TREC 2004 Genomics Track Experiments at UTA: The Effects of Primary Keys, Bigram Phrases and Query Expansion on Retrieval Performance

Ari Pirkola
University of Tampere (UTA), Finland
Department of Information Studies
pirkola@cc.jyu.fi

Abstract We submitted runs for Genomics Track's ad hoc retrieval task. The first official run (*utaauto*) was an automatic run and the second (*utamanu*) manual. For *utaauto,* the main features of query formulation were the removal of performative and marginally topical words from the topics based on average term frequency statistics, the removal of stop-words, the identification of bigram phrases, the weighting of keys with low document frequency (which we call primary keys), and query expansion with MH terms. The *utamanu* queries were Boolean queries formulated on a basis of a concept analysis. New terms were selected from the MeSH, genome databases, and dictionaries. The mean average precision (MAP) for the *utaauto* queries was 0.3324 and for the *utamanu* queries 0.3128. In the additional experiments we studied the effects of utaauto's main features on retrieval performance. The results showed that assigning more weight to the primary key than the other keys of a query improves retrieval performance. The use of bigram phrases in queries was also useful. The bigram phrases were identified using a collocation technique that computes an adjacency indicator for words in topics based on the joint occurrences of the words in a large and small window of document text words. We call the technique *relative key adjacency* (*RKA*).

1. Introduction

University of Tampere (UTA) participated in the Genomics Track and submitted runs for the ad hoc task. The first official run (*utaauto*) was an automatic run. The second official run (*utamanu*) was manual. We also did several additional experiments, and in this paper we describe our official and additional experiments.

The test system was the *InQuery* retrieval system, a probabilistic retrieval system based on the Bayesian inference net model. We indexed the Medline fields Title, Abstract, MH, RN, and GS. In addition to the main index there was a MH-index which allowed MH-field based retrieval. The mean average precision (MAP) for *utaauto* was 0.3324 and for *utamanu* 0.3128.

The structure of the rest of this paper is as follows. For bigram phrase identification we used a collocation technique which we call *relative key adjacency* (*RKA*). The technique is presented in Section 2.1. Section 2.2 presents the indexing methods and the processing of query keys. Section 2.3 considers the retrieval system. Query formulation and the main features of the query types considered in the study are presented in Sections 2.4 and 2.5. Section 3 contains the results and conclusions.

2. Methods and data

2.1 The RKA technique

The *relative key adjacency* (*RKA*) technique computes an adjacency indicator for words in topics based on the joint occurrences of the words in a large and small window of document text words. The function RKA may be defined as follows:

Let l and k denote two keys of a topic. Let *windocs*(N, l, k) denote the number of documents where the key l co-occurs with the key k in a text window of N words. The function RKA(N, l, k) gives the *relative key adjacency* for the keys l and k.

$$\mathrm{RKA}(N, l, k) = \mathit{windocs}(2^*N, l, k) \,/\, \mathit{windocs}(N, l, k)$$

The more there are documents where l and k co-occur in a smaller window the more closely the keys are related with one another. On the other hand, the more there are documents where l and k only co-occur in a larger window the more distantly the keys are related.

In this study, RKA was computed for all two consecutive keys (bigrams) in the topics. The value of the parameter N was set at 10. Thus, we used the windows of 10 and 20 words in determining the RKA values.

Table 1 shows RKA values for the bigrams in the Need field of the Topic # 02 (*Find protocols for generating transgenic mice*). RKAs were computed in the Genomics Track's collection. After the removal of performative words and stop-words, and after word normalization, the words *protocol*, *generate*, *transgenic* and *mice* were kept in the query, and RKAs were computed for the bigrams *protocol generate*, *generate transgenic,* and *transgenic mice.*

Table 1. RKA values for the words of the Topic # 02.

Bigram (l, k)	RKA(10, l, k)
protocol generate	1.77
generate transgenic	1.22
transgenic mice	1.01

As shown in Table 1 the RKA of the bigram *transgenic mice* is 1.01. The low value indicates that the words *transgenic* and *mice* are closely related to each other. Generally the values of 1.00-1.15 are typical of fixed phrases. As can be seen the bigrams *protocol generate* and *generate transgenic* give higher RKAs.

