TREC Genomics Track at UIC

Wei Zhou, Clement Yu
Department of Computer Science,
University of Illinois at Chicago
wzhou8@uic.edu; yu@cs.uic.edu

1. Introduction

We considered that a query of this year's Genomics track consists of two parts, target and qualification. A target refers to any instance of a certain entity type and the qualification refers to the condition that the target needs to satisfy in order to be qualified as an answer to the query. For example, the target and qualification of the query "What [ANTIBODIES] have been used to detect protein TLR4?" are "an instance of ANTIBODIES" and "have been used to detect protein TLR4", respectively. The relevance of a document to a query was measured by to what degree the document contains a target and satisfies the qualification.

We observed that for some entity types (Type I), such as "SIGNS OR SYMPTOMS", we were able to retrieve some candidate targets from some resources, such as UMLS. For example, "headache" and "cough" are found to be candidate targets of "SIGNS or SYMPTOMS" in the UMLS. For other entity types (Type II), such as "ANTIBODIES" and "PATHWAYS", we were unable to find resources to retrieve the candidate targets so far. We employed different strategies for these two types of queries.

For queries involving Type I entity types, we ask whether a document contains any candidate target and in addition, does the document match the qualification. An interesting issue is that whether different candidate targets should be differentiated according to their frequencies. Two experiments, one differentiates candidate targets and the other does not, have been carried out to investigate this issue.

For queries involving Type II entity types, we have compiled a set of representative words for each entity type. These representative words co-occur with the entity type much more often than being independent in the same sentences of PubMed abstracts. Then the raw query was expanded with the set of representative words. The intuition is that, the representative words are expected to capture the context or surrounding neighborhood within which a target of that entity type occurs and a document having that context is much likely to contain a target also.

This year's task is still a passage retrieval task, where a passage could be a part of a sentence, a sentence, a set of consecutive sentences or a paragraph (i.e., the whole span of text that are inside of <P> and </P> HTML tags). We approached the task by having two steps. The first step, document (or paragraph) retrieval, retrieves the top 2,000 paragraphs. The second step, passage retrieval, extracts passages from each paragraph and then ranks those passages according to their relevance with respect to the query. This notebook paper is organized as follows: the paragraph retrieval is introduced in section 2; the passage extraction and retrieval is presented in section 3; in section 4, we present the experiment results; and the summary is given in section 5.

2. Document (or Paragraph) Retrieval

The document collection contains 162,259 Highwire full-text documents in HTML format. We partitioned each document into paragraphs by using the HTML paragraph tag and . The paragraph retrieval step retrieves the top 2,000 relevant paragraphs, assuming that the top 1,000 most relevant passages can be extracted from the top 2,000 paragraphs. We employed a conceptual retrieval model [1] for the paragraph retrieval step.

We divided the 14 entity types into two groups. The first group G_1 consisting of 9 types that either correspond to some UMLS semantic types (e.g., the entity type "SIGNS OR SYMPTOMS" corresponds to the UMLS semantic type "Sign or Symptom") or can be

G1			G2
Entity Type	**Resource**	**Mapping**	**Entity Type**
BIOLOGICAL SUBSTANCES	UMLS	Biologically Active Substance	ANTIBODIES
CELL OR TISSUE TYPES	UMLS	Cell Tissue	MUTATIONS
DISEASES	UMLS	Disease or Syndrome	PATHWAYS
DRUGS	UMLS	Pharmacologic Substance	STRAINS
GENES	Entrez Gene	Gene symbols	TOXICITIES
MOLECULAR FUNCTIONS	UMLS	Molecular Function	
PROTEINS	UMLS	Amino Acid, Peptide, or Protein	
SIGNS OR SYMPTOMS	UMLS	Sign or Symptom	
TUMOR TYPES	UMLS	Neoplastic Process	

Table 1. 14 entity types are classified into two groups, G_1 and G_2. Entity types in G_1 are those that we are able to retrieve their instances from some resources; Entity types in G_2 are those that we are unable to find any resources to automatically retrieve their instances.

found in the Entrez Gene database. The second group G_2 consisting of the remaining 5 types that we are not sure they correspond to any UMLS semantic type, such as "ANTIBODIES". The classification of the 14 entity types are given in Table 1.

