Evaluation of Stemming, Query Expansion and Manual Indexing Approaches for the Genomic Task

Samir Abdou
Institut d'informatique
University of Neuchatel
Pierre-à-Mazel 7
2000 Neuchatel, Switzerland
Samir.Abdou@unine.ch

Patrick Ruck
University Hospital of Geneva
1211 Geneva 4, Switzerland
Patrick.Ruck @sim.hcuge.ch

Jacques Savoy
Institut d'informatique
University of Neuchatel
Pierre-à-Mazel 7
2000 Neuchatel, Switzerland
Jacques.Savoy@unine.ch

ABSTRACT

This paper describes our participation in TREC-2005 for the ad hoc Genomic track, in which we evaluate five different stemming approaches to performing domain-specific searches within a MEDLINE subset. We also evaluate the impact that manually assigned descriptors (MeSH headings) have on retrieval effectiveness. We design a domain-specific query expansion scheme and compare it with the more classic Rocchio approach. In our experiments on this collection subset, we discover that mean average precision does not improve when using different stemming algorithm. We then show how the presence of the MeSH headings significantly enhances mean average precision by about 9%. Finally, we illustrate how the use of various query expansion techniques can impairs retrieval performance.

1. INTRODUCTION

During our participation in the ad hoc Genomic track, we were concerned with domain-specific IR systems that would provide a ranked list of MEDLINE records in response to requests by biologists. This involved a set of available queries describing typical search interests, and in which gene and protein names were essential elements in effective retrieval. Given that in biomedical publications information is evolving rapidly and involves a wide variety of terminology. It is known that large numbers of names, symbols and synonyms are used to denote the same protein or gene. In order to provide a partial solution to some of these problems, this paper describes three strategies that have been suggested to hopefully enhance the effectiveness of biomedical information searches.

First we evaluate the impact of various stemming procedures used to conflate word variants into an appropriate lemma or stem. This is based on the assumption in IR systems that efficient approaches make use of stemming schemes in order to resolve partially the vocabulary mismatches that occur when users submit terms that are very different from those used by document authors. In other words, when a query contains the term "computer," it would seem reasonable to also retrieve documents containing related words such as "computers," or even "computing."

Second, we accept the fact that manually assigned descriptors would increase the chances of retrieving more pertinent documents, especially compared to searches based only on only terms provided in the query. Usually based on controlled vocabularies, manual indexing tends to result in greater indexing consistency. In fact, the underlying thesaurus used would prescribe a uniform and invariable choice of indexing descriptors, normalize orthographic (e.g., "database" or "data base") and lexical variants (e.g., "analyzing," "analysis") or any expressions with similar meanings (e.g., "computer science," "informatics").

Third, we should also assume that when searching information, users will not know all synonyms and related terms needed to accurately express their information needs. Query expansion would thus take different term-term relationships into account and determine which words or phrases should be included in an expanded query. Based on various empirical studies already carried out, such automatic query expansion approaches usually result in better retrieval performance.

The rest of this paper is organized as follows. Section 2 depicts the main characteristics of our test-collection. Section 3 briefly describes the IR models used during our experiments. Section 4 presents our domain-specific and general query expansion approaches. Section 5 evaluates five different stemming schemes and two query expansion methods. The main findings of this paper are presented in Section 6.

2. TEST-COLLECTION

The corpus used in our experiments was extracted from the well-known MEDLINE[1] bibliographic database on biomedical literature. This corpus subset was made available for the TREC-2005 evaluation campaign and includes around 10 years of scientific publications (4,591,008 records or about 10.6 GB of compressed data).

PMID- 10605436
…
DP - 1978 Feb
TI - Concerning the localization of steroids in centrioles and basal bodies by immunofluorescence.
PG - 255-60
AB - Specific steroid antibodies, by the immunofluorescence technique, regularly reveal fluorescent centrioles and cilia-bearing basal bodies intarget and nontarget cells. Although the precise identity of the immunoreactive steroid substance has not yet been …
AU - Nenci I
AU - Marchetti E
PT - Journal Article
RN - 0 (Steroids)
SB - IM
MH - Animals
MH - Centrioles/*ultrastructure
MH - Cilia/ultrastructure
MH - Female
MH - Fluorescent Antibody Technique
MH - Human
MH - Lymphocytes/*cytology
MH - Male
MH - Organelles/*ultrastructure
MH - Rats
MH - Rats, Sprague-Dawley
MH - Respiratory Mucosa/cytology
MH - Steroids/*analysis
MH - Trachea
SO - J Cell Biol 1978 Feb;76(2):255-60.
…

Table 1. Example of a MEDLINE record

Each record is structured according to a specific set of fields[2], such as PMID (PubMed unique identifier), DP (publication date), AU (author), PT (publication type), SO (source), etc. From an IR perspective, the most important sources of information are the article title (TI), the abstract (AB together with the OAB field, other abstracts supplied by an NLM collaborating organization) and the set of manually assigned MeSH headings (MH) extracted from the MeSH[3] Thesaurus. Along with these three main fields, we also might have assumed that the RN field[4], OT (other non-MeSH keywords) would be of some value in an IR application.

