and the non-verbs, and for $n = 2, 3, \ldots, 6$ the part-of-speech was limited to adjectives (JJ), nouns (N), numbers (CD), foreign words (FW), list elements (LS), and symbols (SYM). The words were normalized using WordNet [2]. Table 1 shows the most frequent non-verb unigrams, bigrams and trigrams, and Table 2 lists the most frequent verbs and tetragrams.

For each candidate document, we calculated the following score:

$$score = \sum_{n=1}^{6} e^{2n} \sum_{\overline{w} \in \overline{W}_n} c(\overline{w}) \ln f(\overline{w}) \qquad (1)$$

where n indicates the size of the n-gram (verbs

Verbs	Tetragrams
associate	hepatitis b virus x
interact	mobility group box 1
call	hmg coa reductase inhibitor
include	high mobility group protein
induce	virus cellular receptor 1
cause	high mobility group box
contain	b virus x interact
use	hepatitis c virus core
increase	b virus x associate
inhibit	hepatitis b surface antigen
relate	box transcription factor 1
meet	hepatitis b virus integration
mediate	b virus integration site
report	hepatitis c virus rna
activate	virus cellular receptor 2
develop	c virus core protein
work	mobility group protein 1

Table 2: The most frequent verbs and tetragrams found from Web queries for the example topic.

are considered as regular n-grams in this formula), $\overline{W}_n$ is the set of all different n-grams extracted from the Web for the given topic, $c(\overline{w})$ is the frequency of the word $\overline{w}$ in the target document, and $f(\overline{w})$ is the frequency of $\overline{w}$ in the Google snippets.

Surprisingly, the BiotextWeb run performed rather poorly as compared to Biotext1. After the evaluation, we analyzed the results and surmise the following reasons for this. The score computed totally ignored the original ranking, that is, the documents containing actual topic entities (genes and MeSH terms) or terms were not given any preference. For example, Table 1 shows that, everything else being equal, a document containing the non-query term *virus* is as good as one containing the query term *hepatitis*. In fact, the former would be even better, as the term *virus* is almost 7 times more frequent on the Web than *hepatitis*, and therefore will be weighted more highly in the formula. Another problem is the exponential growth of the weight of the n-gram for longer n-grams: we used e^{2n}, while in the BLEU score [4] (which partly inspired this formula) the weight is given by $\frac{1}{n}$. That is, it actually decreases, since in BLEU the longer n-grams contain several shorter n-grams for which they will be given extra weight. Finally, as mentioned above, some of the synonyms of the named entities are ambiguous words, which are likely to refer to non-related notions when used in a web query, as seen in *columbus* in set S above.

4 Biotext3: Passage Ranking

This run was intended to address the passage extraction portion of the task by honing in on the best sentences. For each potentially relevant legal text span, the algorithm examined different combinations of passage lengths to determine which would be the best subset to return. The algorithm began with the top 2000 documents retrieved for the Biotext1 run and attempted to extract and rank the best subparts of the best spans.

First, to determine which features might be useful, we made up several queries in the GTT (Generic Topic Type) format, and hand-labeled documents with our own relevance judgements. We also wrote code to identify which sentences were from which sections, and by examining our internally built relevance judgements, we came up with a scoring system for passages that promoted sentences from the Abstract, Introduction, Conclusions sections, and to a lesser degree from the Results section, and demoted sentences from Footnotes, References, Appendices, Abbreviation Lists, Acknowledgements, and Methods sections. We also tried to deter-

mine in advance (without looking at any TREC topics) which subtrees of MeSH were most appropriate for each GTT, and assigned weights to the subtrees accordingly. Unfortunately, due to the lack of training data, these weights had to be set in an ad hoc manner.

For the Biotext1 run we had labeled gene synonyms as strong or weak and MeSH labels extracted from the text as strong or weak indicators for query terms. In the reranking step, we gave more weight to occurrences of strong gene names and MeSH labels than to weak ones. The GTT-based MeSH subtree weighting was also incorporated into the scoring of MeSH terms found within the text passage. Again, the scores combination methods were ad hoc.

Unfortunately, just before submission of this run, we realized that our code that had removed the HTML markup for the documents did not properly align the sentence text with the original position information. Therefore, all of our character location calculations were off and we couldn't return precise span information. We then had to adjust the code to return full legal spans, and may have introduced an error at this point. This algorithm was quite complex and we have not yet assessed its behavior to determine why the output was so much weaker than for our base run.

5 Future Work

We look forward to next year's track, which will build on the dataset developed in the judging process for this year's track. Full text documents are an interesting challenge, and we believe that future bioscience journal search engines will be built on these rather than on the traditional PubMed abstracts.

Acknowledgements: This work was supported in part by NSF DBI-0317510 and by an IBM UIMA grant.

References

[1] Bhalotia, G., Nakov, P., Schwartz, A., and Hearst, M. Biotext team report for the trec 2003 genomics track. In *Proceedings of TREC* (Gaithersburg, MD, 2004).

[2] Fellbaum, C., Ed. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[3] Kwok, K., Grunfeld, L., Sun, H., Deng, P., and Dinstl, N. Trec 2004 robust track experiments using pircs. In *13th Text REtrieval Conference (TREC2004)* (Gaithersburg, MD, 2004).

[4] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Morristown, NJ, USA, 2001), Association for Computational Linguistics, pp. 311–318.