Searching for geneRIFs: concept-based query expansion and Bayes classification

Rob Jelier, Martijn Schuemie, Christiaan van der Eijk, Marc Weeber, Erik van Mulligen, Bob Schijvenaars, Barend Mons, Jan Kors.

Rob Jelier
Department of Medical Informatics, Erasmus MC
PO Box 1738,3000 DR Rotterdam,The Netherlands
tel +31 10 4088124, fax +31 10 4089447, r.jelier@erasmusmc.nl

Primary Task

Introduction

The Biosemantics group at the Erasmus MC followed a thesaurus-based approach for the first task of the genomics track. The approach is based on a concept-based indexing engine (Collexis®), suitable for large - cale and high-speed indexing. The thesaurus used for indexing was constructed as a combination of the MESH thesaurus and the gene ontology (GO), with a species-specific gene thesaurus derived from LocusLink gene annotations. Our indexing machinery produces per indexed MEDLINE abstract a list of concepts with an accompanying weight, termed a fingerprint. Searching is done by matching a query fingerprint against fingerprints of all indexed MEDLINE abstracts. Query fingerprints are generated by combining fingerprints of four types. First, a fingerprint containing just the gene concept with all the known gene names and aliases. Second, a combination of MEDLINE fingerprints of all abstracts in which the gene concept was found without ambiguity problems. Third, a generic fingerprint with concepts typical of geneRIFs, when compared to MEDLINE in general. Fourth, a fingerprint containing the concepts of the Gene Ontology (GO) annotation

When it comes to identifying a gene name in a text the large number of synonyms and the frequent occurrence of homonymy are problematic. In our approach we attempt to deal with both. Synonymy as found in Locuslink is incorporated in our thesaurus. An attempt was made to reduce the effects of homonymy by expanding the query with fingerprints where the gene name is found unambiguously. Gene specific information, the GO annotation, is included to select for the correct gene, but also to select for abstracts with terms about basic biology. The generic fingerprint is included to select for abstracts with terms about basic biology.

System description

Producing the thesaurus

The Locuslink database is used as a basis for producing a gene thesaurus. Different thesauri were produced for the different organisms. For all the genes described in the database the following annotations were allowed as synonyms: official symbol, preferred symbol, alias symbol, official gene name, preferred gene name, alias protein, preferred product and product. A distinction was made between symbols and long forms of the gene name or product. Before indexing the lvg2002 normalizing engine (http://umlslex.nlm.nih.gov/index.html) is used to normalize the words in the text and make the system more robust. This includes removal of all capitalization. Hence indexing occurs case-insensitive. As an exception to this rule, words are not normalized when at least half of the letters are in capitals. For building our thesaurus gene and protein symbols are not normalized, though long forms are normalized. If a symbol is composed of a letter and number combination the symbol is also included in lowercase. When symbols or long forms end with a number, two forms are included in the thesaurus to better match spelling variation, one with the number directly after the last letter, the other separated with a hyphen.

To expand the thesaurus with concepts from the biomedical domain all concepts of Gene Ontology (http://www.geneontology.org/) and all MESH concepts that have an unique identifier in UMLS are added

(http://www.nlm.nih.gov/mesh/meshhome.html). The structure of these thesauri is not used. All terms are normalized. From the whole of the thesaurus all words with a length of one or two letters are removed.

Preparation of the texts

For the selected MEDLINE records title, abstract and MESH headings are retrieved. One of the variables manipulated in our experiment is the use of the abbreviation expansion algorithm described by (Schwartz and Hearst, 2003) to replace abbreviations with their matching long forms. Our hypothesis is that abbreviation expansion will reduce the ambiguity of the text. Next are the removal of stop words followed by normalization of the remaining words.

Indexing

For indexing Collexis® indexing software is used (http://www.collexis.com). Identified concepts are assigned a relevance score for vector representation. This value is based on term frequency multiplied with a factor selecting against general concepts (see equation 1). The values are subsequently divided by the value of the highest ranking concept of the document, thereby normalizing to a maximum value of 1. This is the data that will be queried. The list of concepts with relevance scores will be referred to as a fingerprint.

$$F_i = \left(\frac{1}{S_i + 1} \right)^{0.3}$$

Equation 1, factor F_i is used to select against general concepts. S_i represents the number of documents a concept c_i occurs in.

