Expanding Queries Using Multiple Resources

The AID Group at TREC 2006: Genomics Track

Edgar Meij[1] **Machiel Jansen**[2,*] **Maarten de Rijke**[1]

[1] ISLA, University of Amsterdam
emeij, mdr@science.uva.nl
[2] Free University Amsterdam
mgjansen@few.vu.nl

Abstract: We describe our participation in the TREC 2006 Genomics track, in which our main focus was on query expansion. We hypothesized that applying query expansion techniques would help us both to identify and retrieve synonymous terms, and to cope with ambiguity. To this end, we developed several collection-specific as well as online strategies. Our proposed methods yield a noticeable improvement in retrieval performance over the baseline. To counter the negative effects of query expansion on recall, we introduce conjunctive Boolean constraints on the query terms and added expansion terms. When these additional constraints are imposed, results improve even further. The improvements in our results are noticeable on the document, passage, as well as aspect level.

1 Introduction

In this paper we describe our participation in the TREC 2006 Genomics track. One of our working hypotheses is that finding synonymous terms while, at the same time, coping with ambiguous terms is essential. We propose several methods, collection-specific as well as web-based, to identify synonymous terms. To counter any possible query drift, we impose additional constraints on the expansion terms we find.

The remainder of this paper is organized as follows. In Section 2 we describe our collection and query preprocessing, passage identification, and the retrieval model employed for this year's edition of TREC Genomics. We then elaborate on our proposed query expansion techniques in Section 3, and follow with a description of our submitted runs in Section 4. The results can be found in Section 5, together with a more in-depth analysis of some individual topics. We summarize our findings in a concluding section.

*Currently at Collexis, e-mail: jansen@collexis.com

2 Experimental Setup

In this section we elaborate on the particular tools, methods and models used for indexing and retrieving. All of our runs are created using Lucene [13] with a Language Modeling extension that we developed in-house [9]; it uses a multinomial language model.

2.1 Language Modeling

We estimated a language model for each document in the collection and for any given query we rank the documents with respect to the likelihood that the document language model generated the query:

$$P(d|q) \propto P(d) \cdot \prod_{t \in q} P(t|d), \tag{1}$$

where d is a document and t is a term in query q. In the implemented scoring formula the probabilities are reduced to rank-equivalent logs of probabilities. To account for data sparseness, we interpolate the likelihood $P(t|d)$ using Jelinek-Mercer smoothing [6, 19, 20]. This can be viewed as estimating the probability

$$P(d|q) = P(d) \cdot \prod_{t \in q} \left((1-\lambda) \cdot P(t|D) + \lambda \cdot P(t|d) \right), \tag{2}$$

where D is the collection. We need to estimate three probabilities: the prior probability of the document, $P(d)$; the probability of observing a term in a document, $P(t|d)$; and the probability of observing the term in the collection, $P(t|D)$. We assume the query terms to be independent, and use a linear interpolation of a document model and a collection model to estimate the probability of a query term.

The probabilities are estimated using maximum likelihood estimates:

$$P(t|d) \quad = \quad \frac{tf(t,d)}{|d|}, \tag{3}$$

$$P(t|D) = \frac{df(t,D)}{\sum_{t' \in D} df(t',D)}, \quad (4)$$

$$P(d) = \frac{|d|}{\sum_{d' \in D} |d'|}, \quad (5)$$

where $tf(t,d)$ is the termfrequency of term t in document d; $df(t)$ is the count of documents in which term t occurs, and $|d|$ denotes the length of a document d [4].

2.2 Collection Preprocessing

The 2006 Genomics document collection consists of 162,259 full-text biomedical articles, which were preprocessed as follows:

1. replace HTML entities with their ISO-Latin1 counterparts,
2. remove HTML tags,
3. remove top-level tables; these only serve navigational purposes,
4. remove citations within text,
5. cut-off article before the *References* or *Acknowledgements* section,
6. lowercase terms, and
7. remove stopwords.

We do not apply any form of stemming. The used stopword list also includes collection-specific common terms such as "figure," "table," "view," "larger," "version," etc.

2.3 Passage Identification

This year's Genomics track introduced a novel task. Unlike previous years, participants were requested to return relevant passages instead of entire documents and the systems are judged based on 3 levels of granularity: returned documents, passages, and aspects. We have chosen to mainly focus on documents and passages for our participation.

We experimented with various ways of identifying passages, and decided to consider every sentence as being a passage—which we identify using Lingpipe's sentence extractor [1]. Every sentence gets indexed as a separate document and we include positional information and the originating PubMed ID in the index. The intuition behind it is that this approach should give us a result which yields relatively high precision, since query terms should appear within the same sentence.

