Ranking for Medical Annotation: Investigating Performance, Local Search and Homonymy Recognition

Alexander K. Seewald

Austrian Research Institute for Artificial Intelligence, Freyung 6/6,
A-1010 Vienna, Austria
alexsee@oefai.at

Abstract. In this paper we investigate several hypotheses concerning document relevance ranking for biological literature. More specifically, we focus on three topics: performance, risk of local searching, and homonymy recognition. Surprisingly, we find that a quite simple ranker based on the occurrence of a single word performs best. Adding this word as a new search term to each query yields results comparable to elaborate state-of-the-art approaches. The risk of our local searching approach is found to be negligible. In some cases retrieval from a large repository even yields worse results than local search on a smaller repository which only contains documents returned by the current query. The removal of automatically determined homonyms yields almost indistinguishable results to the original query, so it is not inconceivable that the problem of homonymy in biological literature has been overstated. Concluding, our investigation of three hypotheses has been useful to decide implementation issues within our research projects as well as opening interesting venues for further research.

1 Introduction

Genome research has spawned unprecedented volumes of data, but characterization of DNA and protein sequences has not kept pace with the rate of data acquisition. To anyone trying to know more about a given sequence, the worldwide collection of abstract and papers remains the ultimate information source. The goal of the BioMinT[1] project is to develop a generic text mining tool that (1) interprets diverse types of query, (2) retrieves relevant documents from the biological literature, (3) extracts the required information, and (4) outputs the result as a database slot filler or as a structured report. The BioMinT tool will thus operate in two modes. As a curator's assistant, it will be validated on SwissProt[2] and PRINTS[3]; as a researcher's assistant, its reports will submitted to

[1] Biological Textmining, EU FP5 QoL project no. QLRI-CT-2002-02770.

[2] SwissProt is a popular database on protein sequence, function and other features. See [2].

[3] The PRINTS resource is a compendium of protein fingerprints. A fingerprint is a group of conserved motifs which characterise protein families and may be used to infer the function of an unknown protein. See [1].

the scrutiny of biologists in academia and industry. The project is conducted by an interdisciplinary team from biology, computational linguistics, and data/text mining.

Within this paper, we focus exclusively on the first mode of the BioMinT tool as the curator's assistant. More specifically, our evaluation focusses on document relevance ranking for medical annotation of Homo sapiens within SwissProt. Document relevance ranking is an important task within our project, so we have implemented a set of rankers for this task.

Ranking of documents by relevance is a well-researched topic. For example, the Text Retrieval Conferences are organized by the U.S. National Institute of Standards & Technology[4] on a yearly basis and are organized in the form of competitions, yielding dozens of publications per year. Other papers include [3, 7, 10]. A comprehensive overview on the growing field of biological literature text mining can be found in [5].

In this paper we investigate the performance of diverse ranking systems within the BioMinT prototype. We consider both query-independent ranking systems, i.e. those which are trained on a corpus of relevant and irrelevant documents from the same topic and where the score[5] of any one document only depends on its content; and query-dependent ranking systems, i.e. those where the score is dependent on document contents and query while no training corpus is used. The latter correspond roughly to general search engines such as Google, Yahoo and the infamous Windows search function while the former corresponds to the machine learning approach to ranking: learn a model to predict relevance from training data and apply it to generate a ranking.[6] An excellent result of a quite simple ranker prompted us to combine these two approaches in a straightforward manner, improving the result further. We intend to address the combination of these ranking methodologies more comprehensively in the future.

Furthermore, we will also estimate the risk of our local searching approach. That is, we aim to answer whether it would be preferable to have the full MEDLINE[7] database available directly instead of sending queries to PubMed[8], and ranking only the retrieved document references – which is our current local searching approach.[9]

[4] More details and publications see `trec.nist.gov`

[5] The ranking is simply the sorted list of documents according to score, i.e. the first document has the highest score, the second has the second-highest score and so on.

[6] This implicitly assumes that a model which predicts relevance sufficiently well is also a good ranker. This is not always the case – for example, in our experiments we found that models which perform worse at prediction may perform better at ranking and vice versa.

[7] MEDLINE is a large comprehensive repository of around twelve million bibliographic references, managed by the U.S. National Center for Biotechnology Information (NCBI). A few thousand references are added daily from a variety of sources.

[8] `www.pubmed.org`, managed by the U.S. National Center for Biotechnology Information (NCBI), is a retrieval engine for the MEDLINE database.

