An Empirical Study of Lexical Variation Methods for Biomedical Information Retrieval

Yu-Chun Wang[ab], Hsieh-Chuan Hung[ac], Chia-Wei Wu[a], Tzong-Han Tsai[ac], Wen-Lian Hsu[a]

[a]Institute of Information Science, Academia Sinica, Nankang, Taipei, 115, Taiwan
[b]Department of Computer Science, National Chengchi University, Taipei, 116, Taiwan
[c]Department of Computer Science and Information Engineering ,
National Taiwan University, Taipei, 106, Taiwan

Abstract

The rapid growth in the volume of biomedical literature has made it increasingly difficult to find precise information. To implement an accurate and fast biomedical information retrieval (IR) system domain, we must deal with the variants of biomedical terms carefully. In this paper, we focus on the lexical variants of query terms and propose eight lexical variation rules. Based on the biological IR tasks defined for the Genomic Track, we design a series of experiments to examine the effects of these eight rules and combinations thereof. We evaluate the rules by three indicators: MAP, recall, and query term frequency. Our experiment results show that varying hyphenation significantly improves the performance of information retrieval. In addition, the variation rules for Greek transcriptions also increase IR performance when a query contains terms with such transcriptions. However, the rules that generate general variants tend to undermine MAP scores.

Keywords: Biomedical literature, information retrieval, lexical variation, query expansion.

1 Introduction

Advances in biotechnology have given rise to a vast amount of biomedical data, most of which is now available to the scientific community in electronic format. According to Cohen and Hunter [1], more than 1,500 new papers are added to Medline every day. However, the rapid growth in the literature has made it increasingly difficult to locate the accurate information expeditiously. Clearly, if biomedical experts are to experience the full benefits of electronically accessible literature, natural language processing (NLP) applications (such as information retrieval, information extraction, etc.) are a necessity to facilitate navigation through the volumes of biomedical texts.

Information retrieval identifies and extracts documents that are relevant to a user's query from a large database. Naturally, the task has to be performed accurately and efficiently. Most approaches, such as the famous vector space model [2], score the degree of match between the terms in a query and the related terms in a document.

Unlike information retrieval in general domains, biomedical IR systems suffer from low recall, because biomedical terms usually have many aliases, abbreviations, acronyms, and synonyms. There are various ways, called lexical variants, to present the same term. For example, the protein "NF-kappa B" has the fol-

lowing lexical variants: “NF-kappaB”, “NFkappa B”, “NF-kB”, and “NFkB”. In the biomedical domain, there are many lexical resources and databases, such as UMLS [3] and LocusLink [4], which provide a large number of aliases and alternative gene symbols. However, they do not cover all the variants for the biomedical terms in Medline abstracts, which already number over ten million and are increasing rapidly.

Several methods have been proposed for generating lexical variants for query term expansion. Divita et al. [5] used a large knowledge base (the SPECIALIST Lexicon [7]) to manage inflectional morphology. More recently, Tsuruoka and Tsujii [6] proposed an automatic learning method for lexical variant generation. However, both approaches have some disadvantages. The first is based on a set of rules in [10] that both cooperate with and depend on the SPECIALIST Lexicon, which contains over 100,000 entries, and cannot be covered completely. The second approach must still be trained on a well-annotated corpus.

In this paper, we propose a simple and efficient method of lexical variation for query expansion in biomedical information retrieval. Based on the lexical variation method used by Büttcher et al. [7], our method includes eight new rules. To evaluate the effectiveness of these lexical variation rules for biomedical information retrieval, we implement a biomedical information retrieval system using a very large corpus that is composed of Medline abstracts published from 1994 to 2003. We adopt the queries provided by the Genomic Track 2004 for testing, and observe the frequency of the original query terms and their lexical variants to evaluate the eight rules.

The remainder of this paper is organized as follows. In Section 2, we describe our lexical variation method and its application to an IR system, and in Section 3 we deliberate over our experiments, including dataset, settings, and results. The impact of expanded terms on the search results is discussed in Section 4. We wrap up the paper with a brief conclusion and future work.

2 Methods

In this section, we introduce our lexical variation method and describe how its application to an IR system.

1. Information Retrieval System

Figure 1 presents an overview of our IR system for retrieving biomedical documents. It is comprised of three stages. The first stage is document indexing, which stores all documents in an index file and transforms each document into a word list. The file connects the query terms to the words in documents. In addition, we remove all stop words, perform stemming on words, and convert words to lowercase.