There are 37.5 million unique sentences, with a mean length of 14.7 tokens. The distribution is slightly skewed, with a median of 13 tokens. Figure 1 shows a histogram of the indexed document lengths. The risk with parameter estimation using maximum likelihood estimates in Equation 2, is the underestimation of unseen or rare terms and overestimation of frequently occuring ones. Especially when dealing with very short documents (such as sentences) this bias becomes clearly visible.

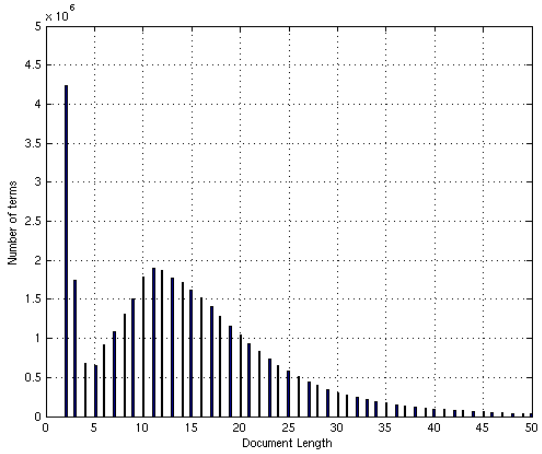

Figure 1: Histogram of passage lengths.

To compensate, we smooth our language model in Equation 2 using collection frequencies rather than document frequencies [19]. The parameter λ is the smoothing parameter, which can be optimized using training data. Since this is the first year in which the current collection is being used, such training data is unavailable. We therefore use the standard value of 0.15, as described by Hiemstra and Kraaij [7].

Singhal et al. [17] have shown that for ad-hoc retrieval, there is a clear correlation between the a-priori probability of relevance and the length of a document. Passage retrieval, using sentence boundaries as denominators, is closely related to XML retrieval. In a sense, the performed passage identification makes the extracted sentences our *units of retrieval*. Kamps et al. [10] confirm the relation between a-priori relevance and XML element length and, additionally, introduce a length prior β. We follow their approach and estimate the prior probability of a sentence being relevant as proportional to the length of a sentence. Hence, the implemented scoring formula for a document d and query $t_1, \ldots, t_n$ becomes:

$$s(d,t_1,\ldots,t_n) = \beta \cdot \log\left(\sum_t tf(t,d)\right) + \sum_{i=1}^{n} \log\left(1 + \frac{\lambda \cdot tf(t_i,d) \cdot \left(\sum_t df(t)\right)}{(1-\lambda) \cdot df(t_i) \cdot \left(\sum_t tf(t,d)\right)}\right). \quad (6)$$

Our extension of Lucene includes the tunable length prior β, which we set to 2.

Example topic.	
Run	**Query**
	What is the role of PrnP in mad cow disease?
Aspects:	PrnP “mad cow disease”
`UAmsBaseLine`	PrnP “mad cow disease” “prion protein gene” “prion protein” Prn-P “Prn P”
`UAmsExp`	PrnP “mad cow disease” “prion protein gene” “prion protein” Prn-P “Prn P” prp Pr-P PrPC BSE “bovine spongiform encephalopathy” $[expansions]_{PrnP}$ $[expansions]_{mad\_cow\_disease}$ *etc.*
`UAmsExpSel`	(PrnP $[expansions]_{PrnP}$) AND (“mad cow disease” $[expansions]_{mad\_cow\_disease}$)

Table 1: Topic 160 showing the nature of the various query expansions for our submitted runs. For reasons of brevity, we replace the full list of expansions for a term with $[expansions]_{term}$. OR is the default operator between query terms.

2.4 Query Preprocessing

We use the Genia parser [3, 12] to syntactically parse the topics and extract all noun phrases (NPs) and headwords, thus identifying all relevant *aspects* from the query [2]. All topics follow a certain topic template, so each contains one or more biological concepts and processes and some explicit relationship between them. We identify the biological *subject(s)* and *object(s)* of the query and discard the relationship term(s).

The resulting query aspects are kept as phrases for subsequent query expansion. Phrases are reported to improve retrieval results when compared to single-word indexing [14, 15], and we believe this is also the case in biomedical IR. We implemented phrase support in our language model through an *n*-gram based index. We elaborate on the used query expansion algorithms in the next section.