Matching algorithm (MA)

To match the search queries to the document fingerprints several formulas were used. The formulas are listed below. f_c q_c represent the value of a concept in respectively the fingerprint and the query. Len(v) represents the length of a vector: Len(v)= sqrt(sum(v_c^2)).

Search queries

Queries are constructed by combining the four search fingerprints:

vector sum(f_c * q_c) / len(f)*len(q)

collexis sum($1/s_i$), s_i represents the number of documents a concept c_i occurs in. Concept c_i is a concept which occurs in both the query and the fingerprint it is compared with.

dice (2*sum(f_c * q_c)) / (len(f)² + len(q)²)

weighted sum(f_c* q_c) * (m_f + 1) / (l_q + 1), where m_f is the number of matched concepts of f, l_q is the number concepts in q.

1. Gene Name (GN). The first search fingerprint is the gene name, including its synonyms.
2. Gene Specific Context (GSC). The second is an expansion of the search with gene specific fingerprints, creating a gene specific context. Fingerprints from the TREC set containing the name of the gene are evaluated and only added to expand the query when they meet the following demands: a. the name (or synonym) of the gene found in the text does not have a homonym in our thesaurus, and it either contains a space (i.e. it is a long-form), or it contains a number (but does not start with a number). b. the abstract or corresponding MESH headings contained the species name associated with the gene.
3. Generic Context (GC). The third fingerprint is constructed based on a database containing all fingerprints from the documents indicated by geneRIFs in Locuslink and on a database containing all fingerprints from the TREC set. All found concepts were extracted from the database and

ordered based on relative overrepresentation in the geneRIF set. Ranking was done based on equation 2. Every concept with a value larger than two is admitted in the generic fingerprint, resulting in a total of 3217 concepts.

4. Gene Ontology (GO). The fourth fingerprint contains concepts representing the GO-annotation for the gene as found in the Locuslink database.

$$C = {}^{2}\log\left(\frac{\left(\frac{S_{geneRIF}}{T_{geneRIF}}\right)}{\left(\frac{S_{TREC}}{T_{TREC}}\right)}\right)$$

Equation 2, The score C calculated using this equation is used to rank concepts for the generic fingerprint, S indicates the number of documents a concept occurs in, T the total number of documents in that set.

The query is constructed by combining the search fingerprints. For combination the weights assigned to the concepts of the different fingerprints are multiplied with a scaling factor, and combined by addition to the other fingerprints. Prior to matching the concepts of the query are multiplied with the factor calculated with equation 1, followed by normalization to 1.

Experiments

Various aspects of our system were tested using the TREC training set in order to find the optimal settings to be used for our final submission. The results presented in this paper may differ from those used for our submitted runs, because of the elimination of several errors in our software.

Variations in combinations

To find the optimal combination of fingerprints, abbreviation expansion and matching algorithm, an experiment was performed evaluating a large number of possible combinations. The different variables were tested with the following discrete values:

- Abbreviation Expansion (AE): on or off
- Gene name fingerprint: weight of 0, 0.5 or 1
- Other fingerprints: weight of 0, 0.1, 0.3 or 0.5
- Matching coefficient: Vector, Dice, Weighted or Collexis

In total 1448 variations were constructed, and for each combination performance on the training was tested.

Variation in the Gene Specific Context

This experiment was aimed at evaluating the contributions of the various requirements used to select those abstracts that will be combined into the GSC fingerprint. Using the optimal combination found in the previous experiment, the system was tested using the gene specific context constructed by varying the following requirements:

- Ambiguity requirement: on or off
- Species name requirement: on or off

Evaluation

The system was used and evaluated according to the standards of the TREC genomics track. The document collection consisted of 525,938 MEDLINE records where indexing was completed between 4/1/2002 and 4/1/2003. The training set were the 47 topics distributed by the track. GeneRIFs taken from LocusLink were the documents to be retrieved for every topic. The test set are the official topics for the TREC competition. As a measure for evaluation mean average precision (MAP) was used.