The second stage is query preprocessing in which we also remove stop words and perform stemming on words that do not trigger lexical variation rules. To guarantee that preprocessed query terms and words in the index file can be mapped correctly, all word processing methods in this stage should be consistent with those in the document indexing stage. Unlike stemming or lowercase conversion which changes the original terms, lexical variation rules merely expand terms from the original terms. Thus, we do not have to apply lexical variation rules in both stages. In most cases, the rules are applied in query preprocessing, rather than document indexing, to avoid generating too many terms.

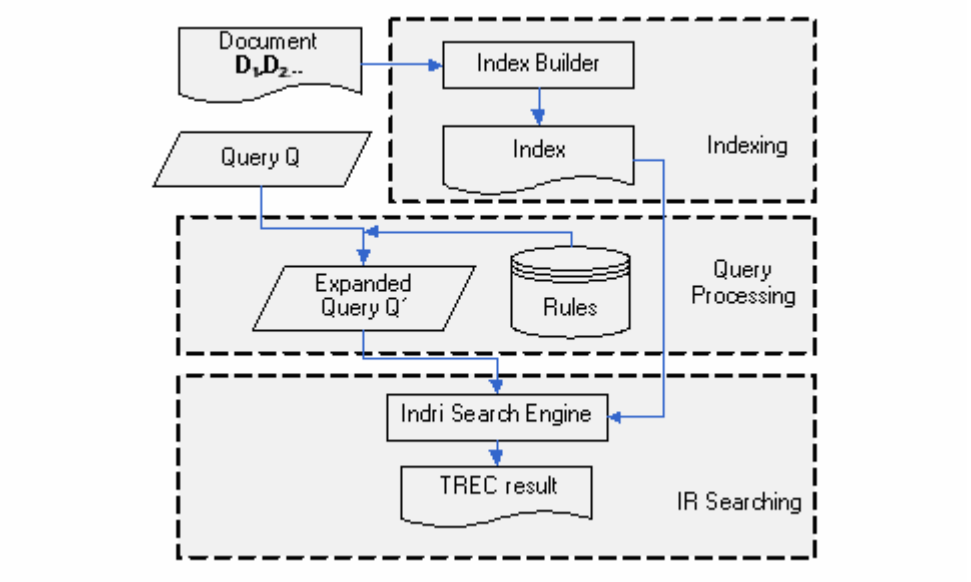

Figure 1 Our IR System

In the final stage, we find the relevant documents by a scoring algorithm, such as TFIDF, OKAPI, or boosting. Discussion of scoring algorithms is beyond the scope of this paper; however, a detailed assessment can be found in [7].

2.2 Lexical Variants

In general, biomedical named entities (NEs), such as protein or gene names, do not follow any particular nomenclature [8] and can comprise long compound words and short abbreviations [9]. Some NEs also contain various symbols and other spelling variations [10]. On average, biomedical name entities have five synonyms, most of which are generated by lexical variations. These variations, generated by different hyphenation/spacing rules and Greek-letter representations, make biomedical information retrieval a challenging problem indeed. If they are not dealt with carefully, many relevant documents containing a term that is a variation of the query term may be missed. To overcome this problem, we have, therefore, devised several rules for obtaining the lexical variants of biomedical terms.

Rules

Extending the lexical variation algorithms described in [7], we propose eight lexical variation rules for obtaining the variants of biomedical terms in Medline abstracts. These rules, which we apply to all query terms, are designed specifically for biomedical terms, and do not affect the standard English definition of words in those terms. The rules are listed below.

- Insert a hyphen at every transition between Latin letters, Arabic numerals and Greek letters.
- Convert the last Arabic numeral to a Latin letter. For example, “Smad-4” would become “Smad-D”.
- Convert the last Latin letter to an Arabic numeral. For example, “NF*kappa* B” would become “NF*kappa* 2”.
- Convert the last Arabic numeral to a Roman numeral. For instance, “Smad-4” would become “Smad-IV”.
- Replace Greek transcriptions with their corresponding Latin letters. For example, “alpha” becomes “a”, “gamma” becomes “g”, and so on.

Replace a hyphen with a space.

- Remove hyphens from terms. However, if the characters preceding and following a hyphen are all letters or all numerals, the hyphen should not be removed.

