Exploring Abbreviation Expansion for Genomic Information Retrieval*

Nicola Stokes, Yi Li, Lawrence Cavedon and Justin Zobel
National ICT Australia, Victoria Research Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne, Victoria 3010, Australia.
{nstokes, yli8, lcavedon, jz}@csse.unimelb.edu.au

Abstract

Abbreviations are commonly found instances of synonymy in Biomedical journal papers. Information retrieval systems that index paragraphs rather than full-text articles are more susceptible to term variation of this kind, since abbreviations are typically only defined once at the beginning of the text. One solution to this problem is to expand the user query automatically with all possible abbreviation instances for each query term. In this paper, we compare the effectiveness of two abbreviation expansion techniques on the TREC 2006 Genomics Track queries and collection. Our results show that for highly ambiguous abbreviations the *query collocation* effect isn't strong enough to deter the retrieval of erroneous passages. We conclude that full-text abbreviation resolution prior to passage indexing is the most appropriate approach to this problem.

1 Introduction

Query expansion is a well-known technique used in Information Retrieval (IR) to address the problem of lexical variation between the query and semantically related terms in relevant documents (Efthimiadis, 1996). While on average query expansion methods, such as *relevance feedback* (Ruthven and Lalmas, 2003), have been shown to improve retrieval performance, there are many examples where query effectiveness has been significantly downgraded. However, in terminology rich domains where word sense distributions are heavily skewed, query expansion has been shown to have more of a consistent positive effect on retrieval performance. This trend is particularly evident in the passage retrieval task investigated at the TREC (Text REtrieval Conference) Genomics Track (Hersh et al., 2006).

In this paper, we investigate the impact of various expansion term types on passage retrieval effectiveness in the biomedical domain. Our results show that expanding with ontologically related words (synonyms, hypernyms, hyponyms) significantly improves performance; however, abbreviation expansion shows more inconsistent results similar to those seen in general domain expansion experiments. One would expect that the performance of IR systems that index paragraphs rather than full-text articles would greatly benefit from this sort of expansion, since abbreviations are typically only defined once in an entire document.

We report the results of our investigation on the TREC 2006 Genomic retrieval task. We compare two abbreviation expansion techniques: the first adds abbreviations found in the ADAM database of abbreviations (Zhou et al., 2006a); the second, uses a pseudo relevance feedback strategy to identify query term abbreviations in the full-text documents of an initial set of retrieved passages. Despite the benefit of mutual disambiguation across query terms, referred to as the *query term collocation effect*

*National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs.

(Krovetz and Croft, 1992), both approaches reduce retrieval effectiveness, leading to the conclusion that abbreviation resolution in the document collection is more appropriate than expansion.

Another contribution of this paper is our novel concept-based IR ranking method. This ranking method is an adaptation of the Okapi method, enhanced so as to deal with multi-concept queries derived from natural language questions. Our method ensures that passages containing at least one occurrence of all the query concepts out-rank passages that contain many occurrences of only one of the concepts. We also describe a paragraph reduction strategy that increases the TREC defined answer extraction accuracy score of our system. Finally, we discuss our plans for future work.

2 Information Retrieval for Functional Genomics

Biomedical text retrieval is a very active area of research, driven by the biomedical community's need for high precision systems that answer specific biological questions not captured in the plethora of database resources (of varying quality) containing different types of biological information. Two distinct user information needs have been recently investigated by the IR community: *clinical text retrieval* (which supports patient-centred clinical research or care) and *functional genomic text retrieval* (which supports researchers involved in laboratory experiments). In this paper, we focus on genomic retrieval. An interesting overview of evidence-based medical retrieval in the clinical domain can be found in (Lin and Demner-Fushman, 2006).

Functional Genomics is the study of gene and protein function and interaction at a molecular level, and the effects of this interaction on biological processes that results in phenotypic outcomes (such as disease) in organisms. An important yet very time-consuming part of the functional genomics pipeline for researchers involves arriving at biologically motivated explanations for the output of bioinformatics-based clustering techniques such as gene expression profiling. Since a single experiment can involve thousands of genes, even a competent biologist needs to turn to a search engine to determining whether the functional dependencies found in these clusters make sense.

