ALPEN-ADRIA UNIVERSITÄT
KLAGENFURT I WIEN GRAZ

Textual Methods for Medical Case Retrieval

Mario Taschwer

Supervisor: Prof. Laszlo Böszörmenyi, AAU
Co-Supervisor: Prof. Oge Marques, FAU, Florida

Institute of Information Technology (ITEC)
Alpen-Adria-Universität Klagenfurt, Austria
Technical Report No. TR/ITEC/14/2.01 v1.0
May 2014

Abstract

Medical case retrieval (MCR) is information retrieval in a collection of medical case descriptions, where descriptions of patients' symptoms are used as queries. We apply known text retrieval techniques based on query and document expansion to this problem, and combine them with new algorithms to match queries and documents with Medical Subject Headings (MeSH). We ran comprehensive experiments to evaluate 546 method combinations on the ImageCLEF 2013 MCR dataset. Methods combining MeSH query expansion with pseudo-relevance feedback performed best, delivering retrieval performance comparable to or slightly better than the best MCR run submitted to ImageCLEF 2013.

Keywords: medical information retrieval, query expansion, Medical Subject Headings

Contents

1 Introduction

Medical case retrieval (MCR) is the problem of finding descriptions of diseases or patients' health records (document corpus) that are relevant for a given description of patient's symptoms (query), as decided by medical experts. MCR is a major building block of clinical decision support systems [39] employing the paradigm of case-based reasoning [1, 10], where the "most similar" medical cases need to be retrieved for a given symptom description before diagnosis and treatment can be proposed by the system. Moreover, MCR is also a relevant problem in medical education and research, because it allows to select interesting cases for students and to generate datasets for studies meeting case-based criteria.

Case and symptom descriptions are multimedia documents, typically consisting of structured text and medical images. Designing an automatic MCR system applicable to general medical datasets (as opposed to datasets in narrow medical domains, see [10]) still presents an open research problem. The ImageCLEF evaluation campaign[1] [55] issued a yearly MCR challenge between 2009[2] and 2013, leading to a general biomedical dataset of about 75,000 documents ("case descriptions") and 35 queries (symptom descriptions) in 2013. The moderate performance of even the best MCR runs submitted to ImageCLEF 2013 (about 24% MAP) emphasizes the need for further research regarding MCR techniques. Interestingly, the best MCR runs submitted to ImageCLEF 2013 employed text retrieval techniques only, any approach combining text retrieval with content-based image retrieval reduced retrieval performance dramatically [28].

This work therefore focuses on textual MCR methods capable of delivering the same (or better) retrieval performance as the best systems of ImageCLEF 2013 participants. A well-known methodology to improve plain text retrieval on general datasets is query expansion [17], where relevant terms generated from some data source are added to the original query, prior to sending the expanded query to the retrieval system. A closely related technique is document expansion, where additional relevant terms are added to documents at indexing time. Due to the wealth of query expansion methods proposed in the information retrieval literature (and the lack of available implementations), it is not feasible to systematically evaluate even only the key methods on the MCR dataset. On the other hand, it would be interesting to utilize data sources for query expansion that are specific to the medical domain.

We therefore chose to apply a well-known pseudo-relevance feedback technique inspired by Rocchio's method [66] for query expansion, and combine it with several novel algorithms to associate queries or documents with Medical Subject Headings (MeSH)[3]. MeSH is a thesaurus of biomedical terms used to index PubMed[4] publications, a large collection of biomedical publications that the ImageCLEF 2013 MCR dataset has been sampled

[1] `http://www.imageclef.org/`

[2] ImageCLEF medical retrieval tasks were issued yearly starting in 2004, but the tasks before 2009 were rather image retrieval tasks than case retrieval tasks. The distinction is blurred, however.

[3] `http://www.nlm.nih.gov/mesh/`

[4] `http://www.ncbi.nlm.nih.gov/pubmed`

from. We consider several variants of these query and document expansion methods and systematically evaluate more than 500 method combinations on the ImageCLEF 2013 MCR dataset. Experimental results reveal that a combination of MeSH query expansion with pseudo-relevance feedback is able to deliver state-of-the-art retrieval performance on this dataset, but additional use of document expansion has no further benefit.

The contributions of this work are (1) novel efficient algorithms to associate queries or documents with MeSH terms, that do not rely on natural language processing or machine learning; and (2) a comprehensive evaluation of query and document expansion methods based on MeSH terms and pseudo-relevance feedback that achieve state-of-the-art retrieval performance on a recent MCR dataset.