[9] It should be mentioned that *local searching* as defined here may be a misnomer.

Lastly, we report an investigation on the usefulness of homonymy recognition and find the recognition to work well but yield no improvement in terms of ranking performance.

Our experiments are based on the medical annotation dataset by the Swiss Institute of Bioinformatics which has been the subject of previous papers, e.g. [3].

2 Motivation

A web-based prototype system for the BioMinT project is in development at our institute. It currently offers the functionality to expand query terms via synonyms extracted from fourteen online databases (see Section on Synonym Expansion); to retrieve bibliographic references from MEDLINE via the online PubMed search engine; and to process these collection to create ranked document listings via diverse ranking algorithms, four of which have been chosen for our experiments here.

We were at first interested in investigating the relative performance of our rankers, and especially in the strengths and weaknesses of query-dependent classical ranking algorithms (i.e. those who determine relevance by computing similarity with the query) versus query-independent learning rankers (i.e. those who determine relevance by generalizing from a given collection of relevant and non-relevant documents, not taking the specific query to be answered into account).

Due to licensing problems it is at present moderately difficult to maintain a local snapshot of the full MEDLINE database even for research purposes; and almost impossible in a commercial setting, at least for non-U.S. companies. So our current approach is to extend the query with synonym expansion (=oversearching), send the query to PubMed and afterwards locally postprocess the retrieved document set with filtering and ranking approaches. We call our approach *local search*. The second main point of this work is thus to determine whether this approach is competitive to having the full MEDLINE database locally indexed, and if not how high the risk is to us it in the future – in terms of significant differences in ranking performance.

Lastly, word sense disambiguation in biology remains challenging (see e.g. [11]). Related to this is the problem of homonymy – a single protein/gene name may refer to multiple protein/gene entities. While we have insufficient data for proper word-sense disambiguation, it is still possible to investigate whether removal of homonyms from the query improves the ranking. This is what we investigated in the third and final experiment. Our current synonym database suggests a simple way to recognize homonyms which has been preliminarily validated by domain experts and looks very promising.

These are the three main topics for this work. We will now proceed to explain the synonym expansion process and describe the medical annotation dataset, followed by experimental setup and – finally – our experimental results concerning these three topics.

3 Synonym Expansion

The synonym expansion is based on a composite database of protein and gene names and synonyms created from fourteen online databases. Here, we focus on species Homo sapiens, so only five databases are relevant: SwissProt[2], Genew[9], OMIM[8], LocusLink[10] and GDB[11].

As general procedure we have extracted all name from appropriate fields of a given entry and created all pairwise combinations of these synonyms, combined with species and source information, as separate entries in the synonym tables. This procedure mainly relies on synonym information being symmetric (i.e. if A is a synonym of B, then B is a synonym of A) and on the synonyms from a given entry being concerned with the same gene or protein.

Several databases contain references, or links, to corresponding entries in other databases. Only links where both endpoints exist, and which refer to the same species, are processed. This additional information is also integrated into the database as follows.

We assume that database links are symmetric and transitive. Both should be instantly obvious from the fact that entries are linked only when they refer to the very same gene or protein. We have accounted for this fact by considering all links symmetric and extending the link structure by transitive closure. Thus, two entries are linked if and only if there is a path of length greater than zero between them in the link graph.

An alternative view of this process is that we partition the link graph into distinct subgraphs, each of which is not connected to any other subgraph. The entries within each subgraph are connected by link paths of arbitrary length. We have called each subgraph a synonym group, and assigned an unique number to it. As a special case, single database entries without associated links are also considered a synonym group and assigned an unique number. We then consider all entries in each group to be part of a super-entry, consisting of names from all those entries; and extend the database with these new synonym pairs.

During the creation of the synonym database we noticed quite a few unusual protein names, which have been brought to the attention of domain experts. These have recently provided cleaning rules which are applied after each entry has been processed.

The current release of the synonym database was updated on 7th of June and contains 501,866 unique names; 11,277,791 unique synonym pairs (including source database, id and source field data) and 329,257 unique synonym groups from a total of 7,395 unique species.

---

[10] `http://www.ncbi.nlm.nih.gov/LocusLink/`, a composite database managed by the U.S. National Center for Biotechnology Information.