Within this collection are fifty topics (see examples listed in Table 2) that correspond to the real information needs expressed by biologists. This topic set is subdivided into five different main scenarios (or typical search interests). Regardless of the topic, the IR system is to return the same type of answer, namely a ranked list of MEDLINE records.

Five information need scenarios are described below. Topics #100 to #109 correspond to search examples for certain standard methods or protocols, for some given type of experiment (e.g., "How to "open up" a cell"). Topics #110 to #119 represent information describing the role(s) of a gene involved in a disease (e.g., "Interferon-beta and multiple sclerosis"). Topics #120 to #129 represent information needs related to the role of a gene in a specific biological process (e.g., "casein kinase II in ribosome assembly"). Topics #130 to #139 represent information needs describing interactions between two or more genes in the function of an organ or in a disease (e.g., "Bop and Pes in cell growth"). Finally, topics #140 to #149 represent information explaining one or more mutations of a given gene and its biological impact or role (e.g., "mutations in metazoan Pes and effect on cell growth").

<ID> 105
<METHOD> Purification of rat IgM

<ID> 111
<GENE> PRNP
<DISEASE> Mad Cow Disease

<ID> 120
<GENE> Nucleoside diphosphate kinase (NM23)
<PROCESS> Tumor progression

<ID> 130
<GENE> BRCA1 regulation of ubiquitin
<DISEASE> cancer

<ID> 140
<GENE> BRCA1 185delAG mutation
<PROCESS> role in ovarian cancer

Table 2. One example taken from each of five topic scenarios

As shown in Table 2, the vocabulary used in these topics tends to indicate that general dictionaries or thesauri do would not be the most appropriate tools for deriving additional useful search terms. Thus more specific tools such as gene ontologies or databases would provide the

[1] See http://www.nlm.nih.gov/pubs/factsheets/medline.html

[2] For more information about all these fields, see the site http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=helppubmed.table.pubmedhelp.T44

[3] See the site http://www.nlm.nih.gov/mesh/meshhome.html

[4] Number assigned by the Enzyme Commission to designate a particular enzyme or by the Chemical Abstracts Service for Registry Numbers.

additional information needed to enhance the information needs submitted by users.

Based on relevance assessments made of this test-collection, for each query we found that the number of relevant records averaged 93.551 (median: 35; standard deviation: 139.6). Query #144 returned only two pertinent documents while Query #117 produced the greatest number of relevant articles (namely 709). Query #135 did not reveal any pertinent articles.

3. IR MODELS

In order to obtain a broader view of the relative merits of the various retrieval models, we analyzed nine different vector-space schemes and two probabilistic models. First we adopted a binary indexing scheme in which each document (or request) was represented by a set of keywords, without any weights. To measure similarities between documents and requests we computed the inner product (retrieval model denoted "doc=bnn, query=bnn" or "bnn-bnn"). Then, to weight the presence of each indexing term in a document surrogate (or in a query), we took term occurrence frequencies into account (denoted tf_{ij} for indexing term t_j in document D_i, denoting the corresponding retrieval model as "doc=nnn, query=nnn"). We could also account for their inverse document frequency (denoted idf_j) and then also normalize each indexing weight by applying different weighting schemes, as described in the Appendix.

Other variants could also be created, especially when the occurrence of a particular term in a document is viewed as a rare event. Thus, it may be preferable to assign more importance to the first occurrence of this word, compared to any successive and repetitive occurrences. Therefore, the *tf* component may be computed as the ln(*tf*) + 1.0 (retrieval model notation: "doc=ltc, query=ltc") or as 0.5 + 0.5 · [*tf* / max *tf* in a document]. Different weighting formulae could of course be used for documents and requests, leading to other different weighting combinations. We might also consider that a term's presence in a shorter document provides stronger evidence than in a longer one, thus leading to more complex IR models; for example the IR model denoted by "doc=Lnu" [1], "doc=dtu" [2].

