Symbol-Based Query Expansion Experiments at TREC 2005 Genomics Track

Michela Bacchin and Massimo Melucci

Department of Information Engineering
University of Padova, Italy
{michela.bacchin, massimo.melucci}@dei.unipd.it

Abstract. This paper illustrates the activity conducted at the TREC 2005 evaluation campaign in the ad-hoc task of the Genomics track. The retrieval effectiveness of a relevance feedback query expansion algorithm, which is based on symbols, is studied. The experimental results suggest that query expansion based on implicit relevance feedback is not always an effective means for improving effectiveness in this domain-specific context.

1 Introduction

The participation of the Information Management Systems (IMS) research group in the ad-hoc retrieval task of the Genomics track aimed at continuing the study started last year. At TREC 2004 the experiments focused on the investigation of different query expansion techniques based on the addition of keyword stems and of genomic product symbols. This year our attention focused on *symbols*, which have been considered in this research as source of evidence for query expansion.

A characteristic of the documents and queries used in used in biomedical domain is the massive presence of symbols. A symbol is a single string which represents an abbreviation for longer words and phrases used to describe genomic, proteic or more generally biomedical products. As the amount of biomedical literature grows, so does the number of symbols per product and the average number of definitions per symbol [15]. This ambiguity makes the retrieval task harder because a short form used to label a particular object in a user query can mismatch the ones used by different authors in relevant documents, even if they refer to the same object.

This paper reports on a study on an automatic query expansion algorithm. Our main aim was to study whether an automatic query expansion algorithm based on relevance feedback can improve the effectiveness of a retrieval system by exploiting the information coded into symbols. The paper is organized as follows: in Section 2 the scenario regarding the query expansion and the use of symbols in biomedical literature is presented. Section 3 reports symbol-based query expansion algorithm. Section 4 and 5 are devoted to the description of the experimental settings and the analysis of results.

2 Symbols in Biomedical Text

In biology and medicine, the employment of abbreviations or acronyms for indicating longer words or phrases is very frequent. A generic biomedical object can be individuated by a set of words or by a symbol which is a shorter form of that description. In this paper the following definition of symbol is given:

> A *symbol* is a string of alphabetical, numerical and special characters, which represents an abbreviation for a longer sequence of words.

This definition includes also the acronyms which are a specialized type of symbols, in fact they consists of the concatenation of word initials, while symbols have no restrictions on letters.

Symbols could be, on the one hand, a valuable source of information because they could identify a biomedical product in a precise manner, but on the other hand, they can not support an exhaustive search because authors could use different symbols to refer to the same product across different research papers or sub-domains, depending on the context in which the symbol is used; for instance, *lymphocyte associated receptor of death* is a protein with several synonymous symbols, such as `LARD`, `Apo3`, `DR3`, `TRAMP`, and `TnfRSF12`. An information retrieval system managing biomedical documents has to cope with the ambiguity problems arising from the use of such symbols in text.

Since Query Expansion (QE) has proved to be an effective method for managing words ambiguity, at least using classic IR test collections [7], [10], [11], the use of this technique could be reasonable also in this domain-specific context to improve the effectiveness level of effectiveness. In this paper it is presented and tested an automatic query expansion algorithm which exploits the symbols for selecting the keywords to be added to the user query. The algorithm, named SQE (Symbol-based Query Expansion), adds to the user query a set of words describing the meaning of each symbol appearing in the query. The hypothesis is that the semantic description of symbols which appear in text queries can be useful for enriching the query.

3 The Symbol-based Query Expansion Algorithm

The algorithm proposed consists of expanding only the queries including symbols. Therefore, the queries which do not include any symbol are not expanded. Moreover, only the symbols occurring in the queries are processed – words are not expanded. The main aim of the algorithm is to find the description of the semantics of the symbols occurring in the query by drawing information from a set of documents. The semantics description is then used to expand the original query. This way, the synonymous symbols are *indirectly* attached to the query symbols – the attachment is an indirect one because the query is expanded through semantics description rather that directly through symbols.

In designing a query expansion algorithm, one has basically to answer to some questions: (*i*) which queries and parts of a query should be expanded, (*ii*) which data are used to expand a query, (*iii*) the source of evidence used to draw expanding data, (*iv*) how the drawn data expand a query. As regards the first question, semantic descriptions are used in this research to expand a query. The *semantic description* of a symbol is a set of words which describes the meaning of the symbol. Therefore, the semantic description of a symbol could not be exactly the symbol definition provided, for example, by a biomedicine dictionary or thesaurus.