[11] `www.gdb.org`, GDB Human Genome Database (GDB) was developed and maintained by the The Hospital For Sick Children, Toronto, Ontario, Canada (1998 - 2002), and Johns Hopkins University, Baltimore Maryland, United States of America (1990-2002). In January 2003, GDB-related software and public data were transferred to RTI International. RTI continues to host GDB as an open, public resource.

4 Medical annotation dataset

The medical annotation dataset is concerned with the relevance of documents encountered during annotation of thirty-two genes for SwissProt. It contains 32 queries and 2,188 documents classified as Good (relevant), Bad (irrelevant) or Unclear (insufficient data to determine relevance). We removed documents classified as Unclear since the task of determining whether insufficient data for relevance determination exists is orders of magnitude harder and not as interesting to study as the task of learning models for known relevance. This approach is equivalent to assuming a missing class value for Unclear. Thus we follow the TREC[12] methodology in that we assume relevance to be a binary value, which greatly facilitates evaluation and comparison of different approaches. 1,834 documents remain after removal of class Unclear, of which 20% are assigned to class Good (relevant). Specific information on the queries is shown in Table 1.

We have arbitrarily chosen nine queries from the medical annotation dataset for testing, and the others for training. Of the remaining 23 queries, we removed two because they did not contain any relevant documents; and one because its query cannot be represented in our current prototype since it contains a negation[13].

The name of each query is equivalent to a gene name, which in turn refers to the main search term used for a PubMed query by the annotators. We expanded the main search term via synonym expansion, adding all unique names from within all synonym groups that contain the search term. We restricted the search to Homo sapiens. We also added the six search terms (mutation mutations variant variants polymorphism polymorphisms) to recreate the original queries as closely as possible.

Since about a year has gone by since the creation of the original medical annotation dataset, it is not surprising that most queries now return more documents ($\forall$ #QDocs>#Docs). Since the relevance of new documents is not known to us, we have chosen to evaluate mainly those documents whose relevance is known from the medical annotation database – except for Avg. rel.Rank which is computed over the current query (see Evaluation Measures).

5 Experimental setup

5.1 Query construction

As we mentioned earlier, each query was constructed from the expanded main search term, which corresponds to the query name. Synonym expansion is done by looking up all synonyms in our synonym database for species Homo sapiens. Additionally, the following search terms were added: (mutation mutations

[12] TREC is a series of yearly **T**ext **RE**trieval **C**onferences organized by the U.S. National Institute of Standards and Technology, see `trec.nist.gov`. The TREC conferences have been centered around specific text mining problems from the beginning in 1992, always in a competitive setting.

[13] It is quite feasible to extend this, but this feature has not yet been implemented.

Table 1. Medical annotation dataset partitioned into training and test queries. #QDocs, returned documents for expanded query as of July 2004; #Docs, documents in original query as of 2003; #RDocs, relevant documents in original query. * denotes removed queries, see text.

Query	#QDocs	#Docs	#RDocs
Test, for all rankers			
wt1	660	128	17
ump synthase	146	13	1
xpa	993	131	1
vhl	590	299	58
wrn	247	137	9
xpc	199	50	2
wfs1	93	17	11
GCDH	77	11	9
tulp1	24	17	1
Train, for query-independent rankers only			
ADRB1*	116	1	0
CDH1	744	32	4
ESR1	3709	80	3
GLB1	3401	4	1
LPL	789	234	65
MRP1	1560	49	3
abcb1	949	100	9
mrp2	839	18	4
mrp6	669	14	10
sur1	2241	78	14
tgfbr2	11349	11	1
tgm1	6640	22	11
tpo not thrombopoietin*	403	66	9
triosephosphate isomerase	312	101	16
tsc1	867	112	6
umps*	146	6	0
urod	270	10	6
uroporphyrinogen-III synthase	41	28	17
vdr	1015	36	9
vmd2	4519	9	6
whn	33	27	1
zap70	8680	7	1
zic3	34	7	1

variant variants polymorphism polymorphisms). Query terms were enclosed in double quotes ("). Other search terms were similarily treated and concatenated to the original query via AND. For example, the final query for tulp1 was:

```
("RP14" OR "tubby like protein 1" OR "Tubby related protein 1"
 OR "Tubby-like protein 1" OR "Tubby-like protein-1" OR "TUBL1"
 OR "TULP1") AND ("mutation" OR "mutations" OR "variant" OR
                  "variants" OR "polymorphisms" OR "polymorphism")
```

The final query was then sent to the online PubMed search engine and all documents were retrieved and processed by the rankers which we shall now describe in turn.