In addition to these vector-space schemes, we also considered probabilistic models, such as the Okapi model [3]. As a second probabilistic approach, we implemented the I(*n*)L2 approach, within the Deviation from Randomness (DFR) framework [4]. This IR model is based on combining two information measures, formulated as follows:

$$w_{ij} = Inf^1_{ij} \cdot Inf^2_{ij} = -\log_2[Prob^1_{ij}] \cdot (1 - Prob^2_{ij}) \quad (1)$$

in which $Prob^1_{ij}$ is the probability of having by pure chance tf_{ij} occurrences of the term t_j in a document. On the other hand, $Prob^2_{ij}$ is the probability of encountering a new occurrence of term t_j in the document, given that we already found tf_{ij} occurrences of this term.

Within this DFR framework, the model I(*n*)L2 is based on the two following formulae:

$$Inf^1_{ij} = tfn_{ij} \cdot \log_2[(n+1) / (df_j+0.5)] \quad (2)$$

$$Prob^2_{ij} = tfn_{ij} / (tfn_{ij} + 1) \quad (3)$$

$$\text{with } tfn_{ij} = tf_{ij} \cdot \log_2[1 + ((c \cdot mean\ dl) / l_i)]$$

where df_j indicates the number of documents indexed with the term t_j, n the number of documents in the corpus, tc_j the number of occurrences of term t_j in the collection, l_i the length (number of indexing terms) of document D_i, *mean dl* the average document length, and c a constant. In our experiments, the constant c = 1.5 and *mean dl* = 146.

4. QUERY EXPANSION METHODS

In this study, we were interested in knowing whether a domain-specific query expansion could provide a better retrieval performance than a general scheme. It is our opinion that in the biomedical domain vocabulary mismatches between requests and documents generated by the terminology variation are a real problem. We thus hope to reduce these mismatches by applying appropriate domain-specific query expansion approaches, as described in Section 4.1. On the other hand, a general query expansion scheme as described in Section 4.2 could also complement our domain-specific approach.

4.1 Domain-Specific Query Expansion

In the biomedical domain, gene or protein name orthographic variants are numerous. Usually, a few rules [5] can be used to generate or to allow a match between some of the name variations. These rules are the following:

- presence of a space or a hyphen ("IL 10," and "IL-10");
- space or hyphen removed ("ddvit1," and "ddvit 1");
- the word *alpha* or *beta* might be replaced by a single letter ("epm2-beta" or "epm2b");
- the final digit '-1,' '-2,' '-3,' or '-4' might be replaced with their Roman equivalent ("UEV-2," and "UEV-II");
- parts of the name might be written in uppercase or lowercase letters ("DDVit-1," and "ddvit1").

In other cases, there might not be any clear relation between the various synonyms used for gene or protein names, caused in part by the various sub-domains present in biological literature. For example, the protein "*lymphocyte associated receptor of death*" could be

denoted as "LARD," "Apo3," "DR3," "TRAMP," "wsl," and "TnfRSF12" [6]. Various databases and repositories[5] have thus been created, partly to assist searchers in finding gene or protein synonyms (and to provide additional more useful information such as gene functions, biological processes, protein sequences or structures). Constructing and updating these information sources would require a great deal of manual input, but they might prove to be useful in our domain-specific query expansions. We know however that these specific and valuable knowledge sources would not contain all the synonyms of any given protein or gene name. A complete description of our domain-specific queries can be found in [7].

Some of the following examples will be useful in understanding our purposes. Topic #113 contains the gene name "MMS2" and the system automatically adds the following orthographic variants: "MMS II," "mms 2," and "mms2." Additionally, the domain-specific query expansion process found the related term "UBE2V2," and this term along with its list of synonyms ("UBE2V 2," "ube2v 2," "UBE2V II," "ube2v ii," "ube2v2") were thus added to the final expanded query.

<ID> 111
<GENE> PRNP
<DISEASE> Mad Cow Disease
<BEST> PRNP
<EXP> PrP33-35C
MGC26679
PrP
PRIP
ASCR
PrP27-30
CJD
PrPc
prion protein
PRP
prion protein relate
MGC 26679
prp33-35c
mgc26679
prp27-30
mgc 26679

Table 3. Example of our domain-specific query expansion

Not all topic descriptions were modified. The following 14 topics were not expanded by our domain-specific query expansion (namely Query #100, #101, #102, #103, #104, #106, #107, #108, #109, #110, #129, #131, #137 and #147). For the other 36 queries, on average we added 27 synonyms/definitions (minimum: 1; maximum: 96; median: 25; and standard deviation: 18.7).