3 Query Expansion

The fact that for any given biomedical concept there are frequently occuring spelling variations and synonyms degrades the performance of regular adhoc IR techniques. To overcome this problem, we propose different forms of collection-specific and online query expansion methods, based on the hypothesis that proper handling of synonymous terms is essential in biomedical text retrieval.

The methods we propose include using acronyms and their corresponding long forms from the collection, the matching of related long forms, and the online lookup of unknown query terms and gene names. Query expansion is performed on the basis of the extracted NPs from the query as described in Section 2.4. We also include the breakpoint algorithm as introduced by Huang et al. [8]. In the following sections we elaborate on our query expansion strategies.

As is common with using query expansion in general, one is likely to improve recall at the cost of precision [11, 18]. Some of the added synonyms, acronyms, or long forms for a particular query term might be identical to other biomedical concepts (e.g., diseases or methods, where the query term is a gene name). Including all possible expansions in the query will therefore result in higher recall but also in more noise. Our intuition was that the high-precision approach to passage identification, as described in Section 2.3, would compensate for any query drift.

3.1 Corpus-Specific Acronym Identification

We mine acronyms and their corresponding forms directly from the documents in the collection, using the algorithm described by Schwartz and Hearst [16]. We adapted their approach in order to also collect frequency information. All found acronyms, long forms and frequencies were stored in a database, with an acronym being defined as a term with a maximum length of 6 characters and containing at least one uppercase character. For every query aspect, we check whether it is an acronym, and proceed with different approaches, depending on whether the term is indeed an acronym.

3.2 Acronyms

If the term is indeed an acronym, we look up all possible long forms in the database and add all results with a frequency of more than 1 to the query.

In addition, we also look up alternative acronyms for a given acronym. These are identified as follows: A list was made for all long forms of every acronym. The most frequent long form with a *different* acronym is identified and the acronym is selected for addition to the query. For example, the most commonly used long form for the term PrnP is “prion protein gene.” The most commonly used acronym for this long form is not PrnP, but prp. We hesitated to put the alternative longer form in the query as well, but chose not to do so.

3.3 Long Forms

If the NP is not an acronym, we check whether it *has* one or more acronyms. Again, all resulting acronyms with a frequency greater than 1 are added to the query.

Acronyms related to a given long form are also searched for in the database. For all long forms we check if they

Results of the submitted runs.						
	MAP					
Run	Document		Passage		Aspect	
UAmsBaseLine	0.1624		0.0226		0.0457	
UAmsExp	0.2081	+28.13%	0.0285	+26.31%	0.0495	+8.35%
UAmsExpSel	**0.232**	+42.36%	**0.0484**	+114.04%	**0.114**	+148.90%

Table 2: Results of our submitted runs. Best scores are in boldface and the percentage improvements over the baseline are indicated.

occur as a substring in other long forms. If so, these *related* long forms, together with their most frequent acronym, were returned. As an example "Alzheimer's disease" produces "fad" and "Familiar Alzheimer's disease" as related acronym and long form respectively.

Finally, for long forms which don't have a long form in the database, we turn to Google. These long forms were submitted to the search engine, prefixed with the *define* operator, and acronyms that occurred more than once in the snippets returned were used as query expansion terms, together with their long forms. This helped us, for example, to find the acronym BSE for "mad cow disease" and PCD for "apoptosis."

3.4 Breakpoints

We implemented the breakpoint algorithm as described by Huang et al. [8] and used in last year's TREC Genomics track. The generated breakpoint alternatives are looked up in the database in the same fashion as regular acronyms. Only if something was returned the breakpoint alternative gets added to the query. For example: "Prn-P" "Prn P" "Pr P" "Pr-P" are breakpoint alternatives for "PrnP" but none are selected for addition because none of these occur in the collection.

3.5 Gene Name Expansion

For Gene names we propose a different algorithm. A gene name is defined as a long form or an NP that ends in "gene[s]". It turns out that all identified acronyms that don't have an entry in the database are indeed gene names. We look these up on BioInformatics.org[1] and again mine the output for (synonymous) acronyms. All those that start with a bracket or a digit are discarded; the resulting ones are added to the query.

4 Runs

We submitted the following three automatic runs.

[1]BioInformatics http://bioinformatics.org/textknowledge/synonym.php

UAmsBaseLine A baseline run, using collection-specific acronym and long-form expansion on identified NPs from the queries and breakpoint variant generation.

UAmsExp Same as UAmsBaseLine, with additional query expansion: substring matching, gene name expansion, alternative acronyms, and Google define.