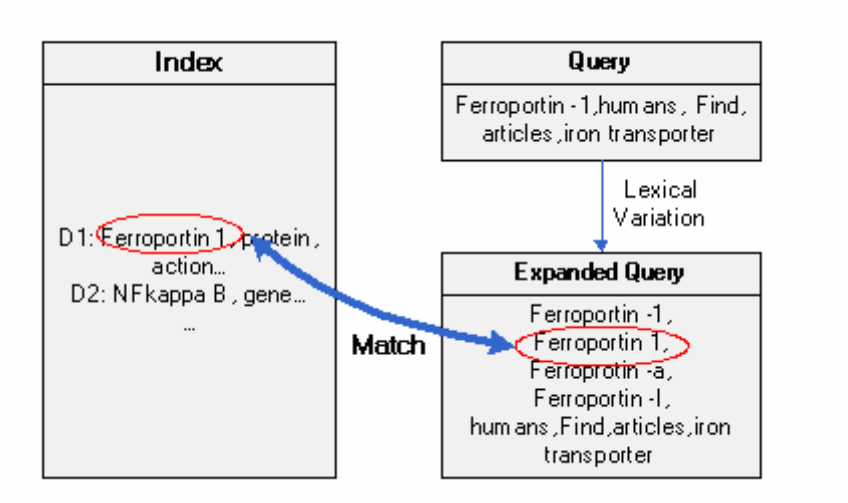

Figure 2 An example of matching the variant with the index

- Remove hyphens and numbers at the end of the term.

After lexical variation, all terms generated by the rules are added to the original query. We also check if the length of each variant is greater than one. If the length is one, it should be omitted. For instance, after applying Rule 8 to the term "p-53", it becomes "p". Obviously, "p" is a harmful term for searching.

2.2.2 Rule Combinations

The order of the rules is important, as different sequences may produce different lexical variants. Some rules may compensate for the effect of precedence rules. The rules and their permutations used in this experiment are Rule 1 & 2, Rule 1 & 2 & 3, Rule 5 & 6 & 7, and so on.

2.3 The query preprocessing stage

We use the following query to explain this stage:

"Ferroportin-1 in humans Find articles about Ferroportin-1 an iron transporter in humans."

After parsing and removing stop words, it becomes:

"Ferrportin-1|humans|Find|articles|Ferroportin-1|iron transporter|humans."

Since we use the Indri Query Language [11] as our IR engine, these terms are used to generate an Indri query. Note that "#combine" and "#1" are used as functional words in Indri.

#combine (Ferroportin-1 humans Find articles #1 (iron transporter))

Next, we apply our lexical variation method to the query, which gives us:

#combine (#syn (Ferroportin-1 Ferroportin1 Ferroportin-I Ferroportin-a) humans Find articles #od1(iron transporter))

We put the original terms and their lexical variants into a "#syn" block, which means that bracketed words are treated as synonyms.

Figure 2 shows how the lexical variation rule works. Suppose D1 is a document relevant to the query. It can not be retrieved correctly without applying lexical variation rules to expand the query term from **Ferroportin-1** to **Ferroportin1.**

2.4 Dataset

We use the MEDLINE bibliographic database to evaluate the proposed method. The subset of MEDLINE used by Genomic Track in the Text Retrieval Conference (TREC) 2004 [12] includes 4,591,008 abstracts of research papers from 1994 to 2003. For evaluation purposes, Genomic Track also provides the test data, queries, and relevant documents corresponding to each query. There are 50 queries and

Table 1. The difference in the MAP of each query for some special combinations of rules

Combination of Rules*	Examples For Original Term	Examples For Variants	MAP Before Applying Rule	MAP After Applying Rule
1 + 2 (1)	RSK-2 (57)	RSK-B (9)	0.3520	0.3570
1 + 2 + 6	Ferroportin 1 (15)	Ferroportin (50)	0.1731	0.3628
(1 + 6) **	Bfa 1 (0)	Bfa (1259)	0.7180	0.6872
1 + 6 (1)	WD-40 (149)	WD 40 (2)	0.6279	0.6305
2 + 6 (1) **	Ferroportin 1 (15)	Ferroportin (50)	0.1731	0.3628
5 + 7 (7)	TGF-b (10)	TGFb (2)	0.0396	0.0422

* The numbers in parentheses are the combinations of the rules compared.

** See the end of Discussions for explanation of the behavior of this rule.

8,286 relevant documents in the test set.