The paper is organized as follows: Section 2 reviews relevant information retrieval techniques from the literature, emphasizing query expansion methods. Query and document expansion methods evaluated in our experiments are described in detail in Section 3, which includes our novel MeSH term matching algorithms (Section 3.2) and MeSH synonym handling methods (Section 3.3.1). All query expansion methods depend on a number of free parameters that need to be optimized on a validation set prior to testing. The adopted parameter optimization algorithm is described in Section 4. Experimental results are presented in Section 5 together with a description of the dataset and cross-validation technique used for experiments (Section 5.1). Section 6 concludes the paper.

2 Information Retrieval Techniques

Classical information retrieval has been dealing with text retrieval for several decades, and a number of traditional techniques has proven to provide robust and efficient tools to perform text retrieval on general datasets. We chose some of these known methods for our approach to textual MCR for two reasons: (1) to the best of our knowledge, there is no recent technique for text retrieval on general medical datasets that performs substantially better than traditional text retrieval methods; and (2) evaluation and comparison with other approaches becomes more meaningful if they are based on well-known "standard" techniques. This section reviews relevant techniques from the information retrieval literature to establish the state of the art this work is based on. Special attention is paid to query expansion methods.

2.1 Information Retrieval Models

As described in many textbooks on information retrieval (e.g. [61, 49, 7]), two standard models of text retrieval are the *vector space model* [70] and the *probabilistic model* [65], combined with TF-IDF [77, 62, 83] or BM25 [63] term weighting. These methods are able to deliver state-of-the-art text retrieval performance, and mature open-source implementations are available, most notably Lucene[5] and Indri[6] [51].

[5] `http://lucene.apache.org/`

[6] `http://www.lemurproject.org/indri/`

There are several alternative information retrieval models that can be classified into set-theoretic, algebraic, and probabilistic models [7]. Two prominent alternative probabilistic models are language models [67, 44] and divergence from randomness [5]. The latter has been found to be the most effective model on a biomedical dataset [2]. However, due to the lack of available implementations we did not consider these models for experimental evaluation.

Our experiments presented in section 5 use Lucene version 3.6.2 with its default implementation of the vector space model[7]. Lucene defines a variant of TF-IDF weighting $w(t, d)$ of term t in document d as:

$$w(t,d) = \sqrt{\mathrm{TF}(t,d)} \cdot \left(1 + \log \frac{N}{\mathrm{DF}(t)+1}\right) \tag{1}$$

where $\mathrm{TF}(t, d)$ denotes the number of occurrences of term t in document d (term frequency), N is the number of documents in the dataset, and $\mathrm{DF}(t)$ is the number of documents in the dataset that contain term t (document frequency).

2.2 Query Expansion

A fundamental limitation of retrieval performance of textual information systems is the mismatch of words used to express the same concepts in the query and in the document collection, known as the *vocabulary problem* in information retrieval. One methodology to address this problem, called *query expansion* (QE), is to automatically expand the user's query with words related to the user's information need (i.e. the query *topic*) before sending the query to the retrieval system. From the variety of QE techniques proposed during the last four decades, we try to summarize the key methods and principles, as described and classified in a recent survey by Carpineto and Romano [17].

Alternative methodologies to overcome the vocabulary problem are interactive query refinement (e.g. [7]), relevance feedback [68], word sense disambiguation [56], and search results clustering [16]. The first two alternatives cannot be applied to the MCR task covered by this work, as they require interactive user input. Word sense disambiguation techniques do not seem to provide any advantages over QE with respect to effectiveness and efficiency of information retrieval [3, 17], so they have not been investigated in this work. Search results clustering has typically been employed for browsing through web search results and does not seem to be beneficial for the automatic MCR task and rather small dataset considered here.

2.2.1 Query Expansion Process

Query expansion works by leveraging external or in-collection data sources to generate and select expansion features used to reformulate the original query. A general process

[7] See Java class `org.apache.lucene.search.Similarity` in API documentation at `https://lucene.apache.org/core/3_6_2/api/core/`