UAmsExpSel Same as UAmsExp, but with imposed restrictions on the occurances of query aspects. UAmsExp is geared towards achieving high recall through a Boolean OR type of expansion, whilst UAmsExpSel is focused on achieving high precison. To improve precision, we put Boolean AND constraints on the identified query aspects, as proposed by Hearst [5].

Table 1 shows an example topic, to help clarify the proposed query expansion techniques.

5 Results

Table 2 displays the results of our submitted runs (best scores in boldface). Expanding the queries using the proposed algorithms has a clear postive effect on retrieval effectiveness, as compared to the baseline. The addition of the Boolean AND constraints is able to improve the results even further. When compared to the rest of the participants, our best results are well above the median.

5.1 Topic Analysis

Figure 2 gives a graphical representation of the per-topic breakdown of the scores of our best performing run (UAmsExpSel), as compared to our baseline (UAmsBaseLine). The drop in performance for topics 162 and 170 on all performance measures can be explained through the generality of the query terms. Topic 162 contains "cancer" and topic 170 contains "endoplasmic reticulum," which are expanded with equally broad terms. The retrieved passages are therefore not always on topic.

We also note that our method of identifying passages from the source documents leaves room for improvement. Query terms that are in two consecutive sentences, for example, are not considered to be in the same passage and thus not retrieved. This effect can be observed in the document scores

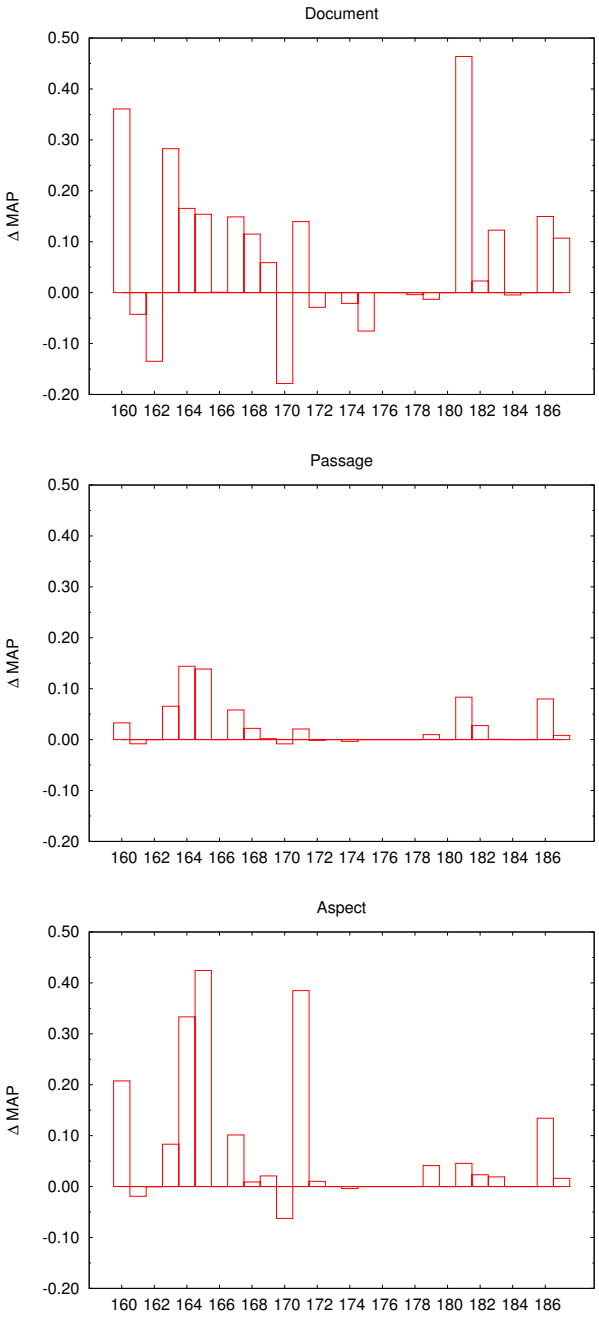

Figure 2: Per-topic breakdown of the results of `UAmsExpSel`, as compared to the baseline: Document MAP (top), Passage MAP (middle), and Aspect MAP(bottom).

for the two topics mentioned earlier; the negative impact on passage and aspect scores for these topics is much lower. Our choice in determining passage boundaries also has its effect on the passage scores; the improvements over baseline are lower here as compared to the improvement in document scores.