3 Experiments

To compare the effectiveness of the eight lexical variation rules and their combinations described in Sections 2.3.1 and 2.3.2, we built an information retrieval system and an IR engine based on the Indri search engine. To evaluate our method, we used fifty queries from 2004 Genomic Track for testing and measured the performance of the search results.

3.1 Experimental Design

We use three indicators to evaluate our method, namely, the mean average precision (MAP), recall, and term frequency.

First, we use the TREC Evaluator [13], which provides MAP values, to evaluate the performance of our system. MAP is widely used for evaluating the performance of IR and QA systems, and can also be used to assess the effectiveness of lexical variation rules.

MAP is affected by the ranking of relevant articles. If the number of relevant articles is small, the MAP will be high if the most relevant articles are ranked high. In contrast, the MAP will be low if most relevant articles have lower rankings, even if the number of relevant articles is large.

However, for some applications, the volume of relevant articles is important. Therefore, we have to find another evaluation measure, instead of ranking.

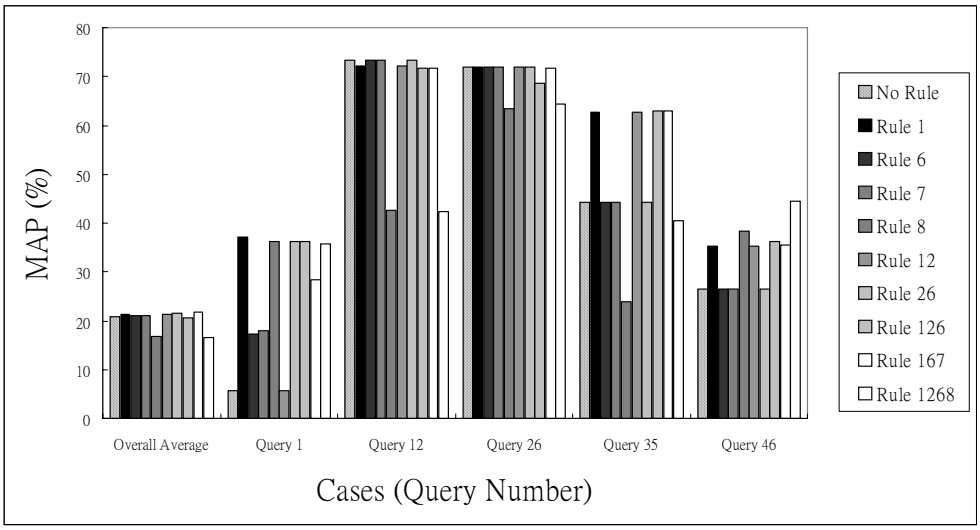

Figure 3 MAP chart for every rule

Table 2. The effects of some special combinations

Combinations of Rules	Overall Average MAP	Examples For Original Term	Examples For Variants
1 + 2	0.2136	RSK2	RSK-2, RSK-B
1 + 2 + 6	0.2048	FancD2	FancD-2, FancD b
2 + 6	0.2144	Ferroportin-1	Ferroportin-a,Ferroportin 1
1 + 6 + 7	0.2188 (best)	Gal1	Gal-1, Gal 1
1 + 2 + 6 + 8	0.1648 (worst)	Smad4	Smad-4, Smad d, Smad

Second, we use the difference in the relevance of the articles found by two rules as another measure. Although using two rules may retrieve an equal number of relevant articles, the articles found using these two articles may actually different. Thus, the difference between retrieved relevant articles is a good measure that gives us detailed information about the search results. The third evaluation measure is the frequency of expanded query terms. Only relevant documents are used to count the frequency, which is an intuitive way to observe the effectiveness of lexical variation methods. If our method can produce additional relevant query terms, the IR system should have better recall. For example, Ferroportin1 is the expanded query term of the original query Ferroportin-1. We believe that the frequency of Ferroportin1 in the index file is a good measure for evaluating the rule that generates Ferroportin1.

3.2 Experimental Results

After conducting the experiments on all the possible combinations of every rule, we get $2^8 = 256$ results. We now use the three indicators mentioned in Section 3.1 to evaluate our lexical variation algorithm.

3.2.1 MAP

In this subsection, we consider the overall average MAP, and then investigate the MAP of each query separately. The overall average MAP of each configuration is shown in Figure 2. The overall average is the average MAP of fifty queries.