The benefits of the RKA technique is that it is simple and easy to integrate in retrieval systems. We implemented the RKA program based on the number of documents retrieved by the proximity operators of #uw10 and #uw20. (For InQuery's operators see Section 2.3.)

2.2 Indexing and query keys

We used the following approaches and techniques in indexing and for query keys:

- Genomics Track's test collection for the ad hoc task was a subset of the Medline collection. Each record consists of several fields. We indexed the fields TI (title), AB (abstract), MH (MeSH headings), RN (Registry Number), and GS (Gene Symbol)
- Letters were normalized to lower case.
- Query keys and the words of documents were normalized using the morphological analyzer *Kstem*, which is part of InQuery.
- Only letters (a-z) and numbers (0-9) were indexed. Hyphens and other characters in strings than letters and numbers were replaced by a space, and were not searchable.

- Strings containing both letters and numbers were decomposed into separate alphabetical and numerical strings. For example, the string *Gis4* was converted into *gis 4*. In searching the decomposed strings were combined by means of a proximity operator. For the strings *gis 4* the proximity statement was as *#od3(gis 4)*.

2.3 Retrieval system and query operators

The test system was the *InQuery* retrieval system (Allan et al., 2000; Larkey et al., 2005). InQuery is a probabilistic retrieval system based on the Bayesian inference net model. It provides a variety of query operators, including the Boolean conjunction operator *#band* which formed the basis of our official runs. For the #band-operator all its argument keys must occur in a document in order for the operator to contribute to the weight computed for that document. Otherwise #band contributes to the document score like the *#and*-operator. The weight of the #and-operator is computed as the product of the weights of its arguments.

Most *utaauto* queries contained one (sometimes more) dominating key. We call the dominating keys *primary keys*. The primary keys were determined on a basis of keys' document frequencies (Section 2.4). The primary key was combined with the #band-operator to the other keys of a query. If there was no primary key in a query the #and-operator was used instead of the #band-operator.

In the *utamanu* queries the keys were grouped into categories (subqueries) based on the aspects they represent in the topics. Different subqueries were combined to each other by the #band-operator.

Other InQuery operators used in the queries of this study were #sum, #field, #odn, and #uw. For the *#sum-operator*, the system computes an average of key (or subquery) weights. The keys of the *#field*-operator are searched within the specified field. Phrases were searched for using the proximity operators of *#odn* (ordered window operator) and *#uwn* (unordered window operator). Both are Boolean conjunction operators, only retrieving documents where all the arguments of the operator appear. In the queries of this study n=3 and n=6 were used as window sizes.

2.4 Query formulation

Next we describe the utaauto and utamanu query formulation. The queries used in the official and additional experiments are presented in Section 2.5.

Utaauto

For *utaauto*, the main features of query formulation were as follows:

- Formulation of queries on a basis of the Title and Need fields of the topics
- Removal of performative and marginally topical words from the topics using average term frequency statistics
- Removal of stop-words using InQuery's stop-word list
- Normalization of topic words using InQuery's Kstem morphological analyzer
- Identification of phrases in the topics using the RKA collocation technique
- Structural weighting of the primary keys
- Query expansion and reformulation

More precisely, *utaauto* query construction proceeded as follows.

Performative and marginally topical words often are frequent in the whole collection but typically occur only once or twice in the same document. We utilized this fact and removed from the topics words with average term frequency below 1.25 in the Genomics Track's collection. For key *k* its average term frequency *atf(k)* = cf_k / df_k , where cf_k is the collection frequency of the key *k*, and df_k its document frequency.

For example, for the performative word *find* atf is 1.20 in the collection. For the topic words atf typically is much higher, e.g., for *xenograft* atf is 1.98.

Phrases were identified in the topics by computing for each bigram a RKA value. If the RKA of a bigram was 1.15 or less it was handled as a phrase by applying a proximity operator for it. Here *bigram phrase* (or *phrase* in short) refers to a bigram with RKA < 1.15. Also bigrams consisting of a letter and number parts (e.g., gis 4) were handled as phrases, irrespective of their RKA values.

