IIT TREC-2007 Genomics Track: Using Concept-based Semantics in Context for Genomics Literature Passage Retrieval

Jay Urbain
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL
urbajay@iit.edu

Nazli Goharian
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL
goharian@iit.edu

Ophir Frieder
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL
frieder@iit.edu

Abstract

For the TREC-2007 Genomics Track [1], we explore unsupervised techniques for extracting semantic information about biomedical concepts with a retrieval model for using these semantics in context to improve passage retrieval precision. Dependency grammar analysis is evaluated for boosting the rank of passages where complementary subject/object concept pairs can be identified between queries and sentences from candidate passages.

In our model, a concept is represented as a set of synonymous terms and a concept-word distribution. Concept terms are identified using an information extraction technique relying on shallow sentence parsing, external knowledge sources, and document context.

The system combines a dimensional data model for indexing scientific literature at multiple levels of document context, with a rule-based query processing algorithm. The data model consists of two hierarchical indices: one for individual words and a second for extracted concepts. The word index provides retrieval of single or multi-word terms. The concept index provides efficient retrieval of single or multiple independent concepts.

A retrieval function combines concepts with term statistics at multiple levels of context to identify relevant passages. Finally, we boost the relevance score of sentences identified within a passage where we can identify term dependencies that complement subject/object pairs between query and passage sentences via dependency grammar analysis.

Our objective for this year's forum was to improve passage retrieval precision. We submitted three automatically generated results for three variations of our retrieval model to the TREC forum. The three results exceeded the track median for character based passage retrieval by 75 to 93%. The mean average precision (MAP) for our top passage retrieval model was 0.0940 which compares favorably to the top result of 0.0976.

1. Introduction

Information retrieval in the genomics literature domain is challenging due to the wide variation of synonymous terms, acronyms, and morphological variants used for identifying the same biological concepts. In addition, acronyms frequently have multiple meanings (polysemy) and require contextual clues for accurate disambiguation. For example, the terms "bovine spongiform encephalopathy", "BSE", and "Mad Cow Disease" are all different terms representing the same named entity or concept. Search terms also have much higher relevance when matched against document terms when occurring within the local context of a phrase, sentence, or passage of text. An acronym like "IP" could represent "immunoprecipitant" or "ischemic precondition." In this case, context captured at the paragraph or document level where an acronym is defined can help disambiguate its meaning [2].

Databases from the National Center for Biotechnology Information (NCBI) [3] and other sources can be helpful in providing semantic evidence supporting identification and extraction of named biological entities. However, it is important to recognize that no knowledge source can fully capture the complexities of human language let alone be fully up-to-date with the dynamic vocabulary of an evolving science. In most cases, there are varying levels of semantic evidence which can make accurate identification of biological concepts difficult. In these cases, optimal retrieval solutions need to integrate additional sources of evidence including identification of key phrases and terms within context.

We propose that effective search requires a systematic approach for combining semantic and contextual evidence. Our approach relies on an indexing model that supports search of single and multi-word terms to support identification of concept term variants, search at different levels of document structure for identifying terms and concepts within context, and integration of external knowledge sources to aid in the identification and resolution of named biological entities and related

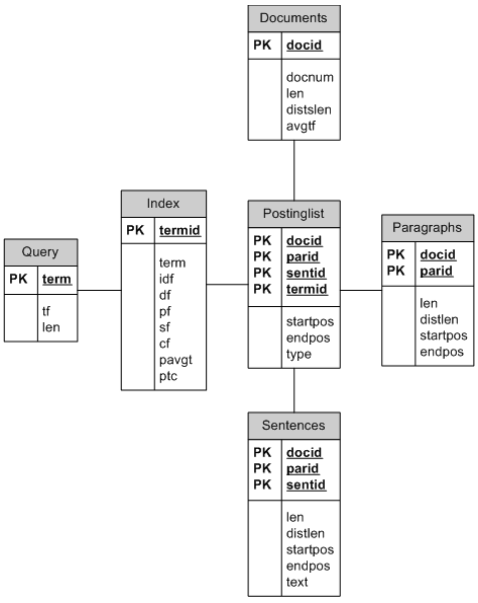
synonymous terms.