The TREC Genomics Track was established in 2003 with the aim of supporting the evaluation of information retrieval systems capable of answering the types of questions typically posed by genomicists such as:

- What is the role of gene A in disease B?
- What effect does gene A have on a particular biological process?
- How do genes A and B interact in the function of a specific organ?
- How do mutations in gene A influence a particular biological process?

Each of these four query templates were investigated at the 2006 Genomics Track. In all, 28 queries were evaluated on a collection of full-text journal papers, where the task was to retrieved relevant answer passages rather than full-text documents. In the following section we describe our novel genomic retrieval system.

3 System Description

In this section, we describe the different components in our Genomic IR architecture. Our IR system is a version of the Zettair engine[1] that we have specifically modified for passage retrieval and biomedical query term expansion.

Collection Preprocessing

The TREC collection consists of full-text journal articles obtained by crawling the Highwire site[2]. The full collection contains 162,259 documents and is about 12.3 GB in size when uncompressed. After preprocessing, the whole collection becomes 7.9 GB. The collection is pre-processed as follows:

Paragraph Segmentation: for evaluation purposes the Genomics Track requests that the ranked answer passages must be within specified paragraph boundaries.

[1] `http://www.seg.rmit.edu.au/zettair/`

[2] `http://www.highwire.org`

Sentence Segmentation: all sentences within paragraphs are segmented using an open source tool.[3]

Character Replacement: Greek characters represented by gifs are replaced by textual encodings; accented characters such as "À" or "Á" are replaced by "A"; Roman numbers are replaced by Arabic numerals. These replacements are very important for capturing variations in gene names.

Removal: all HTML tags, very short sentences, paragraphs with the heading *Abbreviations*, figures, tables and some special characters such as hyphens, slashes and asterisks are removed: (Trieschnigg et al., 2006) has shown that small changes in the tokenisation strategy such as these improve the performance of biomedical IR.

Query Expansion

Once the collection has been indexed, querying can begin. In the 2006 Genomics Track, each query or topic contains at least two biological concepts or entities which could be a gene ("NM23"), a protein ("p53"), a disease ("ovarian cancer") or a biological process ("ethanol metabolism"). TREC simplifies the query preprocessing task by ensuring that all topics conform to the query templates discussed in Section 2. The following is a sample query, Topic 173 from the 2006 track, which contains two concepts: "PrnP" (a gene) and "mad cow disease" (a disease):

What is the role of **PrnP** *in* **mad cow disease***?*

Our query expansion process proceeds as follows. First, each gene or protein in the query is expanded with entries from the Entrez Gene database.[4] Since the same gene may occur in many different species, and many of their synonyms only differ with respect to capitalisation, we choose the first entry retrieved that belongs to the species type *Homo sapien*. Then, terms in the *Official Symbol*, *Name*, *Other Aliases* and *Other Designations* fields, for the gene, are added to the query.

For all disease and biological process mentions in the query, we use the MeSH[5] taxonomy of medical terms to find their synonyms (using the *Entry Terms* and *See Also* fields). The terms' hyponyms (descendants) and hypernyms (ancestors) in the MeSH tree structure are also used as expansion terms.

Gene Variant Generation

As well as expanding with synonyms, we use a "gene variant" generation tool to generate all the possible variants for both original query terms and expanded terms. Our segmentation rules are similar to those used by (Buttcher et al., 2004). We describe our rules as follows:

Given a gene name containing a hyphen or punctuation, or a change from lower case to upper case, or from a character to a number (or vice versa), or a Greek character (e.g. "alpha"), we call this a *split point*. A word is split according to all its split points, and all variants are generated by concatenating all these split parts, optionally with a space inserted. Greek characters are also mapped to English variants, e.g. "alpha" is mapped to "a".