Expanded Query #111 is shown in Table 3, which lists all added synonyms (under the label <EXP>), including closely related spellings ("PrP" and "PRP," or "PrP33-35C" and "prp3-335c"), as well as certain definitions or synonyms ("prion protein").

These added terms do not occur in the collection with the same document frequency (in fact, we have no guarantee that these added terms even appear in the corpus). For example, the term "prion" (shown in Table 3) appears in 4,112 MEDLINE records, while the term "prp" in 2,711 ("prpc" in 679, "ascr" in 57, "prip " in 22 and "prp27-30" in 19). The terms "prp33-5c" and "mgc26679" however do not occur at all in the corpus. For the additional terms, document frequency seems to be lower than in the original query (with some exceptions, such as "protein," occurring in 858,669 records).

An inspection of the entire original query set shows that there were 219 search terms (or 4.38 terms per query) occurring in at least one document[6]. Based upon the document frequency of these search terms, the average document frequency was 160,628.5 (minimum: 1; maximum: 1,814,074 (term "t"); median: 32,194; and standard deviation: 308,456).

In our domain-specific query expansion, we added 468 terms, and from this set there were 81 terms that did not appear in the corpus at all. The difference in retrieval performance was due to the 387 remaining terms only. When a query was expanded (for 36 queries over a total of 50), on average the system added 13.0 new search terms per query (or 10.74 when ignoring terms not occurring in the corpus). Upon computing the document frequency of these 387 new terms, we found the average document frequency to be 65,912.1 (minimum: 1; maximum: 3,283,925 (term "human"); median: 1,092; and standard deviation: 221,453). Clearly theses values tend to confirm our first impression: the frequencies of words added by our domain-specific query expansion approach were, on average, lower.

4.2 Blind-Query Expansion

Various general query expansion approaches have been suggested, and in this paper we will compare our domain-specific query expansion with Rocchio's scheme [1]. In this latter case, the system was allowed to add *m* terms extracted from the *k* best ranked documents from the original query. New queries were derived by applying the following formula:

[5] See the SwissProt Web site http://us.expasy.org/sprot/, the GenBank site at http://www.ncbi.nlm.nih.gov/, or the Gene Ontology at http://www.geneontology.org/

[6] The terms "Nurr-77" (in Query #115), "aapolipoprotein" (in Query #117), "HFN4" (in Query #138), and "4-GABAA" (in Query #149) do not appear in any record of our corpus.

$$Q' = \alpha \cdot Q + (\beta/k) \cdot \sum_{j=1}^{k} w_{ij} \quad (4)$$

in which Q' denotes the new query built for the previous query Q, and w_{ij} denotes the indexing term weight attached to the term t_j in the document D_i. In our evaluation, we fixed $\alpha = 2.0$, $\beta = 0.5$.

5. EVALUATION

To evaluate our various IR schemes, we adopted non-interpolated mean average precision (MAP) to measure retrieval performance. This was computed by the TREC_EVAL program, based on the retrieval of 1,000 items per request. To statistically determine whether or not a given search strategy would be better than another, we applied the bootstrap methodology [8].

In our statistical testing, the null hypothesis H_0 states that both retrieval schemes result in similar mean performances. Thus this null hypothesis plays the role of a devil's advocate, meaning this assumption would be accepted if two retrieval schemes returned statistically similar means, otherwise it would be rejected. In the tables included in this paper, we have thus underlined any statistically significant differences resulting from a two-sided non-parametric bootstrap test, based on the MAP difference (significance level 5%).

5.1 IR Models & Stemming Evaluation

Based on this evaluation methodology, Table 4 depicts the MAP for the MEDLINE collection subset, using different IR models. In this table, the best performance under a given condition (depicted in bold) will be used as the baseline for statistical testing. The first column lists the IR models tested, while the second to sixth columns contain evaluations of the different stemming approaches.

As a first stemming procedure, we may assume that stemming must be discarded and the indexing of requests and documents would therefore ignore this word normalization procedure (performance shown under the label "None" in Table 4).