We first describe our indexing model, followed by the indexing process, query processing, our methods, results, and a discussion of related work.

2. Dimensional Data Model

Paragraphs, sentences, and terms, representing complete topics, thoughts, and units of meaning respectively, provide a logical breakdown of document lexical structure into finer levels of meaning and context. We capture these hierarchical relationships within a search index based on a dimensional data model. As shown in **Figure 1**, the dimensional index has a *dimension* table for each level of document structure (document, paragraph, sentence, term) and one *central fact* table or *postinglist*. The *postinglist* represents a single mapping table, containing foreign key fields that map the relations between all dimensions. The "grain", i.e., the smallest non-divisible element of the database, is the individual word. Sentences aggregate words in sequence by position, paragraphs aggregate sentences, and documents aggregate paragraphs. In the data warehousing literature, this model is refered to as a *star schema* [4,5].

Figure 1. Search index based on dimensional model.



The *postinglist* includes a term's position within a sentence, textual representation, as well as term and morphological variants. The dimensional indexing model can be extended to include additional dimensions, and allows for efficient formulation of SQL search queries. By indexing each individual word, queries can be developed for searching single- and multi-word terms, and term statistics can be aggregated over different levels of document structure.

3. System Description

Indexing, retrieval, and analysis applications were developed in Java and the system utilizes the Oracle 10g Personal Edition database. The system is platform and database independent. TREC retrieval runs were performed on a 3.1GHz Pentium 4 PC with 2 GB of main memory.

4. Indexing Process

The indexing process includes the following:

1. *Lexical Partitioning*: Documents are parsed into paragraphs. Paragraphs are parsed into sentences.
2. *Acronym identification*: Acronyms and their long-forms are identified during indexing using the Schwartz and Hearst algorithm [6]. A long-short form would include "vasoactive intestinal peptide (VIP)", and a short-long form would include "VIP (vasoactive intestinal peptide)". The algorithm works backwards through the long form text and attempts to identify corresponding letters in the acronym. Acronyms and their long-forms are added to an acronym table to help with disambiguation. Long-form variants are added to the same indexing location as acronyms during indexing (and vice versa). This technique helps disambiguate acronyms, and allows identification of passages using either the short- or long-form of an entity.
3. *Tokenization*: Sentence terms are tokenized, stop words removed, and lexical variants of gene and protein names are generated [7]. Porter stemming [8] is used on each token with the following exceptions: gene names (as defined by the Entrez Gene database); all upper case, mixed case, alpha-numeric terms; and non-gene terms that would become a gene name after being stemmed. Small "s" is also stripped from all upper-case terms.
4. *Indexing*: Each term along with its long-form expansion and lexical variants are stored in the index with the same positional information.

5. Query Processing

Structured query generation is illustrated with the following query: *"Provide information about the role of the gene PRNP (prion protein) in the disease Mad Cow Disease"*.

1. Sentences are extracted, acronyms and their long forms are identified: PRNP (PRioN Protein).
2. Part-of-speed tagging is performed using our 2nd order statistical Hidden Markov Model tagger: … *role_NN of_II the_DD gene_NN PRNP_NN (_(prion_NN protein_NN)_) in_II the_DD disease_NN Mad_NN Cow_NN Disease_NN*.

3. Stop and function words are removed from further processing.

4. Candidate entities are identified by locating non-recursive noun phrases ("noun chunks"): *[gene PRNP], [prion protein], [Mad_NN Cow_NN Disease_NN]*.

5. Candidate entities are verified in the index, and resolved using the UMLS Metathesaurus®, OMIM™ (Online Mendelian Interface to Man), MeSH (Medical Subject Headings), and Entrez Gene databases. If an entity is successfully resolved, all synonyms and one level of hyponyms, i.e., child terms, are identified.

Prior to including synonyms as a concept term variant, its level of ambiguity is determined. If the synonym is considered ambiguous it is not included. We consider a term ambiguous if either of the following tests is met:

1. The synonym's normalized inverse document frequency (NIDF) is < 0.1. Where NIDF is the IDF=*log (N/df) normalized to between 0 and 1*.

2. The synonym correlates with the correct long-form in less than 50% of all instances within the acronym table

Resolved concepts and corresponding synonyms are shown in **Table 1**. Resolved concept instances are added to a *concept index* with the same structure and fact tables as the dimensional *term index* described in **Figure 1**, except the *postinglist* table is replaced with a *conceptlist*.