Comparison of test and training set

To test whether differences existed between the composition of the test and training set we also used our optimization scheme for the test set (after submitting results). Additionally, we calculated the difference in the ratio of # geneRIFs / # retrieved documents, for test and training set.

Manual evaluation of results

An expert in molecular biology manually evaluated the first 10 retrieved documents for 10 queries of the test set. As a standard for a good result the definition as distributed by the TREC organization was used:

> *For gene X, find all MEDLINE references that focus on the basic biology of the gene or its protein products from the designated organism. Basic biology includes isolation, structure, genetics and function of genes/proteins in normal and disease states.*

Results

Variations in combinations

In table 1 the 15 highest scoring settings for our system are represented. Table 2 shows the highest score when the condition in the first column is true. This allows comparisons between different parameters. For all pairs a statistical test was performed (paired t-test, N=47) to assess whether observed differences between settings are significant at the .05 level. The best settings have a higher score than the baseline consisting of a query with only the gene name (p=0.047). Also GC=0.5, GN=0 and MA=collexis scored significantly lower than the optimal settings.

Table 1, MAP scores for 15 highest scoring settings. Abbreviations: GN, genename; GSC, gene specific context; GC, generic context fingerprint; GO, go annotation; MA, matching algorithm; AE, abbreviation expansion.

	MAP	GN	GSC	GC	GO	MC	AE
1	0.374	0.5	0.1	0	0.1	dice	false
2	0.372	1	0.3	0	0.1	dice	false
3	0.368	1	0.3	0	0.3	dice	false
4	0.367	0.5	0.1	0	0.1	vector	false
5	0.366	1	0.1	0.1	0.1	vector	false
6	0.366	1	0.1	0.1	0	vector	false
7	0.363	1	0.1	0	0.3	vector	false
8	0.361	1	0.5	0	0.1	dice	false
9	0.361	1	0.5	0.1	0	vector	false
10	0.361	1	0.3	0.1	0.1	vector	false
11	0.36	1	0.3	0	0	dice	false
12	0.359	1	0.5	0.1	0.1	vector	false
13	0.359	1	0.1	0	0.1	vector	false
14	0.359	1	0.1	0.1	0.1	vector	true
15	0.359	0.5	0.1	0	0	dice	false

Table 2, maximum average MAP scores achieved when expression in the first column is true. Abbreviations: GN, genename; GSC, gene specific context; GC, generic context fingerprint; GO, go annotation; MA, matching algorithm; AE, abbreviation expansion.

Condition	MAP	Condition	MAP	Condition	MAP
Only GN	0.336	AE=true	0.359	GO=0.1	**0.374**

Only GN & GSC	0.360	GN=0	0.096	GO=0.3	0.368
Only GN & GC	0.338	GN=0.5	**0.374**	GO=0.5	0.343
Only GN & GO	0.341	GN=1	0.372	MA=dice	**0.374**
GC=0	**0.374**	GSC=0	0.341	MA=vector	0.367
GC=0.1	0.366	GSC=0.1	**0.374**	MA=weighted	0.331
GC=0.3	0.309	GSC=0.3	0.372	MA=collexis	0.269
GC=0.5	0.237	GSC=1	0.361		
AE=false	**0.374**	GO=0	0.366		

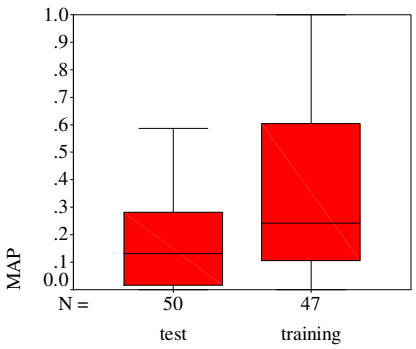
Variation in the gene specific context

Observed differences are not significant at the .05 level according to evaluation with the paired t-test.