For example, on the query term "Sec61alpha", we would generate the following lexical variants which are also commonly used forms of this term in the collection: "Sec 61alpha", "Sec61 alpha", "Sec 61 alpha", "Sec 61a", "Sec61 a", "Sec 61 a", "Sec61a";

In phrases, we replace hyphens ("-"), slashes ("/") and asterisks ("*") in the queries with spaces. For example, "subunit 1 BRCA1 BRCA2 containing complex" is a variant of "subunit 1 BRCA1/BRCA2-containing complex".

Concept-based Query Normalisation

Our document ranking method is based on the Okapi model (Robertson et al., 1994). Many participant systems at the TREC Genomics track use the Okapi method for ranking documents with respect to their similarity to the query. However, there are two fundamental problems with using this model on TREC Genomic queries.

The first problem regards Okapi not differentiating between concept terms and general query terms

[3] `http://l2r.cs.uiuc.edu/~cogcomp/atool.php?tkey=SS`

[4] `http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene`

[5] `http://www.nlm.nih.gov/mesh`

in the query. For example, consider two documents, one containing the terms “mad cow disease” and “PrnP”, and the other containing the terms “role” and “PrnP”. Clearly the first document containing the two biological concepts is more relevant. The second problem occurs because TREC 2006 topics contain more than one concept term. It is possible that a short paragraph that discusses one concept only will be ranked higher than a longer paragraph which mentions two concepts. Again this is an undesirable outcome.

To overcome these problems, a *Conceptual IR* model was proposed in (Zhou et al., 2006b). In this paper we propose another method called the *concept-based query normalisation* which is based on the Okapi model and similar to the method introduced in (Li, 2007; Stokes et al., 2008) for geospatial IR.

The first problem is solved by dividing query terms into two types: *general terms* t_g and *concept terms* t_c. Given a query with both concept and general terms, the similarity between a query Q and a document D_d is measured as follows:

$$\mathit{sim}(Q, D_d) \quad = \quad \mathit{gsim}(Q, D_d) + \mathit{csim}(Q, D_d)$$

where $\mathit{gsim}(Q, D_d)$ is the *general similarity score* and $\mathit{csim}(Q, D_d)$ is the *concept similarity score*. The general similarity score is given by:

$$\mathit{gsim}(Q, D_d) \quad = \quad \sum_{t \in Q_g} \mathit{sim}_t(Q, D_d) = \sum_{t \in Q_g} r_{d,t} \cdot w_t \cdot r_{q,t}$$

where Q_g is the aggregation of all general terms/phrases in the query. The concept similarity score is given by:

$$\begin{aligned} \mathit{csim}(Q, D_d) \quad &= \quad \sum_{C \in Q_c} \mathit{sim}_c(Q, D_d) \\ &= \sum_{t \in C, C \in Q_c} \mathit{Norm}(\mathit{sim}_{t_1}(Q, D_d), \ldots, \mathit{sim}_{t_N}(Q, D_d)) \\ &= \sum_{t \in C, C \in Q_c} (\mathit{sim}_{t_1} + \frac{\mathit{sim}_{t_2}}{a} + \cdots + \frac{\mathit{sim}_{t_N}}{a^{N-1}}) \end{aligned}$$

where Q_c is the aggregation of all concepts in the query, C is one concept in Q_c, and t_i is a term/phrase in the query, after expansion, which belongs to the concept C; the t_i are listed in descending order according to their Okapi similarity scores sim_{t_1}, ..., sim_{t_N}:

$$\mathit{sim}_t(Q, D_d) \quad = \quad r_{d,t} \cdot w'_t \cdot r_{q,t}$$

where

$$\begin{aligned} r_{d,t} \quad &= \quad \frac{(k_1 + 1) \cdot f_{d,t}}{k_1 \cdot [(1 - b) + b \cdot \frac{W_d}{avgW_d}] + f_{d,t}} \\ w'_t \quad &= \quad \log \frac{N - \max(f_t, f_{t_q}) + 0.5}{\max(f_t, f_{t_q}) + 0.5} \\ r_{q,t} \quad &= \quad \frac{(k_3 + 1) \cdot f_{q,t}}{k_3 + f_{q,t}} \end{aligned} \qquad (1)$$

where k_1 and b are usually set to 1.2 and 0.75 respectively, and k_3 can be taken to be ∞. Variable W_d is the length of the document d in bytes; $avgW_d$ is the average document length in the entire collection; N is the total number of documents in the collection; f_t is the number of documents in which term t occurs; and $f_{\{d,q\},t}$ is the frequency of term t in either a document d or query q.