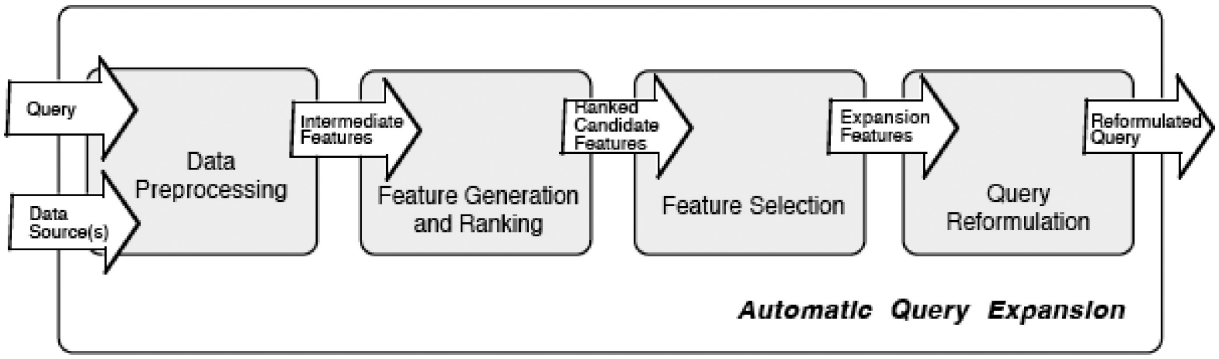


Figure 1: Stages of query expansion process, taken from [17].

pipeline common to all QE techniques proposed so far consists of four stages (Figure 1): (1) preprocessing of data sources, often performed at indexing time; (2) generation and ranking of candidate expansion features; (3) selection of expansion features; and (4) query reformulation.

To illustrate the process pipeline and to describe a QE method used in our experiments, consider the following simple pseudo-relevance feedback approach inspired by Rocchio's relevance feedback method [66, 17]. An inverted index implementing a vector space model using TF-IDF weights is used initially to retrieve a ranked list of documents matching the original query. This list of documents acts as data source for QE, and stage (1) of the process pipeline needs to ensure that the inverted collection index allows to access the TF-IDF weights of terms. The TF-IDF weights of every term (word) in the n top-ranked documents are summed up, and terms are sorted by their accumulated weight. Initial retrieval and sorting terms of top retrieved documents represent stage (2). Finally, the first k terms of the sorted list (stage (3)) are added to the original query (stage (4)).

From the four process pipeline stages, feature generation and ranking (2) is the most critical one and gave rise to a large variety of proposals in the literature. We try to identify the key approaches in Section 2.2.2. The feature generation method determines the required preprocessing (1), and the ranking method enables or disables certain feature selection techniques (3).

Feature selection Selecting the first k features is always possible, and there is empirical evidence that a value of k between 10 and 30 is a good choice for many general datasets, because retrieval performance decreases only slowly for sub-optimal values of k [17]. When the feature scores allow for consistent semantic interpretation (e.g. as probabilities), features with a score greater than a certain threshold can be selected. It is known that, on average over many queries, a rather large fraction of terms selected by these simple approaches are harmful to retrieval performance [14]. Several advanced feature selection methods have been proposed to improve the fraction of relevant expansion terms for a given query, including the combination of multiple term ranking functions [18], generating multiple feedback models by resampling documents and varying the query [24], choosing k as a function of the ambiguity of the (Web) query [20], employing supervised learning

to discriminate between relevant and irrelevant expansion terms [14], and solving an optimization problem with respect to uncertainty sets [22].

Query reformulation The simplest method for query reformulation (4) is to add the selected expansion features to the original query without modifying their weights. The most common approach, however, is to give different weights to terms of the original query and to expansion terms, and to incorporate the score of expansion features computed in stage (2). A general formulation based on Rocchio's reweighting formula for relevance feedback [66, 69] is the following.

$$w'_{t,q'} = (1 - \lambda) \cdot w_{t,q} + \lambda \cdot s_t \cdot w_{t,Q} \tag{2}$$

Here $w_{t,q}$ and $w_{t,Q}$ are the weights assigned by the underlying retrieval system to term t within the original query q and within the sequence Q of expansion terms, respectively. s_t is the term score computed in stage (2), λ is a parameter ($0 \leq \lambda \leq 1$) to set the relative importance of expansion terms with respect to original query terms, and $w'_{t,q'}$ is the modified weight of term t in the expanded query q'. If the order of magnitude of expansion term scores s differs from 1, normalization is needed [82]. Alternatively, the values s_t can be computed from an inverse function of term ranks produced in stage (2) [18, 35].

Although giving expansion terms a fixed lower importance than original query terms (e.g. $\lambda = 0.3$) is common practice, a query-specific value of λ can also be predicted by supervised learning in a pseudo-relevance feedback setting [47]. Alternatively, a parameter-free query reweighting method has been proposed [4].

When expansion features are generated using a thesaurus or ontology, score values s_t may accommodate properties and relationships of nodes in the term network [38], or the importance factor λ may depend on the type of such properties and relationships [80].