As regards the source of evidence, the algorithm to discover the semantic description of a symbol exploits a widely accepted paper writing rule: When a symbol is used in a scientific paper it is often introduced by a definition of it in the same paper or, if the symbol is already well known, in other papers. The hypothesis is, hence, that the words surrounding the symbol could be strongly semantically related to it. The analysis of the text surrounding the symbol provides useful evidence to select the words which describe the symbol semantics. The set R_i of retrieved documents with respect to the $i-$th query is the repository from which semantic description are built. Discovering the semantic description by using the information coded into R_i can be useful when the symbol can have more than one single meaning. The rationale hence of this algorithm is that R_i is the privileged document set which can provide useful evidence to implement a semantic description of the symbol in a consistent way with the symbol meaning intended by the user. The whole document collection will be considered and each symbol occurrence in the collection analyzed only if there is no evidence in R_i for extracting the semantic description. At indexing time, each keyword is recognized either as symbol or word according to a set of regular expressions reported in Table 1.

Regular expressions
`[a-zA-Z0-9]+[A-Z0-9]+[a-zA-Z]*[-/][a-zA-Z]*[0-9]+[a-zA-Z]*`
`[a-zA-Z][A-Z0-9]*'('[a-zA-Z0-9]+')'[a-zA-Z0-9]*`
`[a-zA-Z0-9]+[A-Z0-9]+[a-zA-Z]*`

Table 1. Set of Regular expressions for detecting symbols in text.

The set of regular expressions is designed to capture the more symbols as possible; as a consequence some words could be recognized as symbols. These expressions defined a symbol as a string containing upper- or lower-case letters, digits or some special characters. Some normalization was done to make special characters little influential. The characters `-()/` were removed from the strings matching one of the first two regular expressions, and when a string is recognized as symbol, all the characters forming the string is mapped into uppercase, to allow the matching among the same symbol written with different uppercase conventions. For instance the following strings are recognized as symbols and normalized to the same form:

$$\left.\begin{array}{l}\texttt{TTF-1}\\ \texttt{TTF1}\\ \texttt{Ttf-1}\end{array}\right\}\Longrightarrow \texttt{TTF1}.$$

Each symbol occurrence is then tagged by recognizing whether the symbol occurrence appears within a symbol definition or not. A symbol can be defined in the text centered around the symbol, or it can be simply used as a generic word of the vocabulary. Here the *symbol definition* is considered to occur when a couple of parenthesis appears in the text surrounding the symbol s_i. There are two possible patterns:

SDA (Symbol Definition After): $s_i\ (k_j\ k_h \ldots)$
SDF (Symbol Definition First): $\ldots k_j\ k_h\ (s_i)$

A *Symbol Usage* (SU) occurs when the symbol is used in the phrase just like a word of the common vocabulary, i.e., the author hypothesizes that the reader already knows the meaning of the symbol, and hence he may use it with no definition in the same phrase. A symbol usage pattern occurs when the symbol appears with no parentheses in the text surrounding the symbol itself. The pattern, then, is the following:

SU (Symbol Usage): $\ldots k_u\ k_j\ s_i\ k_l\ k_h \ldots$

Note that k_u is a generic keyword and it can be either word or symbol; hence the semantics of a symbol can be given by a set of words and symbols. Some examples of different symbol usages, taken from the narrative description of the topics for 2005 TREC Genomics Track, are reported in Table 2, where the symbol occurrences are marked in bold. The hypothesis underlying this symbol tagger is that when a symbol is used in a paper it is often introduced by a definition of it which follows one of the patterns labeled as SDA, or SDF, and hence individuating each symbol usage allows to discriminate which text windows have to be considered for discovering the semantic description of a symbol. Considering the patterns SDA and SDF as the symbol definitions, is not a novel hypothesis; it is already drawn in literature, see for example [4] and [16], while other authors prefer to consider only the pattern described by SDF, see [3]. Differently from the approaches of those articles, our research concentrated on extracting the set of words which are semantically associated to the symbol. At this aim, an algorithm was developed to exploit all the information which can appear in the text surrounding each occurrence of the symbol, even if the occurrence is not tagged as symbol definition.