Note that (1) is an adjustment of the calculation for the weight w'_t of an *expansion* term t appearing in the query: for expansion term t, its own term frequency f_t and the corresponding original query term's frequency f_{t_q} are compared, and the larger value used — this ensures the term contributes an appropriately normalised “concept weight”.

To solve the second problem, we use the following rules to ensure that for two passages P_1 and P_2, where one contains more unique concepts than the other, the number of concepts *ConceptNum(P)* will override the Okapi score *Score(P)* and assign a higher rank to the passage with more unique concepts:

```
if $\mathit{ConceptNum}(P_1) > \mathit{ConceptNum}(P_2)$ then
    $\mathit{Rank}(P_1) > \mathit{Rank}(P_2)$
else if $\mathit{ConceptNum}(P_1) < \mathit{ConceptNum}(P_2)$ then
    $\mathit{Rank}(P_2) > \mathit{Rank}(P_1)$
else if $\mathit{Score}(P_1) \geq \mathit{Score}(P_2)$ then
    $\mathit{Rank}(P_1) > \mathit{Rank}(P_2)$
else
    $\mathit{Rank}(P_2) > \mathit{Rank}(P_1)$
```

Abbreviation Finder

Although MeSH and Entrez Gene contain many synonyms and related terms, one important type of lexical variant, *abbreviations*, has very low coverage in both databases. For example, "AD" is a commonly used abbreviation for "Alzheimer's Disease". Since the long and short form ("Alzheimer's Disease (AD)") only appear together at the beginning of each journal document, many relevant passages will contain "AD" only and so will appear less relevant than they should against a query containing "Alzheimer's Disease". Hence, expanding the given query with "AD" should improve retrieval effectiveness.

As already mentioned, there are two methods for collecting abbreviations from the literature: the first uses the static resource ADAM (Zhou et al., 2006a), while the second uses our pseudo relevance feedback method for extraction these abbreviations during run time. The advantage of the latter approach is that it dynamically collects abbreviations and so does not suffer from the coverage and update problems of static resources like ADAM. The following is an overview of how our abbreviation feedback step contributes to the retrieval process:

1. Retrieve the first 1000 documents which include at least one instance of each concept in the query.

2. From this subset of documents, find terms which fit the pattern "*Term* (*Abbr*)", where "*Term*" is a concept in the query (original or expanded) and "*Abbr*" is the abbreviation or synonym defined in the text.

3. Among all the detected abbreviations or synonyms, remove all the multi-word terms, terms that do not have any overlapping characters with the original term, and terms which occur less than three times.

4. For all remaining abbreviations or synonyms, use the above generation tool to formulate all their lexical variants, and add them to the query. The expanded query is then re-submitted to the retrieval engine, and the passage extraction step, described below, is applied.

Passage Extraction

As already mentioned the 2006 Genomics Track defined a new question answering-type task that requires short full-sentence answers to be retrieved in response to a particular query. However, before answer passages can be generated, we first retrieve the first 1000 ranked paragraphs for each topic, and use the following simple rules to reduce these paragraphs to answer spans.

Two methods are examined in this paper which are best described with an example. Given a paragraph consisting of a set of sentences $\{(s_1, i), (s_2, i), (s_3, r), (s_4, r), (s_5, i), (s_6, r), (s_7, i), (s_8, i), (s_9, r), (s_{10}, i)\}$, where r is relevant (that is, mentions at least one query term) and i is irrelevant. *Method A* shortens a paragraph by removing irrelevant sentences from its start and end until a relevant sentence is detected. Hence, it would produce the following passage of sentences: $\{(s_3, r), (s_4, r), (s_5, i), (s_6, r), (s_7, i), (s_8, i), (s_9, r)\}$.