5.2 Rankers

We chose four rankers for our evaluation – two classic ranking systems which assign scores to documents based on their similarity to the query (LR and SR), and two learning systems who try to learn scores for documents based solely on their content without reference to the query (NBR, ORR). The former are called query-dependent and the latter query-independent rankers because of this important distinction. The query-independent rankers utilize documents from all training queries for learning while this source of information is not used by the

query-dependent rankers. For the query terms, it is vice versa. RND is a special case which estimates query complexity as the performance of a random ranker.

- LuceneRanker (LR) utilizes the java-based text indexing and retrieval engine Jakarta Lucene[14] for ranking. The high performance of this engine allows us to index all given query documents on-the-fly and search the index for the final query. Lucene has also been used as competitive baseline for the TREC 2003 Genomics Track competition.
  Lucene uses the following formula to compute the score for each document. No term boosting was used: $\forall t : boost_t = 1$.

$$score_d = coord_{qd} \sum_t tf_q \frac{idf_t}{norm_q} tf_d \frac{idf_t}{norm_{dt}} boost_t \tag{1}$$

  where

$$score_d = \text{score for document d} \tag{2}$$
$$coord_{qd} = \text{number of terms in both query and document divided by number of terms in query} \tag{3}$$
$$tf_q = \text{the square root of the frequency of t in the query} \tag{4}$$
$$idf_t = \log \frac{numDocs}{docFreq_t + 1} + 1.0 \tag{5}$$
$$numDocs = \text{number of documents in index} \tag{6}$$
$$docFreq_t = \text{number of documents containing t} \tag{7}$$
$$norm_q = \sqrt{\sum_t (tf_q idf_t)^2} \tag{8}$$
$$tf_d = \text{the square root of the frequency of t in d} \tag{9}$$
$$norm_{dt} = \text{sqrt number of tokens in d and same field as t} \tag{10}$$
$$boost_t = \text{the user-specified boost for term t} \tag{11}$$

  A variant of LR, LuceneIndexRanker (LIR) will be used for evaluating local search. The only difference is that LIR uses a one-year snapshot of MEDLINE as background database and adds the documents from the local query to this index before searching this larger index. This simulates what the search would be like if we were to create an index on the full MEDLINE database.

- SimpleRanker (SR) ranks by a simplified score. For each document, it computes the proportion of query terms which actually appear in the document. This ranker is intended to serve as a simplified baseline to the more refined score computation by LR, but has the advantage that documents can be ranked instantly without having to wait until the full document collection becomes available. The latter is necessary for LR as all documents are needed to build the full-text index.

[14] `http://jakarta.apache.org/lucene`

- NaiveBayesRanker (NBR) utilizes a pre-trained Naive Bayes classifier for ranking. Naive Bayes is a common machine learning algorithm based on Bayes' Rule, see [4]. Probabilistic classifiers like Naive Bayes have been shown to tackle the problem of document relevance ranking for biomedical literature successfully, see [7, 10]. Title and abstract of each document are transformed into a bag-of-words representation prior to processing, where each word is represented by one attribute encoding binary occurrence. The top 1000 most frequent words appearing in documents from the training queries plus the document's classification as Good or Bad were used for training this classifier. No stemming, lexical preprocessing, or normalization took place. NBR outputs a numeric score, i.e. the probability of a document being relevant, estimated from the training data.
- OneRRanker (ORR) also utilizes a pre-trained model like NBR. However, the model by ORR is much simpler and consists of one rule based on a single word – the word which yields most information on the class, see [6]. In our case, the rule obtained from training data was:

  ```
  IF 'missense' appears in document THEN score=1.0 (relevant)
                                    ELSE score=0.0 (irrelevant)
  ```

  Contrary to NBR, this ranker can only output binary scores (either 0 or 1) which means that all documents considered relevant or irrelevant will be output in input order, i.e. reverse chronological which is usually not well correlated to any reasonable relevance order[15]. This is usually not a good property for a ranker since this means it cannot hedge its bets – all relevant documents are considered equal as are all irrelevant documents. Ordering within the set of relevant/irrelevant documents is given by the document source, and not by the ranker. Exactly the same training data as for NBR was used here.
- RandomRanker (RND) is an even simpler baseline than SR. It shuffles the input documents randomly, corresponding to assigning a random numeric score to each document – independent of its contents! RND is obviously not suited for meaningful ranking, but measures the complexity of each test query – queries with mostly relevant documents will perform quite well with this approach, while queries with few or only one relevant document will fare poorly. It should be easy to beat RND, which is indeed the case. The measures for RND have been averaged over 1000 runs for each query to yield more stable estimates.