For the English language, we may use either the Porter stemmer [9], having about 60 rules, or the Lovins stemmer [10] based on about 260 rules. The SMART system proposed a third approach based in part on the Lovins scheme, but producing different stems. These three approaches are relatively aggressive, removing both inflectional and derivational suffixes. It seemed reasonable to assume that inflectional endings, used to indicate genre (masculine vs. feminine) or number (singular vs. plural), would not really modify the meaning of a given word. For example the words "algorithms" and "algorithm" are closely related and thus if one of them appears in a query and the other in a document, we would assume that both word variants refer to the same meaning and that the corresponding document should be retrieved.

```
if final is '-ies' but not '-eies' or '-aies' then
        replace '-ies' by '-y', return;
if final is '-es' but not '-aes', '-ees' or '-oes' then
        replace '-es' by '-e', return;
if final is '-s' but not '-us' or '-ss' then
        remove '-s';
return.
```

Table 5. Minimal S-stemmer [11]

Of course in English, as in other natural languages, there are exceptions to this rule (e.g., words appearing only in plural form, such as "scissors"). Matches obtained from words derived by adding suffixes (e.g. '-ment', '-ably', '-ship') would however be more questionable. For example, "algorithm" and "algorithmic" do not have the same meanings. Thus it might be in our interest to propose a simple and light stemming approach, one that simply removes the most important inflectional suffixes, such as the plural '-s' form for the English language. Table 5 lists an approach suggested by Harman [11] and Table 4 lists its corresponding retrieval performance under the "S-stemmer" heading.

The various stemming approaches are evaluated in Table 4, showing that the I(*n*)L2 probabilistic model provided the best retrieval performance. The underlined values in Table 4 show that the MAP differences between the various IR models are always statistically significant.

In order to verify whether or not a stemming procedure might statistically improve mean average precision, the second column without stemming (label "None") was used as the baseline. To limit the number of comparisons, we no longer considered the "bnn-bnn" and "nnn-nnn" IR models, since as shown in Table 4, both of which resulted in poor retrieval effectiveness.

Upon comparing these nine best performing IR models, we found that of the four models (Okapi, "ltn-ntc," "lnc-ltc," and "ltc-ltc"), the S-stemmer proved to be the best approach, while the SMART stemmer performed best for only one model (namely I(*n*)L2 which was also the best performing approach listed in Table 4). For these three IR models ("dtu-dtn," "atn-ntc," and "ntc-ntc"), ignoring the stemming procedure proved to be the best solution.

After averaging MAP values across these nine best performing models, we found an average of 0.1939 for the "None" approach, 0.1933 when using the S-stemmer, 0.1899 with the Porter, 0.1894 with the SMART, and 0.1789 for the Lovins scheme. From these average values or when considering the MAP of the two best performing models (namely Okapi and I(*n*)L2 in Table 4), there is an

\ Stemmer IR Model	Mean average precision (% change) None	Porter	Lovins	SMART	S-stemmer
doc=Okapi, query=npn	0.2564 (-2.6%)	0.2551 (-2.8%)	0.2454 (-2.6%)	0.2562 (-2.9%)	0.2572 (-2.5%)
I(*n*)L2, query=nnn	**0.2633**	**0.2624**	**0.2519**	**0.2639**	**0.2637**
doc=Lnu, query=ltc	0.2232 (-15.2%)	0.2235 (-14.8%)	0.2081 (-17.4%)	0.2213 (-16.1%)	0.2211 (-16.2%)
doc=dtu, query=dtn	0.2365 (-10.2%)	0.2292 (-12.7%)	0.2127 (-15.6%)	0.2290 (-13.2%)	0.2328 (-11.7%)
doc=atn, query=ntc	0.2058 (-21.8%)	0.2019 (-23.1%)	0.1853 (-26.4%)	0.1979 (-25.0%)	0.2018 (-23.5%)
doc=ltn, query=ntc	0.1852 (-29.7%)	0.1834 (-30.1%)	0.1716 (-31.9%)	0.1807 (-31.5%)	0.1879 (-28.7%)
doc=lnc, query=ltc	0.1333 (-49.4%)	0.1357 (-48.3%)	0.1347 (-46.5%)	0.1335 (-49.4%)	0.1402 (-46.8%)
doc=ltc, query=ltc	0.1341 (-49.1%)	0.1229 (-53.2%)	0.1124 (-55.4%)	0.1248 (-52.7%)	0.1344 (-49.0%)
doc=ntc, query=ntc	0.1069 (-59.4%)	0.0948 (-63.9%)	0.0881 (-65.0%)	0.0975 (-63.1%)	0.1007 (-61.8%)
doc=bnn, query=bnn	0.1246 (-52.7%)	0.1370 (-47.8%)	0.1269 (-46.9%)	0.1375 (-47.9%)	0.1361 (-48.4%)
doc=nnn, query=nnn	0.0326 (-87.6%)	0.0283 (-89.2%)	0.0250 (-90.1%)	0.0279 (-89.4%)	0.0274 (-89.6%)

Table 4. MAP of various IR models, applying different stemming strategies

evident performance difference between an IR system with or without stemming is rather small. Moreover, the performance differences between the various stemming schemes are also small. Although the Lovins approach does however seem to perform as well as approaches without a stemming phase, these differences are not statistically significant (except for the "ntc-ntc" model).