This extraction method does not split a paragraph into multiple passages if irrelevant sentences occur within the resultant passage. *Method B*, on the other hand, addresses this issue by splitting a passage if there are two or more consecutive irrelevant sentences within this span. Hence, Method B would produce the following two passages for this paragraph: $\{(s_3, r), (s_4, r), (s_5, i), (s_6, r)\}$ and $\{(s_9, r)\}$.

After one of these passage extraction techniques has been applied for a particular topic, we re-rank passages by re-indexing them, and re-querying the topic against this new index, using the global statistics from the original indexed collection, i.e. using term frequency f_t and the average paragraph length $avgW_d$.

4 Experimental Methodology

4.1 Data and Evaluation Metrics

We used the TREC 2006 Genomics Track evaluation resources to determine the effectiveness of our system. The TREC 2006 collection consists of 162,259 full-text documents from 49 journals publish electronically via the Highwire Press website[6]. The track also provided 28 topics expressed as natural

[6] More information on the TREC dataset can be found at: `http://ir.ohsu.edu/genomics/2006data.html`

language questions, formatted with respect to seven general topic templates. Participants were asked to submit the first 1,000 ranked passages returned by their system for each of the topics (Hersh et al., 2006). Passages in this task are defined as text sequences that cannot cross paragraph boundaries (delimited by HTML tags), and are subsets of the original paragraphs in which they occur. As is the custom at TREC, human judges were used to decide the relevance of passages in the pooled participating system results. These judges also defined exact passage boundaries, and assigned topic tags called *aspects* from a control vocabulary of MeSH terms to each relevant answer retrieved.

Mean Average Precision, or MAP, is a popular IR metric for evaluating system effectiveness. The TREC Genomics Track defines three versions of the MAP score calculated at various levels of granularity: *Document*, *Passage* and *Aspect*. Traditionally the MAP score is defined as follows: first, the average of all the precision values at each recall point on a topic's *document* ranked list is calculated; then, the mean of all the topic average precisions is determined. Since the retrieval task at the Genomics Track is a question answering-style task, a metric that is sensitive to the length of the answer retrieved was developed.

Passage MAP is similar to document MAP except average precision is calculated as the fraction of characters in the system passage overlapping with the gold standard answer, divided by the total number of characters in every passage retrieved up to that point in the ranked list. Hence, a system is penalised for all additional characters retrieved that are not members of the human evaluated answer passage.

The TREC organisers also wanted to measure to what extent a particular passage captured all the necessary information required in the answer. Judges were asked to assign at least one MeSH heading to all relevant passages. Aspect average precision is then measured as the number of aspects (MeSH headings) captured by all the relevant documents up to the recall point in the ranked list for a particular query. Relevant passages that did not contribute any new aspect to the aspects retrieved by higher ranked passages were removed from the ranking. Aspect MAP is defined as the mean of these average topic precision scores.

4.2 Experimental Results

In this section, we examine the increased effectiveness obtained when different expansion information is added to the original query. We also evaluate the effect of our proposed abbreviation feedback technique, and our novel answer expansion module, on system performance.

As explained in Section 3, our system uses Entrez Gene for expansion of genes to their synonymous instances. In addition, all term variants are generated for their abbreviations as described in Section 3, while other biological entities in the query (e.g., diseases) are expanded using MeSH. Table 1 presents the MAP scores for the following system runs:

- `Baseline`: Zettair system with no expansion
- `SYN`: query expansion using Entrez gene and MeSH expansion (*Synonym* and *See Also* entries in MeSH) of query terms
- `SYN+HYPO`: query expansion using Entrez gene and MeSH expansion, including *Hyponyms* (i.e., specialisations)
- `SYN+HYPER`: query expansion using Entrez gene and MeSH expansion, including *Hypernyms* (i.e., generalisations)
- `SYN+HYPER+VAR`: query expansion using Entrez gene, *Gene Variant Generation*, and MeSH expansion, including *Hypernyms*