	+Species	-Species
+Non-ambiguous indexing	0.37	0.36
- Non-ambiguous indexing	0.34	0.34

Table 3, average MAP for different ways to produce gene specific context, other parameters same as at the best configuration

Queries with the test set



	Average MAP
test	.17
training	.37

Figure 1, results on training and test set, boxplot and average MAP

Comparison of test and trainingset

Manual evaluation of results

Abstracts Analyzed	100
Relevant, geneRIF	31
Relevant, not geneRIF	53
Disputable*	14
Irrelevant	2

Table 4, summary of results of expert evaluation, first 10 results for 10 query. The annotator called an document Disputable when it contained content about the gene but less about function or also about (many) other genes

Discussion

The simplest query within our system, only the gene name, already gives a reasonable average MAP of 0.33. After optimization, the use of the gene specific fingerprint and the GO annotation fingerprint led to a better result than optimized settings for the gene name alone. Let it be noted that most of the other observed differences in table 1 are not statistically significant and that a larger set would be required to determine real effects. The results give a weak indication that the gene specific context is mostly responsible for the improvement. The GO annotation fingerprint could have a neutral or slight positive effect. The significance of the generic fingerprint is difficult to asses, though it clearly has a detrimental effect when given a large role in the query.

When the gene specific context is studied in more detail, the conditions for inclusion of fingerprints seem to play an important role. Though the differences in table 2 are small and may potentially reflect no real effects, there appears to be a trend towards better performance with more specific context.

The role of the abbreviation expansion algorithm is apparently minimal. One could expect more specific indexing (and better performance) as ambiguous acronyms are removed. On the other hand it has been observed that the long forms that are put in place have much more variations and hence are more likely not to be included in the thesaurus.

The optimal settings retrieved for the training set resulted in an average MAP score of 0.37. This appears to be reasonably good when compared to the preliminary runs reported previously (e.g. 0.35 by Prof. Jacques Savoy with the SMART system) on the TREC website. A very different score was achieved for the test set, an average MAP of 0.17. The boxplots in figure 1 clearly show that a very different distribution of MAP scores exists for the test set. A possible explanation could be that the system was over-trained on the training set and that the optimal result on the training set is far from robust. When we optimized settings for the test set the MAP score improved only modestly (MAP = 0.19), which rules out over-training. A better explanation would be that the test and training set are different in composition. After preliminary experimentation with the training set during the preparations for the TREC it was decided to exclude genes for the testset that have only one or two geneRIFs. This most likely led to the striking difference in the average number of geneRIFs per gene, 6.2 for the training set and 11.3 for the test set. Also we found a significantly lower ratio of # geneRIFs / # retrieved documents by our system for the test set. If our system can not distinguish very well between geneRIFs and other retrieved documents, the geneRIFs become more spread out in the retrieved list of documents. This would explain, at least for a part, the difference in the results between the test and training set.

Expert evaluation of retrieved documents showed some clear tendencies, which were also noted by the evaluation by the group of William Hersh (TREC genomics overview presentation, 2003). All checked geneRIFs were considered to be appointed appropriately. A large number of other retrieved documents, however, also appear to fit the description of a geneRIF. It therefore appears the collection of geneRIFs is incomplete. This makes the value of the evaluation of retrieval experiments with geneRIFs as standard difficult to assess.

Conclusion

The system of combining four different fingerprints was successful in improving performance relative to a query with the gene name. The most important contribution to this improved performance appears to be from the gene specific context fingerprint.

Results on the test set were much lower and different from those on the training set. This appears to be caused by a difference in composition of the sets.

Expert evaluation of queries showed that numerous results matched the given definition of a geneRIF but were not annotated as such. Large incompleteness in annotation and lack of difference between positives and some non-positives makes comparison of results very difficult.

Secondary Task

Introduction

With regard to the second task, our starting point was the observation that the GeneRIF annotators in approximately 42% of the cases simply use (part of) the title of a paper as annotation, in spite of the GeneRIF guideline that annotations are '*preferably more than a restatement of the title of the paper'*. Even when the title was not used, a test using the classic dice coefficient showed that the annotation usually matched (part of) a sentence from the abstract. For this reason we have reduced the problem from generating a GeneRIF to mimicking the annotator's choice for a certain sentence or title. We have approached this problem as a classification task using the naïve Bayesian classifier as proposed in (Mitchell, 1997).