The MAP of fifty queries with "no rule" is 20.82, but with the combination of Rules 1, 6, and 7, it is 21.88. Thus, the combination of these three rules improves performance, while Rule 8 degrades it. The remaining rules are nearly neutral./do not affect the performance significantly.

Note that although Rule 2 does not affect MAP and Rule 1 and Rule 6 are both beneficial, combining them degrades performance. Table 2 shows the effects of some special combinations.

3.2.2 Recall

We compare the difference of each combination of the rules by using the following approach:

$$newfound = Rule_i \times Rule_j - Rule_i \cup Rule_j$$

$$missed = Rule_i \cup Rule_j - Rule_i \times Rule_j$$

where $Rule_i \times Rule_j$ means applying both.

We chose the number of relevant documents as one of the indicators of our method, and the total numbers of newfound and missed documents in each configuration. Roughly speaking, albeit some relevant might lost, the rules or their combination can achieve better recall.

In Figure 4 we show the ratio of the variance of MAP to the variance of relevant documents we retrieved. As shown in this figure, for most of the rules the MAP and the relevant documents we retrieved are in positive correlation. But there are still exceptions, such as the configurations containing Rule 8, give negative ratios.

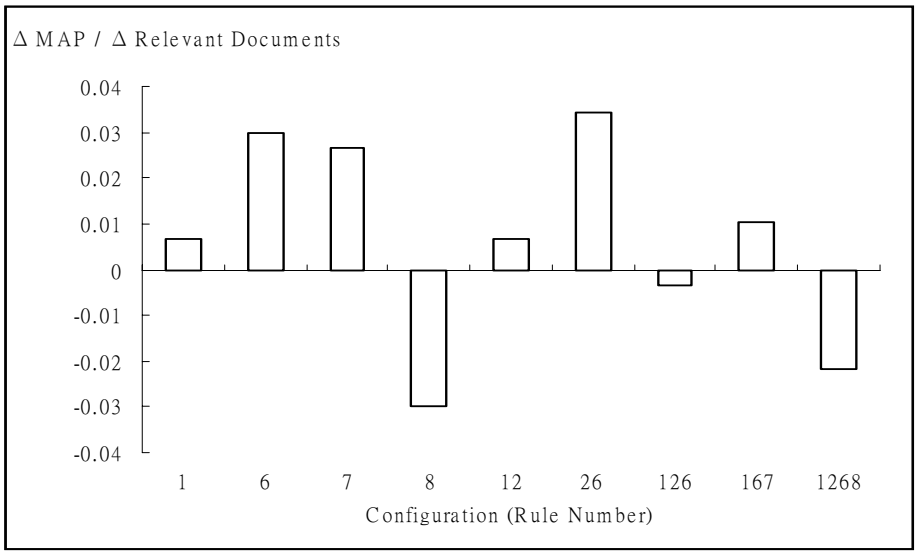

Figure 4 The relationship between relevant documents and MAP

By these ratios the relative effect of ranking of each rule can be evaluated. In general, finding more relevant documents raises MAP, leading to a positive ratio. For those cases with negative ratios, we can refer these exceptions to incorrect ranking. Rule 8 produces generalized variants, which retrieves more relevant document but also ruined the ranking and undermines MAP as well.

3.2.3 The frequency of expanded query terms

In this section, we discuss how a query's change caused by a lexical variation rule influences that query's retrieval result, which is usually measured by MAP. Given a query q and a lexical variation rule r is applied on q, we define the relative frequency as follows:

$$\text{Relative Freq}(q,r) = \frac{Freq_{lexi}}{Freq_{org} + Freq_{lexi}} \quad ,$$

where $Freq_{org}$ denotes the frequency of q's most variant term, t, before applying r, while $Freq_{lexi}$ denotes t's term frequency after applying r. Here, the most variant term means the term in q which has the highest frequency variance before and after applying r. To measure the variance of MAP, we define the marginal MAP as follows:

$$\text{MarginalMAP}(q,r) = \frac{MAP_{lexi} - MAP_{org}}{MAP_{org}} \text{, where}$$

MAP_{org} denotes the MAP of q's retrieval result before applying r, and MAP_{lexi} denotes the MAP of q's retrieval result after applying r.