Based on this test-collection, MAP differences between an approach without stemming and five different stemming schemes are usually not statistically significant. With this test-collection, stemming approaches do not improve mean average precision. For the "ntc-ntc" IR model however, the difference between an approach without stemming and the five stemming schemes is always statistically significant, and favors an approach without stemming.

5.2 High Precision Evaluation

It was assumed that a light stemming approach such as the S-stemmer would produce better results during high precision searches. To verify this hypothesis, we computed the precision achieved following the retrieval of ten documents, for each of the four stemming approaches (mean average precision is depicted in Table 6).

These MAP values listed in Table 6 show that the I(*n*)L2 model performed best, using the Porter or SMART stemmer. The second best performance was obtained from the Okapi model, using the Porter stemmer. For the last five IR models only, the best performance was achieved either with the S-stemmer or with an approach that ignored the stemming phase (under the label "None" in Table 6).

On the other hand, when the I(*n*)L2 model was used as a baseline, the performance differences with the Okapi or "Lnu-ltc" model were not statistically significant when considering the four different stemming approaches. The last five IR models however always had statistically lower performance levels when compared to the precision obtained by the I(*n*)L2 probabilistic model (values underlined in Table 6).

IR models	Mean average precision None	S-stemmer	Porter	SMART
Okapi-npn	0.4163	0.4245	0.4306	0.4224
I(*n*)L2	**0.4224**	**0.4286**	**0.4347**	**0.4347**
Lnu-ltc	0.4020	0.4061	0.4041	0.4020
dtu-dtn	0.3776	0.3837	0.3796	0.3857
atn-ntc	0.3571	0.3245	0.3490	0.3347
ltn-ntc	0.3347	0.3408	0.3245	0.3224
lnc-ltc	0.2122	0.2204	0.2122	0.2041
ltc-ltc	0.2020	0.2163	0.1898	0.1796
ntc-ntc	0.2265	0.2122	0.1918	0.1918

Table 6. Mean precision after 10 documents

IR models	Mean average precision with MeSH	TI & AB only
Okapi-npn	0.2551	0.2398 (-6.0%)
I(*n*)L2	**0.2624**	**0.2486** (-5.3%)
Lnu-ltc	0.2235	0.2139 (-4.3%)
dtu-dtn	0.2292	0.2139 (-6.7%)
atn-ntc	0.2019	0.2059 (+2.0%)
ltn-ntc	0.1834	0.1695 (-7.6%)
lnc-ltc	0.1357	0.0787 (-42.0%)
ltc-ltc	0.1229	0.1034 (-15.6%)
ntc-ntc	0.0948	0.0966 (+1.9%)

Table 7. MAP with and without MeSH headings (with Porter's stemmer)

5.3 Manually Assigned Headings

In Table 7, we listed the mean average precision achieved with various IR models, where only the article title and

abstract (under the label "TI & AB only") were used to build the document surrogate. Under this indexing restriction, the overall retrieval performance was lower than the corresponding system with manually assigned descriptors (2nd column in Table 7). When taking the nine best performing IR models into account, average percent decreases were about 9.3% when the search system did not include the MeSH headings. These performance differences were usually statistically significant, except for the "ntc-ntc" model.

5.4 Evaluating Query Expansion Models

As explained in Section 4, we designed a domain-specific query expansion scheme. In order to compared this search strategy with the more classic Rocchio scheme, we used the Okapi and I(*n*)L2 probabilistic models and applied different parameter settings.