Methods

The classifier constructed for this task assigns a given abstract to a class v_j that represents the choice to use one of the sentences as the annotation, where we define the title as sentence 0. To determine the number and type of the classes the set of 38.193 GeneRIFs was examined. For each GeneRIF we computed the classic dice coefficient between the annotation and each of the sentences and title of the corresponding abstract. In 16.163 (42 %) of the cases the annotation matched best with the title.

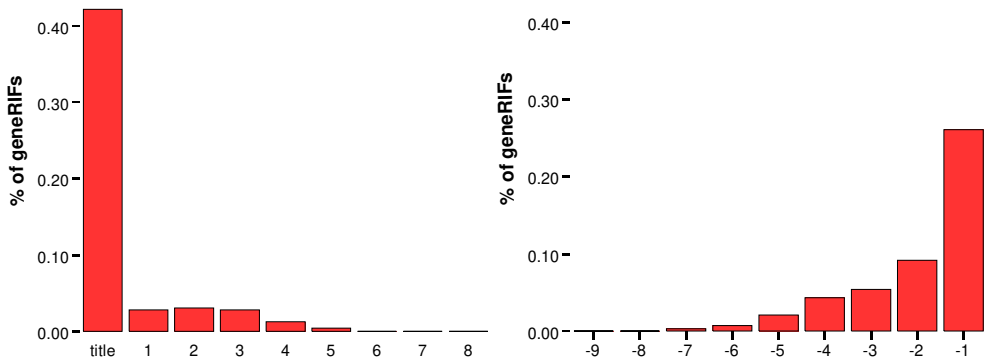


Figure 2, Distribution of positions of the annotation sentence, counting from the start and from the end of the abstract.

Figure 2 shows the distribution of the position of the non-title sentences that best matched the annotation. The first half of the figure shows the distribution of geneRIFs that were mapped to the first 8 sentences or the title of the abstract. The second half shows the same for the last sentences of the abstract. In 3.304 cases (9 %) the annotation was matched to one of the first three sentences and in 17.661 cases (46 %) the annotation matched one of the last five sentences. Because few abstracts (3%) in our test set were matched to annotation positions outside this set of 9, we did not include other positions. As a result, the classifier can assign to class v_j with j=0,..,8 which represent sentence 0 (i.e., the title) as annotation, the first sentence as annotation etc.

As features, we use the presence of normalized words in sentences. Both for training and classification, these features are determined by extracting sentences from the abstracts: nine sentences if the abstract has a length of at least nine, or less if the abstract is shorter. If the abstract is shorter than nine sentences we first extract the title, then the last five sentences, starting with the last moving backwards and then, if any sentences remain, the first three starting with the first. All words in a sentence are then normalized, using the lvg2002 normalizer. We also experimented with other features, such as presence of gene-symbols or thesaurus-based concepts in sentences, but this did not improve the results.

The prior probability of the class v_j is estimated by

$$P(v_j) = \frac{N_j}{N}$$

where N_j is the number of abstracts assigned to class v_j and N the total number of abstracts used for training.
The conditional probabilities of the occurrences $w_{k,i}$ of the normalized word k in sentence i given that the abstract is in class v_j, $P(w_{k,i} \mid v_j)$, is estimated as:

$$P(w_{k,i} \mid v_j) = \frac{n_{k,i,j} + \varepsilon}{n_{i,j} + \varepsilon \cdot d_i}$$

where $n_{k,i,j}$ is the number of occurrences of word k in sentence i in all abstracts in class v_j, and $n_{i,j}$ is the total number of distinct words in all abstracts in class v_j. The factor ε is added to ensure that $P(w_{k,i} \mid v_j)$ is never equal to zero, in which case the absence of a word in a sentence i in class v_j would cancel out all other probabilities in the next calculation. The variable d_i is the number of distinct words in sentence i. We established empirically that ε is best assigned a small value: for our experiments it was set at 10^{-6}.
An abstract a is assigned a class v_j by calculating v_{NB}:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \cdot \prod_{i \in S} \cdot \prod_{k \in W_{a,i}} P(w_{k,i} \mid v_j)$$

where S is the set of sentence position and $W_{a,i}$ is the set of all words positions in sentence i in abstract a and V is the set of all classes.