Table 3. The impact of different rules on the lexical variants.*

Configuration	Ex. of Original Term **	Ex. of Variants **	MAP Before Applying Rule	MAP After Applying Rule	New Docs	Missed Docs
Rule 1	WD40 (224)	WD-40 (149)	0. 4416	0. 6279	62	1
	RSK2 (345)	RSK-2 (57)	0.2659	0.3520	18	0
Rule 6	Ferroportin-1 (1)	Ferroportin 1 (35)	0.0560	0.1731	3	0
	Single-strand (1740)	Single strand (299)	0.0115	0.0141	8	3
Rule 7	Ferroportin-1 (1)	Ferroportin1 (35)	:0.0560	0.1806	7	0
	TGF-beta (3908)	TGFbeta (328)	:0.0058	0.0396	3	0
	TGF-beta(7803)	TGFbeta (880)	0.0001	0.0005	2	0
Rule 8	Ferroportin-1 (1)	Ferroportin (50)	0.0560	0.3628	13	0
	FancD2 (134)	FancD (9)	0.7329	0.4253	3	14
	Smad4 (1739)	Smad (3347)	0.7192	0.6334	1	0
	WD40 (224)	WD (2115)	0.4416	0.2380	52	1

* Some rules are omitted cause the change is not conspicuous.

** The number in the parentheses stand for the term frequency.

In Figure 5, we use the *x*-axis to represent the relative frequency and *y*-axis to represent the marginal MAP. Each point represents a query-rule pair. In addition, we use different symbols to represent different rules. For example, the points represented by △ means that we apply Rule 1 on these queries and get lexical variants. We observe that, in Rule 1, 6, and 7, the correlation between relative frequency and marginal MAP is positive. It means the more frequent terms we expanded, the higher MAP we can achieve. However, we also notice that, in Rule 8, the correlation between relative frequency and MAP is negative.

Relative frequency of rules is another way to measure whether the rule is too aggressive. The relative frequency made by Rule 1, 6, 7 are less than 0.5, which means the frequencies of lexical variants are less than the frequency of the original term, while the relative frequency made by Rule 8 is close to 1, which means the frequencies of lexical variants are much more than the frequency of the original term. This phenomenon explains why Rule 8 causes MAP decrease.

3.3 Discussion

To analyze the impact of expanded terms on the search results, we categorize the relationship between the expanded term and the original query term as one of two types:

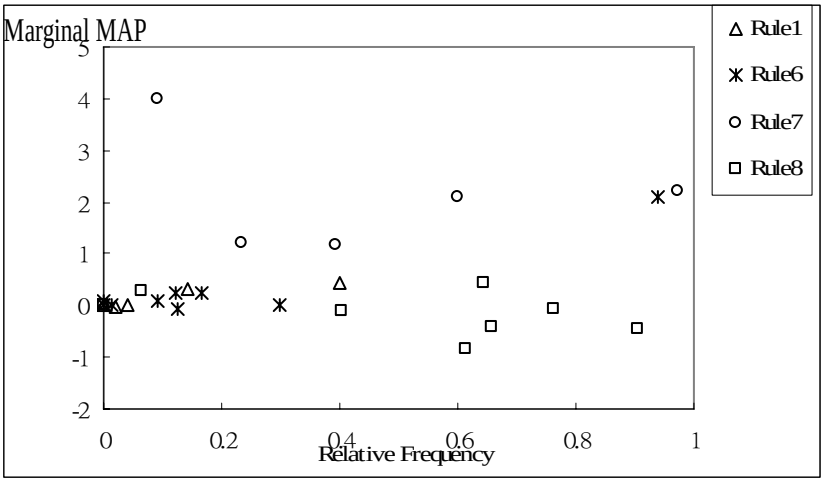

Figure 5 The relationship between relative frequency and marginal MAP

1) The expanded term has a higher-level concept meaning than the original term

In this case, the expanded query term may be less specific than the original term. Thus, the MAP may improve or deteriorate, depending on the semantic meaning of the original query. For instance, adding the term “Ferroportin” (the original term is Ferroportin-1) to query 1 increases the MAP by 31%. However, adding “Samd” (the original term is Samd-4) to query 12 reduces the MAP by 30%. Generally, in this case the precision decreases, but the recall increases.

2) The expanded term has a different meaning to the original term

In this case, the search performance always deteriorates, because the semantics of the original query are corrupted. The accuracy of the expanded term then depends on that of other terms in the query. However, the performance will not decrease significantly if the domains of the generated variations do not overlap too much with the domain of the original term. For instance, applying Rule 4 and 6 to query 26 generates the variant BFA I, which is different to the original term, BFA1. This reduces the MAP by 0.01%.