In language modeling approaches of information retrieval [44, 7], query reweighting arises naturally by smoothing the probability distribution of query terms (*query model* θ_q) with that of query expansion terms (*query expansion model* θ_Q), in analogy to smoothing the document model with the collection model [86]. When applying the Jelinek-Mercer interpolation [37] to smoothing the query model, the probability distribution of the final expanded query model is given by

$$p(t|\theta'_q) = (1 - \lambda) \cdot p(t|\theta_q) + \lambda \cdot p(t|\theta_Q), \tag{3}$$

which is analogous to reweighting formula (2).

A more general approach to query reformulation is to use Boolean [34] or structured queries [23], or the advanced query formulation features of recent query languages like Indri[8], as proposed in [6] for instance.

[8] `http://www.lemurproject.org/indri/`

2.2.2 Query Expansion Approaches

Following and extending the classification of Carpineto and Romano [17], we give an overview of known query expansion techniques according to the conceptual paradigms used to generate expansion features (stage (2) of the query expansion process, Figure 1). For each class of techniques, we try to identify the key approaches characterizing the main ideas and results of its class.

We can distinguish five classes of query expansion approaches: (1) those based on linguistic analysis, (2) corpus-specific global techniques, (3) query-specific local techniques, (4) approaches using external knowledge models, and (5) other innovative techniques that do not fit into the former classes.

Linguistic Analysis Approaches applying linguistic analysis use morphological, lexical, syntactic, or semantic word relationships to generate expansion features from query words. A frequently used technique is stemming [40, 36, 58], which replaces inflected or derivational forms of a word by its stem, usually at indexing time. Syntactic analysis has been used to derive relationships between parse trees of query and top-ranked passages, in order to learn the most relevant relations for the query [78]. Semantic associations of words are often represented by thesauri or ontologies, which are the subject of class (4).

Corpus-specific global techniques These techniques use information extracted from the the entire collection of documents during the pre-processing stage to derive associations between the query and candidate expansion features. Early approaches exploited term co-occurrence at the document or passage level, but could not consistently improve retrieval performance [54]. Two successful key strategies are term concepts [59] and term clustering [27, 72, 8]. *Term concepts* are vector representations of terms indexed by document weights, which can be viewed as a dual representation of the standard document vector space model. The query is represented as a linear combination of term concepts and compared to indexed term concepts by cosine similarity. The resulting ranked list of expansion term candidates is supposed to be more relevant to the whole query than to individual query terms.

The *term clustering* approach of Crouch and Yang [27] clusters documents by cosine similarity and assigns low-frequency terms of clusters to term classes, which are used as synonym classes for query expansion. Schütze and Pedersen [72] efficiently construct a thesaurus of terms sharing neighbors in the document corpus (second-order co-occurrence) by iterative clustering of columns of co-occurrence submatrices, followed by an SVD decomposition that allows to represent terms by dense 20-dimensional real-valued vectors. However, the authors do not use the thesaurus directly for query expansion (although this would be possible), but perform retrieval on document representations derived from term vectors (context vectors). The advantage of global techniques, namely the generation of potentially discriminative features for query expansion, is also their main limitation: features that co-occur frequently in the document collection may be irrelevant for the given query.

Query-specific local techniques The aforementioned problem is addressed by query-specific local techniques, which aim at utilizing the local context provided by the query for expansion. Usually top-ranked documents retrieved in response to the original query (also called *pseudo-relevant documents*) are analyzed to generate expansion features. A simple and well-known method, inspired by Rocchio's relevance feedback technique [66], is *pseudo-relevance feedback*, where collection-based term weights (e.g. TF-IDF weights) are collected from pseudo-relevant documents and used to rank terms as expansion candidates. However, the effectiveness of this approach may be limited by the fact that top-ranking terms may not be relevant for the query, although discriminative for the entire collection.

More advanced local key approaches are analysis of *feature distribution difference*, *query language modeling* and *document summarization*. The former derive term-ranking functions from measuring the term distribution difference between the set of pseudo-relevant documents and the entire collection. Well-known instances of term distribution difference models are the binary independence model [65], the chi-square distance [31], Robertson's selection value [64], and the Kullback-Leibler distance [15]. More term-ranking functions and an experimental study comparing different methods are reported by Wong et al. [82].