Results

We trained our classifier on all GeneRIFs, excluding the 139 GeneRIFs that were specified as a test set for this secondary task. On the test set, the classifier correctly found the sentence from which the human annotation was derived in 65.47% of the cases. This compares favorably to the score of 43.88% for the naïve system that selects titles as annotation. However, during the TREC conference it was brought to our attention that our training set might contain GeneRIFs that duplicate with those in the test set, and indeed when we checked a number of LocusLink entries outside the test set turned out to have exactly identical GeneRIFs as those in the test set. This means that some LocusLink entries not only share PMIDs,but – rather surprisingly– annotations as well. Without these duplicates in the training set the performance of our classifier drops to that of the naïve system. The results reported in this section are our initial results obtained using all GeneRIF entries as a training set, only leaving out the test set.
Table 5 shows the results for the 139 test GeneRIFs using the TREC scoring system.

Classic Dice	54.37%
Modified unigram Dice	56.27%
Modified bigram Dice	44.58%
Modified bigram Dice phrases	46.25%

Table 5 Scores for all test GeneRIFs

The classic Dice coefficient was somewhat lower than could be expected from our calculated classification score at 54.37%.
Table 6 shows the scores of all abstracts that according to our measures were classified correctly

Classic Dice	73.26%
Modified unigram Dice	75.93%
Modified bigram Dice	67.30%

Modified bigram Dice phrases	69.32%

Table 6 Scores for correctly classified GeneRIFs

The results in Table 6 suggest that the annotators often change sentences taken from the abstract.
Table 7 shows the scores for incorrectly classified abstracts.

Classic Dice	24.38%
Modified unigram Dice	25.02%
Modified bigram Dice	6.65%
Modified bigram Dice phrases	8.35%

Table 7 Scores for incorrectly classified GeneRIFs

Because some words in the suggested annotation occur in the actual annotation the classic dice score is still 24.38 %. The scores that include some measure of word order are lower, because the two sentences are very different.
Table 8 shows how a hypothetical perfect classifier would perform for the 139 GeneRIFs. This classifier always selects the sentence from the abstract that according to the classic dice coefficient most closely matches the actual annotation.

Classic Dice	70.36%
Modified unigram Dice	72.69%
Modified bigram Dice	62.51%
Modified bigram Dice phrases	64.83%

Table 8 Scores for 139 GeneRIFs using a perfect classifier

The scores in table 8 are very similar to those in table 2, albeit slightly lower, likely because the annotations that were incorrectly classified also differed most from the original sentences in the abstract.

Discussion

Our classifier is capable of selecting the title or sentences of the abstract that best matched the human annotations in 65.47 % of the cases. This is a considerable improvement when compared to the trivial method of always taking the title as the annotation, which yields a score of 43.88 %. It should be noted, however, that our high score relies on duplicate entries in the training set. Once removed, our classifier performs no better than the simple approach of always selecting the title. One could argue that for randomly selected GeneRIFs, which may have duplicate entries, our classifier performs well, but for entirely new GeneRIFs the classifier will perform no better than the baseline algorithm, which selects titles in all cases.
At least two directions for further improvement of our approach can be envisaged. First, the performance of the classifier might be improved; possibly by incorporating other features than the ones we have experimented with so far. However, even a perfect classifier will only give a moderate improvement of the TREC scoring measures. For example, the classic dice score of 54.37 % that was obtained for our current classifier will only become 70.36 % when the classifier is perfect.
A second direction for improvement could be a post-processing of the annotations suggested by the classifier to better match the human annotations. NLP techniques might be helpful in this respect. However, before embarking on further research, it would seem important to assess the quality of the GeneRIFs and to determine in how far a suggested annotation that partially differs from the actual annotations, could serve as a GeneRIF equally well.

References

- Schwartz,AS, M A Hearst, 2003, A simple algorithm for identifying abbreviation definitions in biomedical text: Pac.Symp.Biocomput., p. 451-462
- Mitchell TM. Machine learning. McGraw-Hill: New York, 1997.