4 Conclusion

In this paper, we conduct a series of experiments to investigate which lexical variation rule can help the retrieval of biomedical abstracts most. In traditional experiments, most researchers only use MAP or term frequency to evaluate lexical variation rules’ effectiveness. We use MAP, recall, and term frequency to analyze the effectiveness of each rule. In addition, we further define two indicators: relative frequency and marginal MAP to measure how much a query’s change caused by a lexical variation rule influences that query’s retrieval result, which is usually measured by MAP. According to our observation, if the correlation between these two indicators is positive, the rule is useful to generate good lexical variants which improve performance.

According to our evaluation on all lexical variation rules, we have found that the rules for dealing with hyphens improve the precision of information retrieval the most, which implies that biologists often insert hyphens in different positions. Also, the rule for manipulating Greek transcriptions substantially improves the MAP and recall of query terms with such transcriptions. Converting Greek transcriptions in query terms to Latin letters and applying the hyphen variation rules improve both the MAP and recall rate of biomedical information retrieval.

5 Future Work

In the future, we will compare our lexical variation rules with other lexical variation systems, for example, NLM’s LVG (Lexical Variant Generator) [14]. The LVG can generate inflectional lexical variants such as “acting” and “acted” for the term “act”. When incorporated into information retrieval engines, some engines do not use stemming on their term indexes and query terms. Therefore, when using these engines, we need to generate the inflectional variants of query terms. On the other hand, as mentioned in Section 2.1, our methods incorporate the Indri Query Language, which does stem query terms and term indexes. We will compare the MAP scores of our method, which uses stemming, with those of LVG, which do not use stemming.

We will also test our method on more queries to obtain more cases of lexical variants from new query terms. This would enhance our comprehension of the properties of lexical variations and provide more precise evaluation.

6 Acknowledgement

We thank Dr. Chao-Lin Liu for his useful comments.

We are grateful for the support of National Science Council under GRANT NSC94-2752-E-001-001.

7 References

[1] K. B. Cohen and L. Hunter, "Natural Language Processing and Systems Biology," in *Artificial Intelligence and Systems Biology, Springer Series on Computational Biology*, W. Dubitzky and F. Azuaje, Eds.: Springer, 2005.

[2] G. Salton, Introduction to Modern Information Retrieval: McGraw-Hill, 1983.

[3] C. Lindberg, "The Unified Medical Language System (UMLS) of the National Library of Medicine," *J. Am. Med. Rec. Assoc.*, pp. 40-42, 1990.

[4] D. Maglott, "Locuslink: a directory of genes," in *NCBI Handbook*, 2002, pp. 19-1 to 19-16.

[5] G. Divita, A. C. Browne, and T. C. Rindflesch, "Evaluating Lexical Variant Generation to Improve Information Retrieval," presented at American Medical Informatics Association Annual Symposium 1998, 1998.

[6] Y. Tsuruoka and J. c. Tsujii, "Probabilistic Term Variant Generator for Biomedical Terms," presented at Special Interest Group on Information Retrieval 2003, 2003.

[7] S. Büttcher, C. L. A. Clarke, and G. V. Cormack, "Domain-Specific Synonym Expansion and Validation for Biomedical In-formation Retrieval (MultiText Experiments for TREC 2004)," presented at TREC 2004, 2004.

[8] H. Shatkay and R. Feldman, "Mining the biomedical literature in the genomic era: an overview," *Journal of Computational Biology*, vol. 10, pp. 821-855, 2003.

[9] S. Pakhomov, "Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical text," presented at the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002.

[10] D. Hanisch, J. Fluck, H. Mevissen, and R. Zimmer, "Playing biology's name game: identifying protein names in scientific text," presented at Pacific Symposium on Biocomputing '03, 2003.

[11] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri: A language-model based search engine for complex queries," presented at Internaional Conference on Intelligence Analysis 2005, 2005.

[12] W. R. Hersh, R. T. Bhuptiraju, L. Ross, A. M. Cohen, and D. F. Kraemer, "TREC 2004 Genomics Track Overview," presented at TREC 2004, 2004.

[13] C. Buckley, "trec eval IR evaluation package."

[14] NLM, "Lexical Variation Generator," 2005.