Table 1 – Retrieval function weighting and similarity coefficients

Resolved concepts	Synonyms
[Encephalopathy, Bovine Spongiform]	[Mad Cow Disease] [MCD] [BSE] [Creutzfeldt-Jakob disease] [CJD]
[PRNP gene]	[prion protein] [prnp]

Search can be performed within the context of an individual term/phrase, sentence, paragraph, or document. For TREC, we first perform paragraph-level searches using the probabilistic BM25 retrieval function [9] shown in equation (1) implemented in standard SQL [7, 10].

BM25: (1)

$$\sum_{wq} \ln\left(\frac{N-df+0.5}{df+0.5}\right)\left(\frac{(k_1+1)*tf_d}{k1*(1-b)+b*(\frac{docLen}{avgDocLen})+tfd}\right)\left(\frac{(k_3+1)*tfq}{k_3+tf_q}\right)$$

Note: We used k1=1.4, k2=0, k3=7, and b=0.75 [7].

Next, using the top 1000 paragraphs we perform a concept search as follows:

1. The position of all term variants of each concept is retrieved from the dimensional index by paragraph.

2. A concept graph is constructed by creating an adjacency list using each concept term as a vertex.

3. A minimum-spanning tree is constructed from the adjacency list by determining the maximum number of distinct concepts within the shortest lexical distance. Distance measurements are weighted such that terms within a lexical unit, e.g., a sentence, are always closer than terms in separate units.

4. Finally, the passage boundary based on the first and last occurrences of distinct concepts is expanded out to include sentence boundaries.

Passage level concept search is further illustrated with the following query: *"Exact reactions that take place when you do glutathione S-transferase (GST) cleavage during affinity chromatography"*.

First, the following concepts and term variants (shown in stemmed form) are identified:

- *Cleavage:* [[cleavag], [merogenesi], [cytokinesi]]
- *Affinity purification*: [affin, purif], [affin, chromatographi]]
- *Glutathione S-transferase*: [[glutathion, s, transferase], [gst]]

Second, the index is searched for all term variants of each concept. The following query searches for the concept term variant "*affinity, chromatography*":

```
select i1.term as term1, i2.term as term2, p1.docid,
    p1.parid, p1.sentid, p1.startpos, p1.endpos
from invertedindex i1, invertedindex i2, postinglist p1,
    postinglist p2
where i1.term='affin' and i2.term='chromatographi'
and i1.termid=p1.termid and i2.termid=p2.termid
and p1.docid=p2.docid and p1.parid=p2.parid
and p1.sentid=p2.sentid and abs(p2.seq-p1.seq)<=2;
```

Third, passages are identified: "***affinity chromatography***, *and purified Mce1A and Mce1E, free of the fusion partner, were recovered following specific proteolytic* ***cleavage*** *of the* ***GST****"*

Finally, passages are expanded to sentence boundaries: *"The fusion proteins were purified to near homogeneity by* ***affinity chromatography****, and purified Mce1A and Mce1E, free of the fusion partner, were recovered following specific proteolytic* ***cleavage*** *of the* ***GST*** *portion by thrombin protease."*

The following similarity coefficients (SC) are identified for each candidate passage:

- Number of distinct concepts for the entire passage weighted by the likelihood of the words in the sentence containing the concept.

- The normalized sum of the normalized IDF's of each concept within the passage.
- Number of distinct concepts for the top sentence within the passage weighted by the likelihood of the words in the sentence containing the concept.
- The normalized sum of the normalized IDF's of each concept within the top sentence within the passage.

A linear weighted sum (2) is used to generate various retrieval models by weighting and combining similarity coefficients (SC) for each passage.

$$SC_{composite} = w_1SC_1 + w_2SC_2 + \ldots + w_nSC_n \quad (2)$$

Passages with the same $SC_{composite}$ are ranked by the passage's lexical distance, i.e., the width of the MST of distinct concept instances.