All rankers and supplementary tools are written in Java which facilitated interoperability. We also removed nine documents which were present in both training and test queries since this may lead us to overestimate the performance of the system[16]

[15] This is due to the search via PubMed which returns documents in reverse chronological order (i.e. the newest document is on top)

[16] In unpublished experiments, changes from 0.74 to 0.64 in average precision were observed when removing overlapping documents. There are two viewpoints on this:

5.3 Evaluation

We have chosen three measures for evaluating our rankers. Let us assume that the ranked documents are numbered as $d_1, d_2, d_3, ...d_n$ in ranked order, where d_1 is predicted to be most relevant. Let the index of the ith relevant document be r_i and the total number of relevant documents be R. Let R_i be the number of relevant documents which are returned before or at i, i.e. the size of the set $\{r_i <= i\}$. Then $Prec(i) = \frac{R_i}{i}$, $Recall(i) = \frac{R_i}{R}$ are precision and recall after returning the ith document.

- Average precision (Avg.Prec), i.e. the mean of precision at each relevant document retrieved (defined as in TREC: $\sum_{i=1}^{R} Prec(r_i)$).
- Precision-Recall break even point (PRBE), i.e. the point within the ranking where precision equals recall ($Prec(i) == Recall(i)$ for any i). If it is not defined, we chose to take the i in the ranking where precision and recall differ least ($=i_{minDiff}$), and computed the average of recall and precision there: $\frac{Prec(i_{minDiff})+Recall(i_{minDiff})}{2}$. Contrary to average precision which is hard to interpret, this is a real recall/precision value which can actually be achieved by cutoff at $i_{minDiff}$.
- Average relative rank (Avg.rel.Rank) in the current query from July 2004. Here, the rank of all relevant documents within the larger and more current query is averaged and normalized to the number of returned documents (#QDocs in Table 1). This value tells us how the relevant documents are distributed in the present query and roughly at which place we would expect a known relevant document to appear on average. Because we do not know how many of the documents ranked before known relevant documents are also relevant, this may be of limited use. Here, smaller values indicate better performance.

For each comparison, all three values are reported. Also, arithmetic average of *Average precision*, *PRBE* and *Avg. rel.Rank* over the nine test queries is reported throughout. We chose not to use standard deviation over Average precision because it is not usually used in TREC evaluation. Also, the queries are of quite different complexity (see RND) so a proper normalization procedure would have to be devised. It is not obvious how to achieve this in a fair manner.

6 Results

6.1 Ranking comparison

The results of the ranking can be found in Tables 2, 3 and 4. Surprisingly, the simplest query-independent ranker ORR is best both on mean Avg.Prec and on

One, we might assume that some overlap between queries is realistic and do not bother. Two, we might ensure that training and test set are not overlapping to prevent such overestimation of performance. We have chosen the second approach since an overlap of nine documents for nine test queries is quite significant and – given that three of the queries have only one relevant document – could potentially bias results dramatically. So we chose to err on the side of caution.

Table 2. Average precision. Avg., mean average precision over all queries.

	LR	SR	NBR	ORR	RND	LR_M
wt1	0.199	0.164	0.267	**0.366**	0.165	0.364
ump s.	**1.000**	**1.000**	0.333	**1.000**	0.245	**1.000**
xpa	**0.500**	0.333	**0.500**	0.019	0.043	0.250
vhl	0.449	0.407	0.617	**0.677**	0.209	0.604
wrn	0.698	0.438	0.462	0.282	0.096	**0.699**
xpc	0.292	0.171	0.500	0.559	0.106	**0.700**
wfs1	0.874	0.884	**0.930**	0.873	0.700	0.907
GCDH	0.977	**1.000**	0.878	0.792	0.863	0.977
tulp1	0.091	0.111	0.333	**1.000**	0.211	0.100
Avg.	0.564	0.501	0.536	0.619	0.293	**0.622**

mean PRBE; NBR wins on mean Avg.rel.Rank which is a less reliable indicator of performance. Generally, query-dependent approaches (LR and SR) perform satisfactorily given that they did not use any information except the query terms themselves but there is a slight performance gap.