All expansion run MAP scores show a statistically significant[7] improvement over the baseline MAP. The only expansion experiment that does not incrementally improve the results is the addition of hyponym terms (i.e. specialisation) from MeSH. On the other hand, hypernyms (i.e. generalisations) improve the performance of the `SYN` run by nearly 5%. This result may be explained by the fact that at a passage level, generalised expressions are commonly used to refer to query terms that have been discuss earlier in the document. For example, the following sentence is clearly relevant to the *mad cow disease* query presented in Section 3: "These *prion diseases* are characterised by the accumulation of an abnormal (aberrantly folded) isoform of a cellular host

[7] We use a paired Wilcoxon signed-rank test at the 0.05 confidence level to determine significance.

Table 1: Table showing improvement in MAP score obtained over baseline MAP when the query is expanded with various combinations of related terms: synonyms (SYN), hyponyms (HYPO), hypernyms (HYPER) and gene lexical variants (VAR)

Run	Passage MAP			Aspect MAP			Document MAP		
`Baseline`	0.0480			0.1838			0.3355		
`SYN`	0.0888†	+85.0%	$P = 0.005$	0.3499†	+90.3%	$P < 0.001$	0.4711†	+40.4%	$P = 0.008$
`SYN+HYPO`	0.0878†	+83.0%	$P = 0.007$	0.3417†	+85.9%	$P = 0.001$	0.4632†	+38.1%	$P = 0.02$
`SYN+HYPER`	0.0933†	+94.4%	$P < 0.001$	0.3695†	+101%	$P < 0.001$	0.4843†	+44.3%	$P = 0.002$
`SYN+HYPER+VAR`	0.0949†	+97.6%	$P < 0.001$	0.3827†	+108%	$P < 0.001$	0.5080†	+51.4%	$P < 0.001$

protein PrPC". However, it would only be ranked highly if the generalisation relationship from *mad cow disease* to *prion disease* has been established. Expanding the query term with the immediate parent terms in the different MeSH hierarchies usually results in a few focussed terms being added to the query. In contrast, adding specialisations may result in a much larger number of term additions, depending on the generality of the query term. For example, the term *neurons* has 18 unique subcategories one level below its position in the MeSH hierarchy and many more beyond this level.

Our best system run (SYN+HYPER+VAR) used ontological and gene variant expansion, and achieved a 97.6% increase in Passage MAP over the baseline run. Similarly large increases in Aspect and Document MAP were also observed. A detailed analysis showed that many passages had been either missed or ranked lower than expected by our system due to the occurrence of query term abbreviations in the relevant passage. These abbreviations were not captured in either of our ontological resources.

Table 2 compares the performance of the two abbreviation expansion strategies described in Section 3. Ontological expansion using the ADAM abbreviation database reduces our best Passage MAP score by 36%. Our abbreviation feedback loop performs better, producing a small increase in Document MAP over the baseline, but slightly lower Aspect and Passage MAPs. In some respects, this feedback result is disappointing as a manual analysis of the added abbreviations shows that many useful synonyms were added to the query, which should, in theory, help to retrieve additional passages and boast the rankings of other relevant passages.

However, there is one big drawback to abbreviation expansion that isn't characteristic in other types of expansion we have explored: abbreviations are much more ambiguous. For example, the abbreviation "AD" is a very commonly used reference to "Alzheimer's disease"; however, according to ADAM, "AD" has 35 unique long forms defined in MedLINE abstracts. For example, "AD" can also refer to the phrases "after discharge", "autosomal dominant", "autistic disorder", and other unrelated concepts.