Finally, we apply Stanford's dependency grammar parser to identify subject/object complements between queries and passage sentences [11]. Dependencies are motivated by grammatical function, i.e., syntactically and semantically. A word depends on another if it is either a complement or a modifier of the latter. If we can identify the modifier of the object of the original query, we increase the likelihood of answering the query. For example, for query 201: *What [mutations] in the Raf gene are associated with cancer?* We retrieved the following passage MST: *…melanoma cell lines with B-RAF and N-RAS mutations…* for which we can identify dependencies between the modifiers *B-RAF* and *N-RAS* and the object of the sentence *mutations* which was the subject of the original query.

6. Results

Retrieval model weighting and similarity coefficients for each submitted run are summarized in **Table 2**. *IITx1* emphasizes passage level weighting, while *IITx2* emphasizes weighting the top scoring sentence for each passage. *IITx2* also used dependency grammar rank boosting. *IITx3* used the *IITx1* retrieval model with dependency grammar boosting.

The results for are three automatically generated runs are summarized in **Table 3**. All three results significantly outperformed the median results for the track, including the character-based passage retrieval measurement we sought to optimize. We believe the heavy emphasize of our retrieval functions on identifying distinct biological concepts helped precision for passage retrieval, but otherwise reduced recall for document, aspect, and the passage2 measurement. Integrating the query term density measurement we utilized with last year's track [2] would most likely improve these scores.

We also discovered an error in our database software where frequently occurring terms were not stored in our index. Such terms included *gene* and *protein*. The corrected results for *IITx3* are shown in **Table 4**.

Table 2 – Retrieval model weighting and similarity coefficients

Run	Retrieval Function
iitx1	0.45*distinct number of passage concepts + 0.05*distinct number of sentence concepts + 0.45*passage norm IDF sum + 0.05*sentence norm IDF sum *no dependency grammar passage rank boost*
iitx2	0.05*distinct number of passage concepts + 0.45*distinct number of sentence concepts + 0.05*passage norm IDF sum + 0.45*sentence norm IDF sum *dependency grammar passage rank boost*
iitx3	0.45*distinct number of passage concepts + 0.05*distinct number of sentence concepts + 0.45*passage norm IDF sum + 0.05*sentence norm IDF sum *dependency grammar passage rank boost*

Table 3 - Results for runs submitted to TREC (% above track median)

Run	Document MAP	Aspect MAP	Passage MAP	Passage2 MAP
iitx1	0.2454 (31.15%)	0.1272 (18.06%)	0.0852 (75.27%)	0.0388 (39.82%)
iitx2	0.2462 (31.60%)	0.1166 (8.16%)	0.0926 (90.38%)	0.0335 (20.56%)
iitx3	0.2414 (28.99%)	0.1253 (16.25%)	0.0940 (93.22%)	0.0443 (59.30%)

Table 4 – Corrected run

Corrected Run	Document MAP	Aspect MAP	Passage MAP	Passage2 MAP
iitx3	0.2670	0.1662	0.1060	0.0616

7. References

1. W. Hersh, et al., "TREC 2007 Genomics track overview," Proceedings of the Fourteenth Text REtrieval Conference, Gaithersburg, MD, 2007.

2. J. Urbain, N. Goharian, O. Frieder, "IIT TREC-2006: Genomics Track," Proceedings of the Fifteenth Text REtrieval Conference, 2006

3. National Center for Biotechnology Information (NCBI), http://www.nlm.nih.gov.

4. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M Venckatrao, F Pells, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Data Mining and Knowledge Discovery," Volume 1, Issue 1, 1997.

5. R. Kimball, "The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses," Ralph, John Wiley, 1996.

6. A. Schwartz, M. Hearst, "A simple algorithm for identifying abbreviation definitions in biomedical text," Pacific Symposium on Biocomputing, 2003.

7. J. Urbain, N. Goharian, "A Relational Genomics Search Engine," BIOCOMP 2006: 69-74.

8. M.F. Porter, "An algorithm for suffix stripping," Program, 14:130–137, 1980.

9. S. Robertson, S. Walker, "Okapi/Keenbow at TREC-8," NIST Special Publication 500-246, 2000.

10. D. Grossman, O. Frieder, "Information Retrieval: Algorithms and Heuristics," Second Edition; Springer Publishers, ISBN 1-4020-3003-7, 1-4020-3004-5, 2004.

11. Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.