The excellent result of ORR inspired us to try another experiment, shown in the last column of the results tables as LR_M. Here, we simply added the single term 'missense' (= the word learned by ORR) to the queries and reran LR. This results in improvements for most queries and makes LR_M the best ranker both by mean average precision and mean PRBE.

The results are intriguing in more than one sense. For once, LR_M and ORR perform comparable to [3] who reported 58.89% precision and 69.28% recall.[17], even though we did not use stemming, lexical normalization, or biological background knowledge; only 67% training data instead of the 80% implicit in their five-fold cross-validation; and although they did not remove overlapping documents between training and test queries which may have lead to an overestimation of precision.[18]

Our results agree well with common wisdom within text mining that ranking approaches with simple word vector representations are competitive to much more elaborate approaches.[19] What is even more intriguing is that the word 'missense' which is so useful in ranking relevant documents is not even mentioned in [3], although some multi-word phrases containing this word are mentioned.

Table 3. Precision-Recall break even point. Avg., mean of PRBE.

	LR	SR	NBR	ORR	RND	LR_M
wt1	0.118	0.118	0.294	**0.353**	0.135	**0.353**
ump s.	**1.000**	**1.000**	0.667	**1.000**	0.629	**1.000**
xpa	**0.750**	0.667	**0.750**	0.510	0.521	0.625
vhl	0.414	0.431	0.586	**0.690**	0.192	0.534
wrn	**0.667**	0.333	0.444	0.333	0.109	0.556
xpc	0.375	0.350	**0.500**	**0.500**	0.311	**0.500**
wfs1	0.727	0.727	**0.818**	0.727	0.647	0.727
GCDH	0.889	**1.000**	0.889	0.778	0.814	0.889
tulp1	0.545	0.556	0.667	**1.000**	0.594	0.550
Avg.	0.549	0.518	0.562	0.589	0.395	**0.637**

Table 4. Average relative rank, i.e. average rank of relevant documents in current query from July 2004 normalized by the number of documents returned. Avg., mean of Avg. rel.Rank over all queries.

	LR	SR	NBR	ORR	LR_M
wt1	0.410	0.520	**0.283**	0.497	0.288
ump s.	0.026	**0.007**	0.027	0.075	0.026
xpa	0.031	**0.018**	0.023	0.912	0.053
vhl	0.319	0.312	0.225	0.234	**0.213**
wrn	**0.098**	0.220	0.126	0.474	0.113
xpc	0.101	0.186	0.070	0.460	**0.054**
wfs1	0.268	0.316	**0.185**	0.428	0.252
GCDH	0.142	**0.091**	0.228	0.707	0.166
tulp1	0.619	0.375	0.167	**0.042**	0.524
Avg.	0.224	0.227	**0.148**	0.425	0.188

6.2 Risk of local search

To determine the risk of our local search approach, we compared LuceneRanker (LR) to LuceneIndexRanker (LIR). The only difference between both rankers is that while LR creates a local index of all documents within the current query, LIR adds all documents within the current query to a one year snapshot of MEDLINE obtained via TREC[20] consisting of half a million MEDLINE references indexed between 1st April 2002 and 2003. We consider this to be a reasonable

[17] Compare Table 3 – PRBE gives a value of e.g. precision=recall=58.9% for ORR and 63.7% for LR_M

[18] In unpublished experiments on this very dataset, not removing overlapping documents led to an overestimation in average precision by 0.1(!)

[19] Personal communications, Walter Daelemans.

[20] `trec.nist.gov` This is the snapshot that was used for the TREC2003 Genomics track and was later made publicly available.

Table 5. Performance comparison between LuceneRanker (LR) and LuceneIndexRanker (LIR). Avg.Pr., Average precision; PRBE, Precision-Recall Breakeven point; Avg. rel.Rank, average rank of relevant documents in current query from July 2004 normalized by the number of documents returned.