IR researchers have found that query-term ambiguity is less of a problem than one might expect because of the *query term collocation effect* (Krovetz and Croft, 1992): query terms mutually disambiguate each other because their intended senses tend to co-occur together in relevant documents in the collection. For example, for the query term "cell", adding the term "blood" to the query ensures that documents using the biological sense are ranked higher. Hence, one would expect that despite abbreviation ambiguity, great gains in IR effectiveness would be possible using expansion. However, when the total number of possible unabbreviated forms is factored into the expansion process, it is clear that an excessive amount of ambiguity is added in.

A manual analysis of the results backs up this observation: although new relevant passages containing abbreviations are being retrieved, paragraph ranking is being affected to such an extent that previously retrieved passages are "dropping out" of the top 1000 items in the ranked list.

However, our results also show that dynamic abbreviation expansion does not degrade performance as dramatically as expansion with ADAM. The feedback process ensures that only abbreviations that occur in documents of high ranked passages, mentioning all query concepts, are added to the query. Thus, these abbreviations have the highest potential for providing positive impact on retrieval effectiveness.

Table 2: Table showing effect on system performance when additional expansion terms are added from the ADAM abbreviation (+Adam) database and our system Abbreviation feedback loop (+Abbr).

Run	Passage MAP			Aspect MAP			Document MAP		
SYN+HYPER+VAR	0.0949			0.3827			0.5080		
SYN+HYPER+VAR+Adam	0.0600†	-36.8%	$P < 0.001$	0.2387†	-37.6%	$P < 0.001$	0.4105†	-19.2%	$P = 0.001$
SYN+HYPER+VAR+Abbr	0.0920	-3.06%	$P = 0.3$	0.3784	-1.12%	$P = 0.4$	0.5171	$+1.79\%$	$P = 0.3$

Table 3: Table showing effect of two passage extraction strategies A and B on system performance

Run	Passage MAP			Aspect MAP			Document MAP		
Best	0.0920			0.3784			0.5171		
Best+A	0.1100†	$+19.6\%$	$P < 0.001$	0.3673	-2.93%	$P = 0.3$	0.5123	-0.93%	$P = 0.3$
Best+B	0.1175†	$+27.7\%$	$P < 0.001$	0.3518†	-7.03%	$P = 0.004$	0.5021	-2.90%	$P = 0.08$

The general conclusion from these abbreviation expansion experiments is clear: knowledge of these synonymous instances is obviously beneficial, but a method that reduces the impact of their high ambiguity is necessary. We discuss our proposed solution to this problem in Section 5.

Our final experiment (see Table 3) shows that the TREC's Passage MAP score can be increased by capturing the exact answer span in each relevant paragraph. Section 3 proposed two methods for achieving this: *Method A* finds the longest text span in paragraph that contains all query terms; *Method B* splits the span and remove sentences if there is a distance of one or more sentences between consecutive mentions of any of the query terms. Both reduction methods show improvements in Passage MAP, but at the expense of the other two metrics. This is to be expected, especially in the case of *Method B*, since splitting paragraphs means some relevant passages may get a lower rank or even drop out of the top 1000 passages.

Table 4 shows how our best run (Best+B) performs with respect to systems that participated in the official TREC 2006 Genomics Track. TREC_MEDIAN is the median value for each MAP score reported at TREC. UIC_TREC[8] was the top performing system submitted by the University of Illinois at Chicago, and UIC_SIGIR is the best post-submission Passage MAP score which was also published by the same group (Zhou et al., 2007). If our system had participated at TREC track we would have ranked 6th for Passage MAP, 3rd for Aspect MAP and 4th for Document MAP out of 92 submitted runs.

Table 4: Table showing performance of our best Passage MAP scoring run Best+B with the top performing TREC systems on the Genomics Track

Run	Passage	Aspect	Document
	MAP	MAP	MAP
UIC_SIGIR	0.1823	0.3811	0.5391
UIC_TREC	0.1479	0.3492	0.5320
Best+B	0.1175	0.3518	0.5021
TREC_MEDIAN	0.0345	0.1581	0.3083

5 Discussion and Conclusions

The most successful systems at the TREC Genomics Track 2006 used a combination of expansion techniques from external resources such as publically-available and hand-crafted thesauri, in addition to lexical variant generation techniques similar to the one described in this paper. One of the principal contributions of this paper is our detailed analysis of what types of ontologically related terms (synonyms, hyponyms, hypernyms, lexical variants, abbreviations) provide the most impact when used as expansion terms. In particular, we have focussed on abbreviation expansion, which has high potential for impact when passages rather than full documents are being retrieved. However, our experiments show that their high ambiguity can in some cases reduce retrieval effectiveness.