2.1 Identifying Query Concepts and Retrieving Domain-specific Knowledge

To apply this conceptual retrieval model, we need to identify the query concepts (MeSH terms or gene symbols in the Entrez Gene). We use the query translation functionality of PubMed to extract MeSH terms in a query. This is done by submitting the whole query to PubMed, which will then return a file in which the MeSH terms in the query are labeled. Query terms in the Entrez Gene database were identified.

For each query concept, we retrieved its domain-specific knowledge, which refers to information about concepts and their relationships in a certain domain. We used five types of domain-specific knowledge in the domain of genomics:

Type 1. Synonyms (terms listed in the thesauruses that refer to the same meaning)
Type 2. Hypernyms (more generic terms, one level only)
Type 3. Hyponyms (more specific terms, one level only)
Type 4. Lexical variants (different forms of the same concept, such as abbreviations. They are commonly used in the literature, but might not be listed in the thesauruses)
Type 5. Implicitly related concepts (terms that are semantically related and also co-occur more frequently than being independent in the biomedical texts)

Knowledge of type 1-3 is retrieved from the following two thesauruses: 1) MeSH, a controlled vocabulary maintained by NLM for indexing biomedical literature. The 2007 version of MeSH contains information about 190,000 concepts. These concepts are organized in a tree hierarchy; 2) Entrez Gene, one of the most widely used searchable databases of genes. The current version of Entrez Gene contains information about 1.7 million genes. It does not have a hierarchy. Only synonyms are retrieved from Entrez Gene. The compiling of type 4-5 knowledge was introduced in [1].

2.2 Retrieval Models for G_1 Type of Queries

A query q involving an entity type of G_1, the query can be written as:

$$q = \{g, \{c_1, c_2, ..., c_m\}, \{w_1, w_2, ..., w_n\}\},$$

where g is the entity type in q, $\{c_1, c_2, ..., c_m\}$ is a vector of query concepts, $\{w_1, w_2, ..., w_n\}$ is a vector of query words. Suppose for entity type g, we can retrieve k candidate targets (or instances) from some

resources, such as UMLS or Entrez Gene. Let g be denoted as $g = \{g_1, g_2, ..., g_k\}$. It is possible a document (or paragraph) contains multiple targets of type g. An interesting issue is whether we should differentiate what targets of type g a document has. We explored two different models to investigate this issue. One model assigns the same similarity value to a document with respect to the entity no matter what targets the document has. The other model weights the targets, $g_1, g_2, ..., g_k$, according to their global frequencies.

Model 1

The similarity between query q and document d is measured on two levels: the concept level and word level. The concept-level similarity is obtained by matching targets and query concepts, while the word-level similarity is obtained by matching words

$$sim(q,d) = (\underset{concept}{sim}(q,d), (\underset{word}{sim}(q,d))$$

In Model 1, each target $g_i \in g$ is assigned a weight $wt(g_i)$ using the IDF of g_i. Two values were computed by matching targets and query concepts. The concept-level similarity is a linear combination of these two values:

$$\underset{concept}{sim}(q,d) = \alpha \times MAX\{wt(g_i) \mid g_i \in d\} + (1-\alpha) \times \sum_{c_i \in d} wt(c_i),$$

where α is a tuning parameter. In the experiments, $\alpha = 0.25$.

The similarity between q and d on the word level is computed using Okapi [2]:

$$\underset{word}{sim}(q,d) = \sum_{w \in q} \log\left(\frac{N-n+0.5}{n+0.5}\right)\left(\frac{(k_1+1)tf}{K+tf}\right)$$

where N is the size of the document collection; n is the number of documents containing w. The other parameters are defined in [2].

Given two documents d_1 and d_2, we say $sim(q,d_1) > sim(q,d_2)$ or d_1 will be ranked higher than d_2, with respect to the same query q, if either

$$\begin{aligned} &1)\ \underset{concept}{sim}(q,d_1) > \underset{concept}{sim}(q,d_2)\ \text{OR} \\ &\quad \underset{concept}{sim}(q,d_1) = \underset{concept}{sim}(q,d_2)\ \text{AND} \\ &2)\ \underset{word}{sim}(q,d_1) > \underset{word}{sim}(q,d_2) \end{aligned} \quad (1)$$

In this model, two similarity values are computed between and a document d and a query q. The concept-level similarity is obtained by matching concepts, while the word-level similarity is obtained by matching words. This conceptual IR model emphasizes the similarity on the concept level. More precisely, documents are ranked in descending order of the concept-level similarity. Only when documents have the same similarity in concepts, then the similarities in words are used to break ties. A similar model but applied to non-biomedical domain has been given in [3]. Okapi relevance scoring formula is known to embody a good model of relevance based upon term occurrences within text documents and the length of the documents. However, Okapi, in general, does not work well if the query contains multiple concepts.