ID	Phrase	Usage Type
106	...chromatin **IP** (Immuno Precipitations) to...	SDA
101	...when you do glutathione S-transferase (**GST**) ...	SDF
105	...purification of rat **IgM**.	SU

Table 2. Examples of symbol usage and their classification.

In particular, if the symbol occurrence follows the pattern labeled as SDF, then n keywords are extracted from the left text neighborhood of the symbol, while if the it follows the pattern labeled as SDA the n keywords are extracted from the right text neighborhood of the symbol. Finally, when the pattern followed is SU then n keywords are extracted both from the right and the left text neighborhoods. These keywords are added to a list of possible candidates. For each candidate keyword a score is calculated and then the l top ranked keywords are chosen as the semantic description of the symbol s_j.

For each symbol s_j in the i-th query, the algorithm extracts at most n keywords surrounding each occurrence of the symbol in the document set R_i. Each extracted keyword is inserted into a candidate list of semantic descriptors. This way, a set of words is associated to the symbol, i.e. the order is not considered. The words associated to the symbol to describe its semantics are sorted by the frequency of occurrence in the text windows. The more a word occurs in the text windows centered around a symbol, the more likely the word is used to describe symbol semantics. Since different but synonymous symbols are likely to share the same set of words, the algorithms aims at associating synonymous symbols together.

4 Experimental Settings

To evaluate the SQE algorithm in different situations, both 2004 and 2005 topic sets were used. Both are formed by 50 topics and are derived from interviews to real biologists, but while the first set of topics is free-form topics, the second set is stricter, and each topic fit within a particular template called Generic Topic Template (GTT). The templates are described in Table 3. Even if some 2004 topics can fit within a 2005 GTT, the two set are very different. The 2005 topic set has much less relevant documents than 2004 topic set.

Generic Topic Template	
GTT1	Find articles describing **standard methods or protocols** for doing some sort of experiment or procedure.
GTT2	Find articles describing the role of a **gene** involved in a given **disease**.
GTT3	Find articles describing the role of a **gene** in a specific **biological process**.
GTT4	Find articles describing interactions between two or more **genes** in the **function of an organ** or in a **disease**.
GTT5	Find articles describing one or more **mutations** of a given **gene** and its biological impact.

Table 3. GTTs for 2005 TREC topic set.

4.1 Equipment

The experiments were performed by using a server machine equipped with Fedora Core Linux Operating System, an Intel Xeon processor at 2.8 GHz and 2GB RAM. The prototype software framework was based on a relational DBMS which was used for storing and indexing text documents. A suite composed of Java modules (JDK 1.5.0) and C++ modules was implemented for enhancing the full-text indexing capabilities and the retrieval features of the DBMS, adapting them to our scope of symbol detection and symbol-based query expansion techniques. The DBMS chosen was MySQL AB version 4.0, which is one of the most popular open source relational database server which offers also full-text indexing and searching capabilities, based on a space-vector model [8].

4.2 Indexing

Before storing the documents into the document table in MySQL a conversion was performed. Only the document identifier, the title and the abstract of the MEDLINE documents were stored, while the other sections were not considered for these experiments. All the stop-words have been removed before storing the documents. Two different stop list were built starting from the SMART stop-list [13]. The first consists of of 599 stop-words while the second, a little bit larger, consists of 668 stop-words. The 69 stop-words added in the second step, were inserted into the stop list in capital letters because they appear frequently as tags in the document collection and are domain-specific words. They are removed from the documents only if they appear in capital letters. Testing this settings for 2005 topics, we have observed a consistent improvement in average precision (over than 22 percent) when the new stop list was used, and the experiments reported in this paper are performed with this larger stop list.

5 Experimental Results

Different runs were performed to test SQE. `NoSQE` was obtained without query expansion, and hence is the baseline, whereas `SQE` automatically expands the query with the set of words which represent the symbol semantic description discovered by the algorithm.

`SQE` was obtained by considering only the occurrences of the symbols which follows the SDF pattern. As a consequence, only the n keywords appearing in the *left* text neighborhood were selected as candidates for the symbol semantic description. The n parameter, which represents the length of the text window surrounding the symbol, was calculated by using the heuristics proposed in [9] and used also in [4]. This heuristics considers that a definition of a symbol should not have more than $\min(|A| + 5, |A| * 2)$ characters, where $|A|$ is the length of the string symbol. The number of keywords associated to each symbol of the query was defined by $\min(|A_a|, 2)$, where $|A_a|$ is the length of the string symbol without considering the digits. The number of documents to retrieve at the first step of retrieval was set to 1000 and the criterion for ranking the keywords in the list of possible candidates as symbol semantic description was based on the frequency of occurrence of the keywords into the text windows associated to each symbols.