	Mean average precision	
Models & parameters	Domain specific	Rocchio
Okapi-npn	**0.2551**	**0.2551**
3 docs / 10 terms	0.2114	0.2454
3 docs / 20 terms	0.2114	0.2492
5 docs / 10 terms	0.2114	0.2416
5 docs / 20 terms	0.2114	0.2478
10 docs / 10 terms	0.2114	0.2386
10 docs / 20 terms	0.2114	0.2436
I(*n*)L2-nnn	**0.2624**	**0.2624**
3 docs / 10 terms	0.2128	0.2417
3 docs / 20 terms	0.2128	0.2540
5 docs / 10 terms	0.2128	0.2409
5 docs / 20 terms	0.2128	0.2554
10 docs / 10 terms	0.2128	0.2324
10 docs / 20 terms	0.2128	0.2439

Table 8. Mean average precision achieved by two query expansion models

Table 8 lists the mean average precision values obtained with these search models. The rows labeled "Okapi-npn" or "I(*n*)L2-nnn" form the baseline (with Porter's stemming) and indicate the MAP before query expansion schemes were applied. Rows starting with "*k* doc / *m* terms" indicate the number of top-ranked documents and the number of terms used to enlarge the original query. The remaining rows indicate the corresponding MAP values achieved by the three query expansion approaches. This parameter setting was of course only applied to the Rocchio scheme. As explained in Section 4.1, the domain-specific query expansion will add, in mean, 10 new terms to each query.

We were surprised to learn that both query expansion approaches resulted in lower MAP values. When compared with the baseline (performance achieved without query expansion) and when using the Rocchio's scheme, differences were however usually not statistically significant. For the domain-specific query expansion, only 36 queries were expanded. If we only consider this query subset, mean average precision for the I(*n*)L2 model is 0.2906 without query expansion, and with our domain-specific query expansion a MAP of 0.2211, a relative decrease of -23.9%. A query-by-query analysis revealed that our domain-specific query expansion improved the retrieval performance for 13 queries, but decreased performance for other 36 queries.

5.5 Official Runs

Table 9 lists the MAP for our two official runs, together with their various components. The run labeled "UniNeHug2" was based on the probabilistic model I(*n*)L2 using the Rocchio query expansion technique (10 documents / 20 terms). The second official run ("UniNeHug2c") was based on data fusion of the two result lists using the Z-score fusion model [12]. Within this scheme, we normalized the retrieval status values (or document scores) for each document D_k provided by the *i*th result list, as computed by the following formula:

$$\text{Z-score } RSV_k = \alpha_i \cdot [((RSV_k - Mean^i) / Stdev^i) + \delta^i],$$
$$\delta^i = ((Mean^i - Min^i) / Stdev^i) \quad (6)$$

within which $Mean^i$ denotes the average of the RSV_k, $Stdev^i$ the standard deviation, and α_i reflects the retrieval performance of the underlying retrieval model.

IR Models	MAP
1. I(*n*)L2-nnn	**0.2624**
2. Rocchio (10 doc/20 term) **UniNeHug2**	0.2439
3. Domain-specific QE	0.2128
4. Domain-specific QE + 10 doc/ 20 terms	0.2150
5. Data fusion (2 & 4), **UniNeHug2c**	0.2375
6. Data fusion (2 & 4), Round-robin	0.2395
7. Data fusion (2 & 4), $\sum RSV_k$	0.2322
8. Data fusion (2 & 4), RSV_k/ Max	0.2424

Table 9. MAP of our official runs

In this data combination, the first result list is simply the "UniNeHug2" run. The second was provided by the IR model I(*n*)L2, involving both our domain-specific and Rocchio (10 documents / 20 terms) query expansion approaches. The 4th row in Table 9 lists our evaluation of this IR scheme, showing a performance level of 0.2150. When fusing this result list with UniNeHug2, we fixed the coefficients $\alpha_i = 1$ for the UniNeHug2 run and $\alpha_i = 1.5$ for the 4th run.

The MAP (0.2375) of the resulting combined run ("UniNeHug2c") was slightly inferior to that of the best single model (MAP: 0.2439). Rows 6 to 8 of Table 9 list the MAP achieved using other data fusion operators, where the resulting MAP is fairly close to performance levels achieved by the UniNeHug2c run.

6. CONCLUSION

During the TREC-2005 Genomic evaluation campaign, we evaluated five different stemming procedures. The empirical evidence collected shows that when stemming procedures are applied to the MEDLINE collection, retrieval effectiveness improvements are not statistically significant. Differences in performance between the various stemmers are also usually not statistically significant. Moreover, when analyzing high precision searches (measured by average precision after 10 documents), we discovered that a light stemming approach does not perform better than the other more aggressive stemmers.

The inclusion of the MeSH headings when indexing the scientific articles improves retrieval performance significantly by about 9%, on average. Compared to other similar test-collections however, this enhancement is rather limited.

During this evaluation campaign, we also proposed a domain-specific query expansion. Both our domain-specific and Rocchio query expansion techniques did not however result in higher retrieval performance (although with the Rocchio scheme the performance difference is not always statistically significant).