Model 2

This model does not differentiate what targets a document has. In another words, all the documents having at least one $g_i \in g$ receive the same similarity value with respect to the targets. The similarity between query q and document d is measured on three levels: the target concept-level, qualification concept-level and word level. The value of the target concept-level similarity is either 0 or 1 to indicate whether d has any target in g. The qualification concept-level similarity is obtained by matching query concepts, while the word-level similarity is obtained by matching words.

$$sim(q,d) = (\underset{target}{sim}(q,d), (\underset{qlf}{sim}(q,d), (\underset{word}{sim}(q,d))$$

Given two documents d_1 and d_2, we say $sim(q,d_1) > sim(q,d_2)$ or d_1 will be ranked higher than d_2, with respect to the same query q, if any of the following three conditions satisfies:

1) $\underset{target}{sim}(q,d_1) > \underset{target}{sim}(q,d_2)$

2) $sim_{target}(q,d_1) = sim_{target}(q,d_2)$ AND $sim_{qlf}(q,d_1) > sim_{qlf}(q,d_2)$

3) $sim_{target}(q,d_1) = sim_{target}(q,d_2)$ AND $sim_{qlf}(q,d_1) = sim_{qlf}(q,d_2)$ AND $sim_{word}(q,d_1) > sim_{word}(q,d_2)$

In this model, documents having at least one target will be ranked higher than documents having no targets at all. For those documents having at least one target, the qualification concept-level similarity will be used next to break the tie. If two documents continue to have a tie on the qualification concept-level similarity, then the word-level similarity will be used to break the tie. In the TREC 2007 Genomics Track protocol [4], a retrieved document is said to be relevant as along as it contains some entity of the specified entity type in the query. Thus Model 2 is more consistent with the protocol than Model 1, which has been demonstrated by the experiment results.

2.3 Retrieval Models for G_2 Type of Queries

Unfortunately, we were unable to find resources to retrieve targets for all the entity types. When this happens, we compile a set of representative words for that entity type. These representative words co-occur with the entity type much more often than being independent in the same sentences of PubMed abstracts. The top 15 representative words were added into the query. Two levels of similarities were computed: concept level and word level. The concept-level similarity is obtained by matching the query concepts and the word-level similarity is obtained by matching the original query words and the added representative words. Then the model given in (1) was used for ranking. The intuition is that, the representative words are expected to capture the context or surrounding neighborhood within which a target of that entity type occurs and a document having that context is much likely to contain a target also.

3. Passage Extraction

The goal of passage extraction is to highlight the most relevant fragments of text in paragraphs. A passage is defined as any span of text that does not include the HTML paragraph tag (i.e., <P> or </P>). A passage could be a part of a sentence, a sentence, a set of consecutive sentences or a paragraph (i.e., the whole span of text that are inside of <P> and </P> HTML tags). It is also possible to have more than one relevant passage in a single paragraph. Our strategy for passage extraction assumes that the optimal passage(s) in a paragraph should have all the query concepts that the whole paragraph has. Also they should have higher density of query concepts than other fragments of text in the paragraph.

Suppose we have a query q and a paragraph p represented by a sequence of sentences $p = s_1 s_2 \ldots s_n$. Let C be the set of concepts in q that occur in p and $S = \Phi$.

Step 1. For each sequence of consecutive sentences $s_i s_{i+1} \ldots s_j$, $1 \le i \le j \le n$, let $S = S \cup \{ s_i s_{i+1} \ldots s_j \}$ if $s_i s_{i+1} \ldots s_j$ satisfies that:

1) Every query concept in C occurs in $s_i s_{i+1} \ldots s_j$ and

2) There does not exist k, such that $i < k < j$ and every query concept in C occurs in $s_i s_{i+1} \ldots s_k$ or $s_{k+1} s_{k+2} \ldots s_j$.

Condition 1 requires $s_i s_{i+1} \ldots s_j$ having all the query concepts in p and condition 2 requires $s_i s_{i+1} \ldots s_j$ be the minimal.