Table 4 and 5 report the number of relevant documents which have been retrieved for 2004 and 2005 topic sets with respect to the total number of relevant documents in the collection. While for 2004 topic set the recall increases when the query expansion algorithm is applied, for 2005 topic set, the number of relevant documents even decreases, by thus showing an anomalous behavior with respect to the expected query expansion effectiveness. Table 6 and Table 7 report a brief summary of the effectiveness measures observed applying the SQE algorithm. The experiment was carried out testing the algorithm both for 2004 and 2005 topic sets.

Total Relevant	8268	–
RunID	Rel-Retr	Recall
NoSQE	3958	0.4787
SQE	4003	0.4841

Table 4. Recall for the SQE and 2004 topic set.

Total Relevant	4584	–
RunID	Rel-Retr	Recall
NoSQE	3088	0.6736
SQE	3012	0.6571

Table 5. Recall for the SQE and 2005 topic set.

RunID	A-P	R-P	P@5	P@10	P@100
NoSQE	0.2585	0.3013	0.5920	0.5300	0.3116
SQE	0.2670	0.3044	0.5680	0.5240	0.3114

Table 6. Effectiveness measures for SQE and 2004 topic set.

RunID	A-P	R-P	P@5	P@10	P@100
NoSQE	0.1937	0.2213	0.4327	0.3571	0.2006
SQE	0.1646	0.1928	0.3918	0.3224	0.1904

Table 7. Effectiveness measures for SQE and 2005 topic set.

There is no a clear improvement in terms of effectiveness measures for 2004, while for 2005 a consistent loss of effectiveness was observed. Figure 1 and 2 show the difference between the Average Precision computed for SQE and NoSQE runs for 2004 and 2005 topics set respectively. Figure 1 shows that the number of 2004 topics for which the query expansion improves effectiveness is greater than the number of topics for which effectiveness decrease, while Figure 2 shows clearly that the number of 2005 topics for which effectiveness decreases when the symbol semantic description discovered is added to the query is greater than the number for which effectiveness increase.

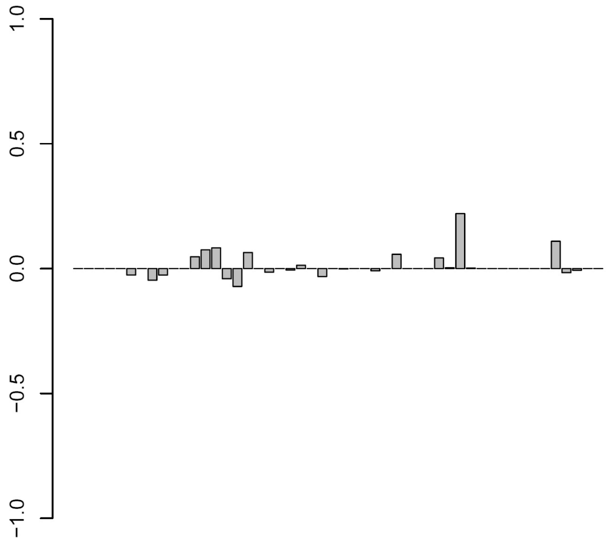

Fig. 1. Comparison between SQE and NoSQE average precision for 2004 topics.

The results were inconclusive because the algorithm behaves differently with the topic sets. QE apparently depends on the topic set, and, in the particular instance of the Genomics Track, it depends on the task since the 2005 topics refer to a quite different task from the ad-hoc task

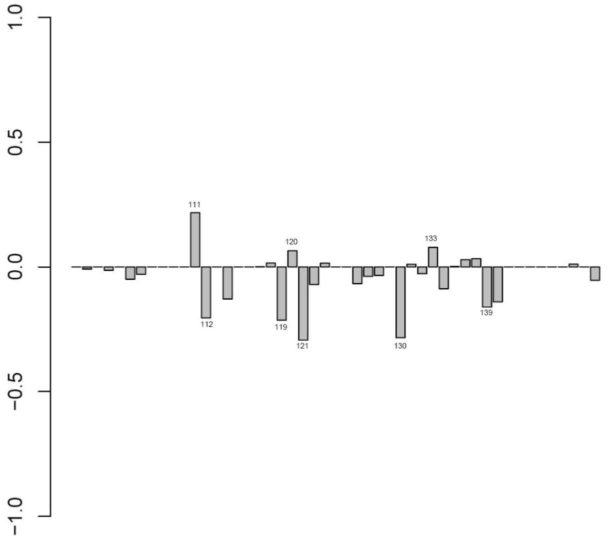

Fig. 2. Comparison between `SQE` and `NoSQE` Average Precision for 2005 topics.