In many queries there are 1-2 dominating keys (primary keys). The primary key is characterized by low document frequency (Pirkola and Järvelin, 2001). Typically, if the primary key is removed from the query the performance of the query is deteriorated. In this study the primary key was defined as a key whose document frequency is 10 000 or less. *MutY*, *ubiquitin*, and *transgenic mice* are examples of primary keys. The primary keys were weighted structurally by combining them with the #band-operator to the other keys of a query. If there was no primary key in a query, the #band-operator was not used but the keys were combined with the #and-operator.

Initial queries were formulated and were run in the retrieval system. Our system does not support pseudorelevance feedback, but we simulated it as follows. For each query the MH terms of the top 20 documents were combined and sorted, and the number of the occurrences of each unique MH term was computed. Other MH terms than those that occurred at least seven times in the list or occurred at least four times and were marked as major descriptors (descriptors assigned asterisk) were removed from the list. The remaining keys were kept for expansion.

Utamanu

For *utamanu*, the main features of query formulation were as follows:

- Concept analysis
- Formulation of Boolean queries
- New terms from the MeSH, genome databases, and dictionaries
- Query expansion with MH terms

More precisely, *utamanu* query construction proceeded as follows.

The *utamanu* queries were Boolean queries constructed on a basis of the Title, Need, and Context fields of the topics and on a basis of a concept analysis made by a human. For each topic the main aspects involved in the topic were identified and were represented by the topic keys. New terms - synonyms, hyponyms, hypernyms, and related terms as well as inflectional and derivational variants for the topic words - were selected from the MeSH, genome databases, and dictionaries. From the genome databases, e.g., LocusLink and OMIM, we searched for synonymous gene names for the gene names contained in the topics. The initial queries were run in the system. New MH terms were identified in the top 20 documents of the initial search. Relevant new terms were found only for some queries.

2.5 Queries

2.5.1 Official runs

Utaauto

The structure of *utaauto* queries is schematically as follows for queries for which one primary key was found (the most usual case):

((the primary key) ++ (the other original keys contained in the topics)) +
((the primary key) ++ (MH expansion terms)) +
(all original keys contained in the topics)

++ denotes the #band-operator while + denotes the #and-operator. To avoid too restrictive queries all the original keys were included also in a separate subquery (the last subquery).

An example of an utaauto query is presented below (query # 01). For the query, *ferroportin 1* is a primary key.

#and(#band(#od3(ferroportin 1) #sum(human iron transport)) #band(#od3(ferroportin 1) #field(MH #sum(hemochromatosis #od6(cation transport proteins)))) #sum(human iron transport ferroportin 1))

Utamanu

The query below provides an example of an *utamanu* query (query # 01):

#band(#sum(#od3(ferroportin 1) #od6(slc 40 a 1) #od3(fpn 1) #od3(hfe 4) #od3(ireg 1) #uw6(iron regulated gene 1) #uw6(iron regulated transporter 1) #od3(mtp 1) #od6(slc 11 a 3) #uw6(solute carrier family 11)) #field(MH human) #field(MH #sum(#od6(cation transport proteins) #od6(carrier proteins genetics) #od6(cation transport proteins genetics) #od6(hemochromatosis genetics) #od6(iron binding proteins) #od6(iron metabolism) hemochromatosis iron)))

If the initial query gave no results the #and-operator was used instead of the #band-operator. The MH terms represented either aspects involved in the topics or narrower, broader or related aspects.

2.5.2 Additional experiments

In the additional experimenst we analyzed the following features of utaauto:

1. Query expansion with MH terms
2. The use of phrases in queries
3. Primary key weighting

In the first experiment we removed the MH terms from the utaauto queries to study their contribution to utaauto's performance. The reduced queries are named *utaauto/no-expansion*.

The effects of the use of bigram phrases in queries were studied by using as a baseline *single key queries* that only contained the single keys from the utaauto/no-expansion queries and *phrase-based queries* that contained the same keys as single key queries but differed from them in that bigram phrases were wrapped in a proximity operator. The identification of bigram phrases is described in Section 2.4. Examples of single key and phrase-based queries are presented below.