	Avg.Pr.		PRBE		Avg. rel.Rank	
	LR	LIR	LR	LIR	LR	LIR
wt1	0.199	0.236	0.118	0.176	0.410	0.344
ump s.	1.000	1.000	1.000	1.000	0.026	0.026
xpa	0.500	0.143	0.750	0.571	0.031	0.139
vhl	0.449	0.463	0.414	0.379	0.319	0.311
wrn	0.698	0.254	0.667	0.333	0.098	0.228
xpc	0.292	0.333	0.375	0.417	0.101	0.083
wfs1	0.874	0.856	0.727	0.727	0.268	0.284
GCDH	0.977	1.000	0.889	1.000	0.142	0.120
tulp1	0.091	0.083	0.545	0.542	0.619	0.667
Avg.	0.564	0.485	0.609	0.572	0.224	0.245

approximation to using the full MEDLINE database of twelve million entries. Table 5 shows the results. As can be seen, LIR performs somewhat similar to LR, and on average performs worse. Clearly, LIR does not perform much better as may have been expected from the fact that term and document frequencies are better estimated in the larger index. It seems that small, query-dependent full-text search may also have its advantages.

Concluding, the risk of our local search approach seems to be marginal. It seems that having a local MEDLINE installation is not essential.

6.3 Homonymy recognition

Lastly, we investigated whether removal of homonyms from the expanded queries improves the ranking. Based on the reasonable assumption that each synonym group concerns a single protein/gene entity, we consider homonyms to be names which appear in more than one synonym group. Five[21] of our nine test queries had at least one term which was present in more than one group, see Table 6. We removed these terms from the queries and reran LR plus SR. ORR and NBR did not show any changes, since exactly the same set of documents was returned for each query.

Results indicate that the improvement is marginal at best and slightly negative at worst. Overall the performance is almost indistinguishable. Generally, 3.9% of synonyms for species Homo sapiens within our database are homonyms according to our approach which roughly agrees with the proportion of search

[21] Initially we additionally found two homonyms for wfs1, but feedback from domain experts enabled us to trace the wrong homonym to an erroneous entry in an imported online database, which has since then been corrected. All entries shown here are verified homonyms.

Table 6. Predicted synonyms, separated by comma. These were removed from each query for the homonymy recognition experiments.

Query	Homonyms
vhl	HRCA1,RCA1
xpc	p125
wrn	RECQL2,RECQL3
tulp1	RP14
wt1	WAGR

Table 7. Left: Average precision. Avg., mean average precision. Right: Precision-Recall break even point. Avg., average PRBE. LRh, SRh are ranking results where homonyms were removed from the query.

	LR	LRh	SR	SRh
wt1	0.199	0.210	0.164	0.169
vhl	0.449	0.442	0.407	0.416
wrn	0.698	0.649	0.438	0.438
xpc	0.292	0.292	0.171	0.171
tulp1	0.091	0.091	0.111	0.111
Avg.	0.434	0.420	0.363	0.365

	LR	LRh	SR	SRh
wt1	0.118	0.118	0.118	0.118
vhl	0.414	0.414	0.431	0.431
wrn	0.667	0.556	0.333	0.333
xpc	0.375	0.375	0.350	0.350
tulp1	0.545	0.545	0.556	0.556
Avg.	0.474	0.456	0.419	0.419

terms removed for our test queries. Thus, the practical consequences of the homonymy problem seem to be negligible in our case.

7 Related Research

[3] report on a refined approach to predict relevance from the same medical annotation dataset. They use normalisation of gene/protein names, a special Part-Of-Speech tagger, feature selection and creation of new features based on Journal names, which was input into a Probablistic Latent Categorizer (PLC). Their results are comparable to our much less elaborate approach which follows the basic machine learning approach towards ranking.

[5] gives a good overview of current approaches in literature data mining, also including some approaches to ranking.

[11] is a very comprehensive approach to named entity recognition of protein and gene names from biological literature. He also tackles word sense disambiguation shortly with good success. Parts of his work have been integrated into the GeneWays project. A synonym resource somewhat similar to the one used within BioMinT can be found at `http://synonyms.cs.columbia.edu/` and is based on this work. Contrary to our resource, it also incorporates information extracted from references and full-text papers, and its synonyms have been extensively reviewed by domain experts.

[7] introduces a system to discriminate papers concerned with protein-protein interactions from others papers. They used a Bayesian approach and a log-likelihood scoring function and report promising results. Results are given in

Table 8. Average rank within recent query. Avg., mean of average relative rank. LRh, SRh are ranking results where homonyms were removed from the query.

	LR	LRh	SR	SRh
wt1	0.410	0.390	0.520	0.510
vhl	0.319	0.325	0.312	0.313
wrn	0.098	0.120	0.220	0.224
xpc	0.101	0.116	0.186	0.187
tulp1	0.619	0.619	0.375	0.375
Avg.	0.302	0.310	0.322	0.322

forms of coverage, accuracy and log-likelihood distributions, none of which can be easily compared to our results.