ACKNOWLEDGMENTS

This research was supported in part by the Swiss NSF under Grant #200020-103420.

7. REFERENCES

[1] Buckley, C., Singhal, A., Mitra, M., & Salton, G. New retrieval approaches using SMART. In *Proceedings of TREC-4*. Gaithersburg, MA, 1996, 25-48.

[2] Singhal, A., Choi, J., Hindle, D., Lewis, D.D. & Pereira, F. (1999). AT&T at TREC-7. In *Proceedings TREC-7*, Gaithersburg, MA, 1999, 239-251.

[3] Robertson, S.E., Walker, S., & Beaulieu, M. Experimentation as a way of life: Okapi at TREC. *Information Processing &Management*, 36(1), 2000, 95-108.

[4] Amati, G., & van Rijsbergen, C.J. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM-Transactions on Information Systems*, 20(4), 2002, 357-389.

[5] Cohen, A.M. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceeding ACL-ISMB*, Detroit (MI), 2005, 17-24.

[6] Yu, H., & Agichtein, E. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19(1), 2003, i340-i349.

[7] Ruck, P., Ehrler, F., Abdou, S., Savoy, J. Report on TREC-2005 experiment: Genomics track. In *Proceedings of TREC-2005*. Gaithersburg, MA, 2005.

[8] Savoy, J. Statistical inference in retrieval effectiveness evaluation. *Information Processing & Management*, 33(4), 1997, 495-512.

[9] Porter, M.F. An algorithm for suffix stripping. *Program*, 14(3), 1980, 130-137.

[10] Lovins, J.B. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1), 1968, 22-31.

[11] Harman, D. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1), 1991, 7-15.

[12] Savoy, J. Data fusion for effective European monolingual information retrieval. In C. Peters, P.D. Clough, J. Gonzalo, G.J.F. Jones, M. Kluck & B. Magnini (Eds.), *Multilingual Information Access for Text, Speech and Images*. Lecture Notes in Computer Science #3491. Springer-Verlag, Berlin, 2005, 233-244.

8. APPENDIX

To assign an indexing weight w_{ij} reflecting the importance of each single-term t_j in a document D_i, we might use the various approaches shown in Table 10, where *n* indicates the number of documents in the collection, *t* the number of indexing terms, df_j the number of documents in which the term t_j appears, document length (the number of indexing terms) for D_i is denoted by nt_i, and *avdl*, *b*, k_1, *pivot* and *slope* are constants. For the Okapi weighting scheme, *K* represents the ratio between the length of D_i measured by l_i (sum of tf_{ij}), and the collection mean is noted by *avdl*. In our experiments, we fixed b=0.55, k_1=1.2, *avdl*=146=*mean dl*, and c=1.5.

bnn	$w_{ij} = 1$	nnn	$w_{ij} = tf_{ij}$
ltn	$w_{ij} = (\ln(tf_{ij}) + 1) \cdot idf_j$	atn	$w_{ij} = idf_j \cdot [0.5 + 0.5 \cdot tf_{ij} / \max tf_i]$
dtn	$w_{ij} = [\ln(\ln(tf_{ij}) + 1) + 1] \cdot idf_j$	npn	$w_{ij} = tf_{ij} \cdot \ln[(n - df_j) / df_j]$
Okapi	$w_{ij} = \left((k_1 + 1) \cdot tf_{ij}\right) \Big/ \left(K + tf_{ij}\right)$	Lnu	$w_{ij} = \dfrac{\left(1 + \ln(tf_{ij}) \Big/ \ln(\text{mean } tf) + 1\right)}{(1 - \text{slope}) \cdot \text{pivot} + \text{slope} \cdot nt_i}$
lnc	$w_{ij} = \dfrac{\ln(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} \left(\ln(tf_{ik}) + 1\right)^2}}$	ntc	$w_{ij} = \dfrac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik} \cdot idf_k\right)^2}}$
ltc	$w_{ij} = \left(\ln(tf_{ij}) + 1\right) \cdot idf_j \Big/ \sqrt{\sum_{k=1}^{t} \left(\left(\ln(tf_{ik}) + 1\right) \cdot idf_k\right)^2}$		
dtu	$w_{ij} = \dfrac{\left(\ln\left(\ln(tf_{ij}) + 1\right) + 1\right) \cdot idf_j}{(1 - slope) \cdot pivot + slope \cdot nt_i}$		

Table 10. Weighting schemes