When zooming in on the document scores, there are three distinct peaks for topics 160, 163, and 181. These topics contain "mad cow disease," "Alzheimer's disease," and "colon cancer," respectively. Contrary to the earlier mentioned and worse performing *broad* terms, each of these query aspects gets expanded properly. As indicated earlier in Section 3.3, the use of a generic web search engine helps us to find the proper acronym for "mad cow disease," and substring matching helps us to find the more common long form for "Alzheimer's disease," namely "familial Alzheimer's disease," together with its acronym FAD.

We did not pursue any specific goal regarding aspect retrieval. However, our overall aspect scores indicate that the proposed query expansion strategies are also beneficial when looking only at retrieved aspects.

6 Conclusions

We have described our participation in the TREC 2006 Genomics track. Our aim this year was to apply query expansion techniques in order to find synonymous terms. First off, we used a POS tagger to identify noun phrases (or *aspects*) in the topics. Next, the identified query aspects were expanded using various query expansion strategies, based on collection-specific as well as online algorithms. The application of these methods yielded a noticeable improvement in retrieval performance over the baseline.

We used sentence boundaries within the original source documents as our passage boundaries, under the assumption that this would yield results with relatively high precision scores. To counter the negative effects of query expansion on recall further, we also introduced Boolean AND constraints on the identified query aspects. Indeed, when these additional constraints are imposed, the results improve even more. We believe, however, that our way of identifying passages leaves room for improvement and we intend to experiment further with various forms of passage identification.

Acknowledgements

We would like to thank Willem van Hage and Sophia Katrenko for their contributions and many insightful discussions. This work was carried out in the context of the Virtual Laboratory for e-Science project (`http://www.vl-e.nl`). This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ). Maarten de Rijke was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001.

7 References

[1] Alias–i. Lingpipe - a suite of java libraries for the linguistic analysis of human language., 2005. http://www.alias-i.com/lingpipe/.

[2] C. Buckley. Why current IR engines fail. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 584–585, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-881-4.

[3] A. Clegg and A. Shepherd. Evaluating and integrating treebank parsers on a biomedical corpus. In *Proceedings of the Association for Computational Linguistics Workshop on Software 2005*, 2005.

[4] W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2003. ISBN 1402012160.

[5] M. A. Hearst. Improving full-text precision on short queries using simple constraints. In *Proceedings of the Symposium on Document analysis and Information Retrieval*, Alexis Park Resort, Las Vegas, Nevada, April 1996.

[6] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

[7] D. Hiemstra and W. Kraaij. Twenty-One at TREC7: Ad-hoc and Cross-Language Track. In *TREC*, pages 174–185, 1998.

[8] X. Huang, Z. Ming, and L. Si. York university at trec 2005: Genomics track. In *Proceedings of the 14th Text Retrieval Conference*, 2005.

[9] ILPS. The ILPS extension of the Lucene search engine, 2005. http://ilps.science.uva.nl/Resources/.

[10] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in xml retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 80–87, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-881-4.

[11] J. Kekäläinen and K. Järvelin. The impact of query structure and query expansion on retrieval performance. In W. B. Croft, A. Moffat, C. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIRConference on Research and Development inInformation Retrieval*, pages 130–137, Melbourne, Australia, Aug. 1998. ACM Press, New York.

[12] J. D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus–semantically annotated corpus for bio-textmining. In *Bioinformatics*, volume 19 Suppl 1, pages 180–182, CREST, Japan Science and Technology Corporation, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan., 2003. URL http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/.

[13] Lucene. The Lucene search engine, 2005. http://jakarta.apache.org/lucene/.

[14] G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In D. E. Losada and J. M. Fernández-Luna, editors, *ECIR*, volume 3408 of *Lecture Notes in Computer Science*, pages 502–516. Springer, 2005. ISBN 3-540-25295-9.

[15] J. Pickens and W. B. Croft. An exploratory analysis of phrases in text retrieval. In *Proceedings of RIAO (Recherche d'Information assiste par Ordinateur)*, pages 1179–1195, 2000.

[16] A. Schwartz and M. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2003)*, 2003. URL citeseer.ist.psu.edu/schwartz03simple.html.

[17] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-792-8.

[18] E. M. Voorhees. Query expansion using lexical-semantic relations. In W. B. Croft and C. J. van Rijsbergen, editors, *SIGIR*, pages 61–69. ACM/Springer, 1994. ISBN 3-540-19889-X.

[19] H. Zaragoza, D. Hiemstra, and M. Tipping. Bayesian extension to the language model for ad hoc information retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 4–9, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-646-3.

[20] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-331-6.