[10] use boosted Bayesian classifiers and Support Vector Machines to learn a discrimination model for papers that should be included into a speciality database. Negative examples were generated by using related documents from MEDLINE which are not part of the speciality database and thus are assumed to have been rejected. They report average precision on the top 100 documents of 80% for the best system. However, their task is completely unrelated to medical annotation and so also cannot be compared.

`www.e-biosci.org` is a resource that has grown out of another EU research project. While not specifically adressing document relevance ranking, its goal is somewhat similar to BioMinT, namely "...a next generation scientific information platform that will interlink genomic and other factual data with the life sciences research literature."

8 Conclusion

We investigated the relative performance of our rankers on a dataset dealing with medical annotation. Surprisingly, a quite simple ranker based on the occurrence of a single word, ORR, was the most successful of the initially considered rankers. In an extension of our experiments, adding this single significant word to each search query yielded an improvement to a query-independent ranker, improving one the simple ranker and yielding comparable results as a state-of-the-art approach from [3]. This is insofar intriguing as we did not use lexical preprocessing or biological background knowledge; and that the word we found is not reported as most significant there, although some multi-word phrases containing 'missense' were reported.

We investigated whether processing the document set returned from querying PubMed is competitive to using a significant subset of the full MEDLINE database locally, in a simplified setting. It turned out that this is indeed the case – if anything, processing the document set from PubMed seems slightly preferable.

Lastly, we investigated whether the removal of automatically recognized homonyms from the query improves the ranking. It turns out this is not the

case, so in the context of ranking for medical annotation the homonymy problem seems negligible.

Acknowledgements

This research was supported by the European Commission as project no. QLRI-CT-2002-02770 (*BioMinT*) under the RTD programme *Quality of Life and Management of Living Resources*, and by the COST Action 282 (*Knowledge Exploration in Science and Technology*). The Austrian Research Institute is supported by the Austrian Federal Ministry of Education, Science and Culture (BMBWK); and by the Ministry for Transport, Innovation, and Technology (BMVIT). We want to thank the Swiss Institute for Bioinformatics for contributing the medical annotation dataset.

References

1. Attwood T.K., Bradley P., Gaulton A., Maudling N., Mitchell A.L. and Moulton G. "The PRINTS protein fingerprint database: functional and evolutionary applications". In Encyclopaedia of Genomics, Proteomics and Bioinformatics, M.Dunn, L.Jorde, P.Little & A.Subramaniam (Eds.), 2004. `www.bioinf.man.ac.uk/dbbrowser/PRINTS`
2. Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M. "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003", Nucleic Acids Research, 31(1):365-370, 2003. `www.expasy.org/sprot`
3. Dobrokhotov P.B., Goutte C., Veuthey A-L. and Gaussier E. "Combining NLP and probabilistic categorisation for document and term selection for Swiss-Prot medical annotation". Bioinformatics, 19 Suppl 1: I91-I94. 2003.
4. Richard Duda and Peter Hart "Pattern Classification and Scene Analysis". Wiley, New York, 1973.
5. Hirschman L., Park J.C., Tsujii J., Wong L., Wu C.H.: "Accomplishments and Challenges in Literature Data Mining for Biology", Bioinformatics Journal 18, pp. 1553–1561., 2002.
6. Holte, R.C. "Very simple classification rules perform well on most commonly used datasets". Machine Learning, Vol. 11, pp. 63-91, 1993.
7. Marcotte E.M. et al. "Mining literature for protein-protein interactions". Bioinformatics, 17, p.359-363, 2001.
8. Online Mendelian Inheritance in Man, OMIM (TM). McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD), 2000. World Wide Web URL: `http://www.ncbi.nlm.nih.gov/omim/`
9. Wain H.M., Lush M.J., Ducluzeau F., Khodiyar V.K., Povey S. "Genew: the Human Gene Nomenclature Database, 2004 updates". Nucleic Acids Res. 2004. 32 Database issue:D255-7, 2004.
10. Wilbur, J.W. "Boosting Naive Bayesian Learning on a Large Subset of MEDLINE". Proceedings of the AMIA Symposium, p.918-922, 2000.
11. Yu, H. "Synonym and homonym resolution of gene and protein names", PhD thesis, Columbia University, U.S.A., 2002.