Step 2. Let $L = \min\{ j - i + 1 : s_i s_{i+1} \ldots s_j \in S \}$. For every $s_i s_{i+1} \ldots s_j$ in S, let $S = S - \{ s_i s_{i+1} \ldots s_j \}$ if $(j - i + 1) > L$. This step is to remove those sequences of sentences in S that have lower density of query concepts.

Step 3. For every two sequences of consecutive sentences: $S_{i1} S_{i1+1} \ldots S_{j1} \in S, S_{i2} S_{i2+1} \ldots S_{j2} \in S$,

If

$$i_1 \le i_2, j_1 \le j_2 \quad \text{and} \quad i_2 \le j_1 + 1 \tag{2}$$

then do

$$S = S \cup \{ s_{i_1} s_{i_1+1} \ldots s_{j_2} \}$$
$$S = S - \{ s_{i_1} s_{i_1+1} \ldots s_{j1} \}$$
$$S = S - \{ s_{i_2} s_{i_2+1} \ldots s_{j_2} \}$$

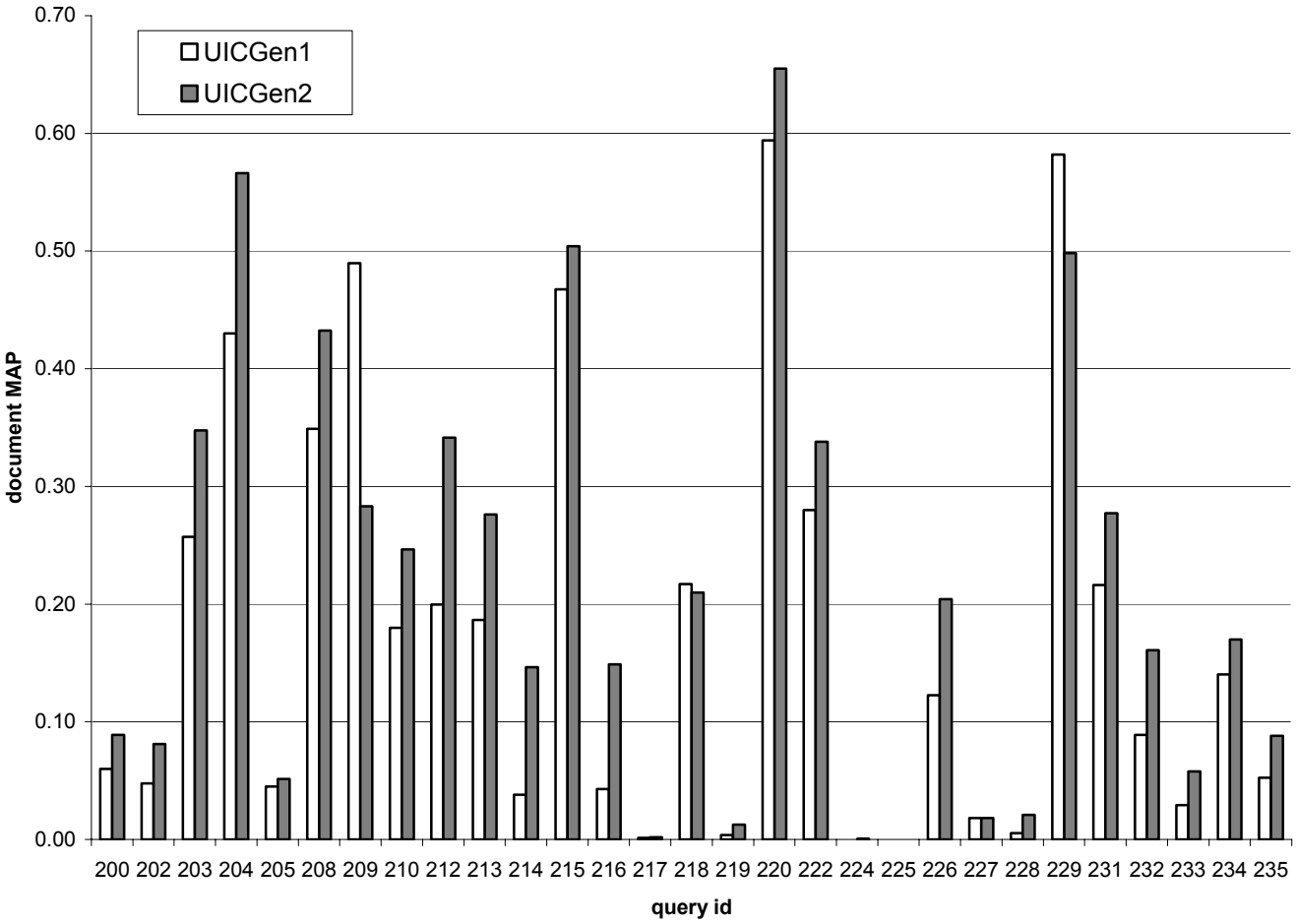

Figure 1. Performance of two runs for queries of G_1 type (i.e., queries involving entity types that their targets can be automatically retrieved from existing resources.)

Repeat this step until for every two sequences of consecutive sentences in *S*, condition (2) does not apply. This step is to merge those sequences of sentences in *S* that are adjacent or overlapped.

4. Experiment Results

We submitted two official runs. For G_2 type of queries, both runs used the same retrieval model. But for G_1 type of queries, different strategies have been used:

UICGen1. We employed Model 1, in which different targets of the same entity type were weighted according to their global frequencies.

UICGen2. Model 2 was employed. Documents having at least one target receive the same weight with respect to the target. These documents were ranked higher than documents having no targets at all.

Table 2 shows the performance of the two official runs. As you can see, the performance of UICGen2 is better than UICGen1 over all 4 different measures, which indicates that a document should be determined as a relevant document as long as it has a target of the specified entity type of the query and also satisfies the qualification no matter which target, more important or less, or how many targets it has.

Run	Doc MAP	Asp MAP	Psg MAP	Psg2 MAP
UICGen1	0.209	0.145	0.083	0.044
UICGen2	0.239	0.181	0.098	0.051

Table 2. Performance of the two official runs.

To further look into the performance of UICGen1 and UICGen2 with respect to each query, Figure 1 shows the document MAP of the two runs for each G_1 type of query. As you can see, for 26 out of 29 queries of G_1 type, the document MAP of UICGen2 is at least as good as that of UICGen1. This result again confirmed the result indicated in Table 2.