Single key query (# 02)

#and(protocol generate transgenic mice)

Phrase-based query (# 02)

#and(protocol generate #uw3(transgenic mice))

For utaauto the effects of primary key weighting were tested by treating the primary keys similarly to the other keys, i.e., in this experiment the primary keys were not weighted more than the other keys. We call these queries *utaauto/no-primary*.

Most of the primary keys were phrases, and the weighting of the phrasal primary keys was tested by using a query type which is named *pkw*. In the pkw queries the most specific component of a phrasal primary key (i.e., the component with the lowest df) was used in the query in addition to the full phrase. The performance of the pkw queries is compared to that of phrase-based queries (for the pkw queries the operators #sum and #and gave approximately the same results). An example of a pkw query is presented below.

Pkw query (# 01)

#sum(#od3(ferroportin 1) ferroportin human iron transport)

3. Results and conclusions

The results of the official and the additional experiments are presented in Table 2. As can be seen, for the *utaauto* queries MAP is 0.3324 and for the *utamanu* queries 0.3128. *Utamanu* is characterized by a comprehensive terminology and the Boolean query structure. The *utaauto* queries have many features that may have contributed to the relatively good performance. In the additional experiments the focus was on analyzing the effects of utaauto's main features as explained in Section 2.5.2.

It is seen in Table 2 that performance is degraded only slightly due to the removal of MH terms. Thus, utaauto queries slightly benefited from query expansion with MH terms.

The utaauto/no-primary queries differed from the utaauto queries in that the primary keys were not weighted but they were treated similarly to the other keys. For the utaauto/no-primary queries the MAP is 0.2575. The low MAP and the good performance of pkw queries (MAP 0.3455) corroborate our earlier findings showing that retrieval performance is improved by assigning more weight to the dominating key(s) than the other keys of the query. In the earlier research the analysis of TREC topics showed that in most topics there are 1-2 dominating keys (primary keys) which are characterized by low document frequency (Pirkola and Järvelin, 2001). In this study a key with document frequency < 10 000 was defined as a primary key.

As shown in Table 2 the phrase-based queries outperform the single key queries. This is expected and reasonable since many of the phrase components are general words (or numbers), being thus bad discriminators. However, we did not test the statistical significance of the findings. Also tuning the threshold RKA for selecting bigram phrases for queries, and the issue of good and bad bigram phrases are tasks for future research.

Table 2. Retrieval performance of the test queries

Query type	MAP
Utaauto	0.3324
Utamanu	0.3128
Utaauto/no-expansion	0.3306
Utaauto/no-primary	0.2575
Single key queries	0.3050
Phrase-based queries	0.3271
Pkw queries	0.3455

Our findings on the blind use of MH terms are consistent with the results reported by Darwish and Madkour (2004). The researchers found that blind relevance feedback did not give statistically significant performance improvements. There are several factors that we need to take into account when examining the blind use of MH terms, and it needs to be investigated more thoroughly.

For single key queries MAP is 0.3050. Thus relatively good performance is achieved without using any method to improve queries. However, as our results show the weighting of the primary keys using the Boolean conjunction and the weighting of the specific components of phrasal primary keys are techniques to improve retrieval performance. The results are consistent with our earlier findings.

Acknowledgements

This research is part of the research project *NLP-based information retrieval systems for the biological literature* funded by the Academy of Finland (funding decision 206568).

The InQuery search engine was provided by the Center for Intelligent Information Retrieval at the University of Massachusetts.

References

Allan, J., Connell, M.E., Croft, W.B., Feng, F.-F, Fisher, D. and Li, X. 2000. Inquery and TREC-9. The Ninth Text REtrieval Conference (TREC-9), Gaithesburg, MD. Available at: http://trec.nist.gov/pubs/trec9/t9_proceedings.html

Darwish, K. and Madkour, A. 2004. The GUC goes to TREC 2004: using whole or partial documents for retrieval and classification in the Genomics Track. The Thirteenth Text REtrieval Conference (TREC 2004). Notebook paper.

Larkey, L.S. and Connell, M.E. 2005. Structured queries, language modeling, and relevance modeling in cross-language information retrieval. Information Processing & Management, 41(3), 457-473.

Pirkola, A. and Järvelin, K. 2001. Employing the resolution power of search keys. Journal of the American Society for Information Science and Technology, 52(7), 575-583.