proposed in 2004. There might be two other reasons why SQE failed. One reason might be an imprecise computation of symbol semantic description, i.e. a wrong association between symbols and symbol semantic description. A per-topic analysis was performed for 2005 topic set to test the hypothesis that the drop of effectiveness is imputable to a wrong association between symbols and symbol semantic description.

5.1 Per-topic analysis

There are three topics, i.e. `111`, `120`, `133` for which the use of query expanded by the discovered symbol semantic description improves consistently system effectiveness. Table 8 shows the keywords added to the original query. Table 9 reports the topics which show a consistent negative impact on effectiveness, and the keywords added to the text queries for expanding symbol semantics. MeSH thesaurus (`http://www.nlm.nih.gov/mesh/meshhome.html`) and two different on-line databases, i.e. The Stanford Biomedical Abbreviation Server (`http://bionlp.stanford.edu/`) and AcroMed (`http://medstract.med.tufts.edu/acro1.1/`) were used to test the words that appear in the symbol semantic description and that are semantically related to the original symbol.

Almost all the keywords added to the queries are semantically related to the symbol to which they are associated, but that a right association

TopicID	symbol	keywords added
111	PRNP	prion prp codon disease
120	NM23	metastasis
133	A2	a1
	PLA2	a2 phospholipase
	SAR1	cerevisiae replacing characterised
	ER	reticulum endoplasmic

Table 8. Topics with a positive impact on effectiveness.

TopicID	symbol	keywords added
112	IDE	insulin degrading enzyme
119	GSTM1	glutathione transferase m1 mu
121	BARD1	brca1 domain ring cancer
	BRCA1	cancer breast susceptibility mutations
130	BRCA1	cancer breast susceptibility mutations
139	GDNF	factor neurotrophic derived glial

Table 9. Topics with a negative impact on effectiveness.

is not sufficient for improving the retrieval effectiveness. Among the 49 TREC 2005 topics considered, four topics present symbols with a wrong semantic expansion, one of them is the topic 133 which presents an improvement in term of effectiveness.

5.2 GTTs distribution

A query classification was performed to have a measure of the degree to which the effectiveness of SQE is related to the type of template within the single query. Figure 3 shows the variability in terms of increment or decrement of the Average Precision within each class of GTTs. The first and the fifth topic templates give a lower variability, as reported also in Table 10.

Generic Topic Template	Variance
GTT1	0.00028
GTT2	0.01578
GTT3	0.00946
GTT4	0.01313
GTT5	0.00214

Table 10. Increment/decrement variability within each class of GTT.

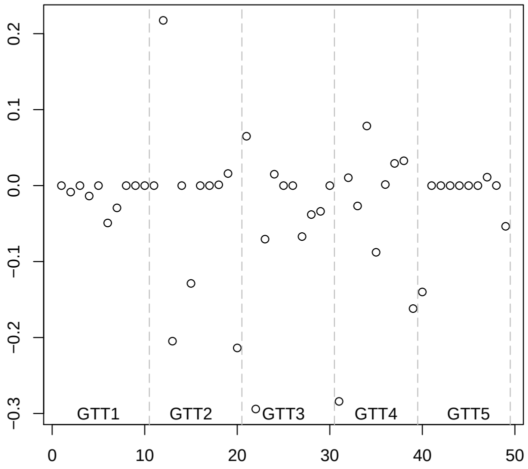

Fig. 3. Increment/decrement of AP within each class of GTT.

6 Statistical Analysis

Another reason why SQE performed poorly might trace back to the level of precision after R_i was retrieved. Indeed, it might be that a few retrieved relevant documents provide little useful evidence to compute useful symbol semantic description. A correlation analysis was then performed between the level of precision after R_i was retrieved and the variation of precision after SQE was performed. For clarity, let us call Δ_{AP} the increment or decrement of Average Precision, with respect to baseline, when the SQE algorithm is applied. First of all the correlation between Average Precision for the baseline and Δ_{AP} is computed to measure the degree of dependency between SQE effectiveness and the baseline retrieval results.