To evaluate whether using existing resources is helpful for retrieving targets, we compared the document MAP of UICGen2 with the median document MAP of all 49 automatic official runs in Table 3. As you can see that, for those queries using existing resources to retrieve targets, the improving of document MAP over the median is a little higher than of those queries without using resources. It is possible that representative words of an entity type, to some sense, have captured the context in which a target of that entity type appears. It is good to have another baseline only using the original query.

	Doc MAP	
	Using resources	Without using resources
Median	0.1659	0.2704
UICGen2	0.2147	0.3410
Diff.	**+29.42%**	**+26.11%**

Table 3. Performance of UICGen2 comparing to the median of all 49 automatic official runs.

5. Summary

In summary, in this year's Genomics Track, we explored two different types of strategies to answer information needs like "What [ANTIBODIES] have been used to detect protein TLR4?", which is asking for a list of biological objects of a certain type. This task is more challenging than last year's task where the biological objects or processes are explicitly given in the queries. Two conclusions have been made based on the experimental results. The first conclusion is that a document should be determined as a relevant document as long as it has some target of the specified entity type and satisfies the qualification no matter which targets or how many targets it has. The second conclusion is that using existing resources, such as UMLS or Entrez Gene, to retrieve targets of the entity type is a little more useful than using the representative words of the entity type.

6. Acknowledgements

We thank Wen Tian for helping analyzing the results.

References

[1] W. Zhou, C. Yu, N. Smalheiser, V. Torvik,. Knowledge-intensive Conceptual Retrieval and Passage Extraction of Biomedical Literature. Proceedings of the 30th Annual International ACM SIGIR2007, Amsterdam, Netherlands. pages 655-662.

[2] S. E. Robertson, S. Walker (2000) Okapi/Keenbow at TREC-8. NIST Special Publication. 500-246:The Eighth Text REtrieval Conference (TREC 8).

[3] S. Liu, F. Liu, C. Yu, M. Y. Meng. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. In SIGIR 2004.

[4] W. R. Hersh, A. M. Cohen, P. Roberts. TREC 2006 Genomics Track Protocol. [http://ir.ohsu.edu/genomics/2007protocol.html]