There are two possible solutions to the abbreviation ambiguity problem: all abbreviations in the collection are identified in advance of indexing, and a unique identifier is assigned to each long-form and its corresponding abbreviated short-form. Hence, when the query is expanded, the unique identifier rather than the lexical form of the abbreviation is added to the query. Similarly, all abbreviations in the collection will be replaced by their identifier be-

[8] The official name for this run was UICGenRun3.

fore passage indexing occurs. Another possible approach would be to explicitly add the long-forms of abbreviations in a passage to its index entry. This is a document expansion rather than a query expansion strategy. We plan to investigate both of these methods in our future work.

Another area for potential improvement that we wish to investigate further is paragraph reduction. Passage MAP is severely affected by long-answer text spans. Paragraph reduction is similar to answer extraction in factoid-based Question-Answering tasks. However, researchers have only recently begun to investigate answer extraction for more complex question types such as *Why* or *How* questions in an ad hoc retrieval setting (Allan, 2005). The Document Understanding Conference (DUC), which focusses on summarisation tasks, is also looking at complex questions; however, answers are typically generated by collating information from multiple documents (Dang, 2006).

References

J. Allan. 2005. Hard track overview in TREC 2005: High accuracy retrieval from documents. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*.

S. Buttcher, C.L.A. Clarke, and G.V. Cormack. 2004. Domain-specific synonym expansion and validation for biomedical information retrieval. In *The Thirteen Text REtrieval Conference (TREC 2004) Proceedings*.

H.T. Dang. 2006. Overview of duc 2006. In *The Document Understanding Conference Workshop Proceedings*.

E. N. Efthimiadis. 1996. Query expansion. *Annual Review of Information Science and Technology*, 31:121–187.

W. Hersh, A. Cohen, P. Roberts, and H. Rekapalli. 2006. Trec 2006 genomics track overview. (Voorhees and Buckland, 2006).

Robert Krovetz and W. Bruce Croft. 1992. Lexical ambiguity and information retrieval. *Information Systems*, 10(2):115–141.

Yi Li. 2007. Probabilistic toponym resolution and geographic indexing and querying. Masters thesis, The University of Melbourne.

Jimmy Lin and Dina Demner-Fushman. 2006. The role of knowledge in conceptual retrieval: a study in the domain of clinical medicine. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 99–106.

S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. In *The Third Text Retrieval Conference (TREC 3) Proceedings*, Gaithersburg, Maryland, November.

I. Ruthven and M. Lalmas. 2003. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145.

Nicola Stokes, Yi Li, Alistair Moffat, and Jiawen Rong. 2008. An empirical study of the effects of NLP components on Geographic IR performance. *International Journal of Geographical Information Science*. To appear.

D. Trieschnigg, W. Kraaij, and F. de Jong. 2006. The influence of basic tokenization on biomedical document retrieval. In *SIGIR 2007 Proceedings*, Amsterdam, The Netherlands, July.

E. M. Voorhees and Lori P. Buckland. 2006. *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*. NIST, Gaithersburg, Maryland.

W. Zhou, V. I. Torvik, and N. R. Smalheiser. 2006a. ADAM: another database of abbreviations in MEDLINE. *Bioinformatics*, 22(22):2813–2818.

W. Zhou, C. Yu, V. Tovik, and N. Smalheiser. 2006b. A concept-based framework for passage retrieval in genomics. (Voorhees and Buckland, 2006).

Wei Zhou, Clement Yu, Neil Smalheiser, Vetle Torvik, and Jie Hong. 2007. Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 655–662, New York, NY, USA. ACM Press.