The Pearson correlation computed for Average Precision on 2004 topic set is 0.0933, while for 2005 is -0.40. Figures 4 and 5 show the relation between the baseline and the variations in effectiveness when the query expansion algorithm is applied. Once again the behavior is very different from year to year: It seems that there is no correlation between the starting level of effectiveness and the increment or decrement in terms of effectiveness for the 2004 topic set, whereas a little negative correlation is reported for the 2005 topic set by thus suggesting that SQE improves effectiveness if the starting level of effectiveness is low, while SQE produces a deterioration of effectiveness but if the starting level of precision is higher.

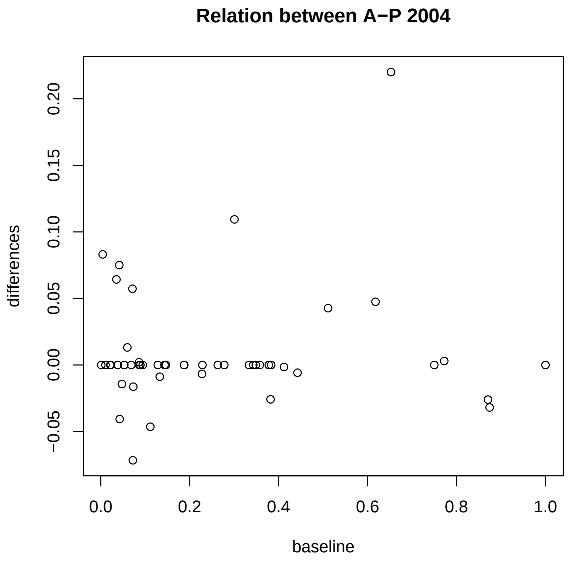

Fig. 4. Correlation between A-P computed for baseline and the differences when SQE is applied for 2004 topics.

A statistical test based on Fisher-Snedecor F test, was applied to measure the statistical significance of the correlation found. The null hypothesis $H_0 : \rho = 0$ is tested versus the alternative hypothesis $H_1 : \rho \neq 0$. The test is given by:
$F = \frac{r^2(n-2)}{1-r^2}$ which follows a $F_{(1,n-2)}$ distribution, under H_0. The results are reported in Table 11.

Topic set	r	F	p-value	Decision
2004	0.0933	0.4126	0.95	H_0 not rejected
2005	-0.3972	8.8210	0.00935	H_0 rejected

Table 11. Summary of correlation significance test.

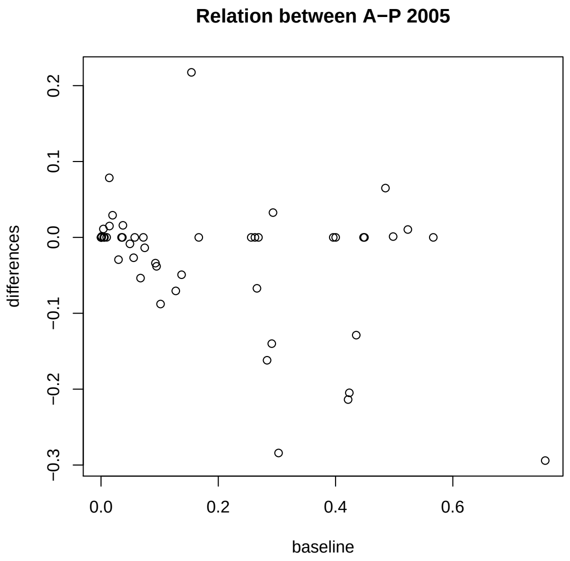

Fig. 5. Correlation between A-P computed for baseline and the differences when SQE is applied for 2005 topics.

Fisher-Snedecor test suggests that these results are significant for 2005 (significance level of 0.05); similar results are reported in literature [7], but in that case the authors observed an improvement of effectiveness when the relevance feedback was applied, confirming a more usual behavior of an implicit relevance feedback query expansion.

7 Considerations

The experimental results have been quite surprising since SQE has not been as much successful as the classical query expansion algorithms tested in diverse domains. Other experiments carried out during the TREC 2005 evaluation campaign and reported in working notes have observed a similar curious behavior also for more traditional query expansion algorithms based on implicit relevance feedback [1], [2], [17]. In particular, in [1] a domain specific query expansion algorithm was built and compared with the classic Rocchio's scheme, and a drop of effectiveness with respect to the unexpanded query baseline was obtained both for the new algorithm and the classic relevance feedback query expansion algorithm.

The surprising results reported in this and other papers stimulates some reflections on the deployment of standard techniques to the domain of IR in Genomics. First, the use of symbols deviates from the ordinary word usages and adhere to some practices being peculiar in the Genomics domain. Second, the search tasks of that domain is radically different from common ad-hoc tasks; it seems more related to discovery than to relevance-driven tasks. Finally, experiments suggest an inverse relationship between retrieval and SQE effectiveness; if one leverages poorly performing queries, significant improvements may be obtained. These and other reflections help shape the future research in this field; in particular the mechanics of query expansion should be investigated since a detailed analysis of the successes and the failures of SQE would provide useful insights.

References

1. S. Abdou, P. Ruck and J. Savoy. Evaluation of Stemming, Query Expansion and Manual Indexing Approaches for the Genomic Task. In *The Fourtheenth Text REtrieval Conference Proceedings (TREC 2005)* `http://trec.nist.gov/act_part/t14_notebook/notebook.papers/uhospital-geneva.geo.pdf`.
2. F. Camous, S. Blott, C. Gurrin, G. Jones and A. Smeaton. Structural Term Extraction for Expansion of Template-Based Genomic Queries. In *The Fourtheen Text REtrieval Conference Proceedings (TREC 2005)* `http://trec.nist.gov/act_part/t14_notebook/notebook.papers/dublincu.geo.pdf`.
3. J.T. Chang, H. Schuntze and R.B. Altman. Creating an Online Dictionary of Abbreviations from MEDLINE. *Journal of American Medical Information Association*, 9(6): pages 612–620, 2002.
4. M.A.Hearst and A.S. Schwartz. A simple algorithm for identifying abbreviation definitions in biomedical text. *Proceedings of the Pacific Symposium on Biocomputing*, 2003.
5. W. Hersh, R. Bhuptiraju, L. Ross, P. Johnson, A. Cohen, and D. Kraemer. TREC 2004 Genomics Track Overview. In *The Thirteenth Text REtrieval Conference Proceedings (TREC 2004)*, NIST Special Publication: SP 500-261, 2005.
6. J. Kekalainen and K. Jarvelin. The Impact of Query Structure and Query Expansion on Retrieval Performance. *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Melbourne, Australia, 1998.
7. M. Mitra, A. Singhal and C. Buckley. Improving Automatic Query Expansion. *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Melbourne, Australia, 1998.
8. MySQL AB. MySQL Homepage. `http://www.mysql.com`. Visited on October 2005.
9. Y. Park and R.J. Byrd. Hybrid Text Mining for Finding Abbreviations and Their Definitions. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, PA, June 2001: pages 126–133.

10. G. Salton and C. Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal of American Society for Information Science*, 411(4): pages 288–292, 1990.
11. P. Srinivasan. Query Expansion and MEDLINE. *Information Processing and Management*, 32(4): pages 431–443, 1996.
12. A. Spink. Term Relevance Feedback and Query Expansion: Relation to Design. *recuperare riferimento corretto*, pages 81–90.
13. The English Stoplist of the SMART System. `ftp://ftp.cs.cornell.edu/pub/smart/english.stop`. Visited on October, 2005.
14. C. Buckley *et al.* The `trec_eval` Evaluation Package. `http://trec.nist.gov/trec_eval/trec_eval.7.3.tar.gz`, 2005. Visited on October, 2005.
15. J.D. Wren, J.T. Chang, J. Pustejovsky, E. Adar, H.R. Garner and R.B. Altman. Biomedical Term Mapping Databases. *Nucleic Acids Research*, 33, Database Issue: pages 289–293, 2005.
16. H. Yu, G. Hripcsak and C. Friedman. Mapping Abbreviations to Full Forms in Biomedical Articles. *Journal of American Medical Information Association*, 9(3): pages 262–272, 2002.
17. N. Yu, Yang Lingpeng, J.Donghong, Z. Jie. TREC 2005 Genomics Track at I2R. In *The Fourtheenth Text REtrieval Conference Proceedings (TREC 2005)*, `http://trec.nist.gov/act_part/t14_notebook/notebook.papers/inst-infocomm.geo.pdf`.