Query language modeling approaches estimate a term probability distribution (language model) for the query and consider the most likely terms for query expansion. The query language model is typically estimated using pseudo-relevant documents, as is done by the two main representatives: the *mixture model* [85] and the *relevance model* [42]. The former considers the likelihood of pseudo-relevant documents as a mixture of the query topic model and the collection language model. The query topic model is estimated using the expectation-maximization algorithm [29] as to maximize the likelihood of pseudo-relevant documents. The relevance model assumes that both the query and pseudo-relevant documents are samples from the same unknown term probability distribution $p(t|\theta_R)$ (θ_R is the relevance model). Using the conditional probability of term t given that the original query words have just been observed, an efficient expression for estimating $p(t|\theta_R)$ from pseudo-relevant documents can be derived. Metzler and Croft [52] propose an important generalization of the relevance model that incorporates term dependencies and proximity-based features by modeling the joint distribution of query and relevant document by Markov random fields.

Document summarization techniques preprocess pseudo-relevant documents to represent them by more compact and informative features before applying a term-ranking function. *Local context analysis* [84] uses term-concept co-occurrence extracted from passages (text windows of fixed size) of pseudo-relevant documents, where a concept is a group of adjacent nouns. Other approaches use text summarization techniques [41] or intra-document feature clustering and classification [19].

External knowledge models Query expansion techniques using external knowledge models utilize linguistic or domain-specific information not already available in the document collection, but in external knowledge representations like thesauri or ontologies (see

[71] for a discussion on the distinction between these concepts). Ontology-based query expansion is analyzed in [57] and reviewed in [11].

A well-known linguistic thesaurus is WordNet[9] [53], which has frequently been used to find synonyms and related words of query words for general collections [80, 48, 33]. The major problem with the use of WordNet is word sense disambiguation [56], which has been addressed by several advanced approaches [43, 32, 74].

The semantic relationships between concepts defined in knowledge models may be used to generate query expansion features based on their conceptual distance in the semantic network. Liu et al. [45] rank key phrases extracted from pseudo-relevant documents according to their conceptual distance to the query phrase on WordNet. Tudhope et al. [79] assign traversal costs to the relationships in a domain-specific thesaurus and generate expansion concepts by traversing the semantic network until a predefined cutoff distance threshold is reached. Candidate concepts are ranked by their average conceptual distance to all query terms.

In the medical domain, many ontologies and thesauri have been developed to store and classify medical knowledge [9, 30]. Some of them are UMLS[10] [13], SNOMED, ICD, RadLex, and MeSH[11]. Query expansion using the MeSH thesaurus has been applied to medical case retrieval with varying success. Diaz-Galiano et al. [30] observed a significant increase in retrieval performance on the ImageCLEF 2005 and 2006 MCR datasets, whereas Mata et al. [50] could not using the ImageCLEF 2011 dataset. However, the latter authors reported a more successful approach in [26].

Other techniques There are some other principled approaches that do not fit into the classes described above. Collins-Thompson and Callan [23] construct a query-specific term network whose relations can be generated from various sources (WordNet, stemmer, external corpus, top retrieved documents) and are assigned transition probabilities. The term network is modeled as Markov chain, and terms with highest probability according to the stationary distribution are selected for expansion. Riezler et al. [60] apply supervised machine learning to translate the query to semantically related phrases, and extract expansion terms from them.

3 Improving IR for Biomedical Collections

To improve retrieval effectiveness for collections of biomedical documents, many of the query or document expansion techniques applicable to general collections may be used (see Section 2). Additionally, it would be interesting to see how these techniques could benefit from domain-specific properties of biomedical collections. An obvious approach is to use a biomedical ontology or thesaurus as external knowledge source for query expansion, and

[9] `http://wordnet.princeton.edu/`
[10] `http://www.nlm.nih.gov/research/umls/`
[11] `http://www.nlm.nih.gov/mesh/`

combine it with some query-specific local technique that is known to work well with general collections. The contribution of each query expansion method to retrieval effectiveness as well as the synergy effect of combining them can then be analyzed experimentally.

We chose to use *Medical Subject Headings* (MeSH) as biomedical thesaurus, because MeSH annotations were available with the dataset used for experiments (see Section 5). Query expansion using MeSH is combined with a simple pseudo-relevance feedback scheme based on Rocchio's method [66], using TF-IDF weights for term ranking. The reason for choosing this simple query-specific local method is primarily its relative low implementation cost, which enabled us to add more variants and combinations of selected techniques. Moreover, such a combination of techniques has been rarely studied in the literature [2], although there are some results using MeSH query expansion alone [46, 30, 50, 26].

Query expansion with MeSH terms relies on the ability to associate relevant MeSH terms with the query. Although this mapping ability is implemented in the popular search engine PubMed[12] for biomedical publications, called Automatic Term Mapping (see e.g. [46]), we developed an alternative MeSH term matching algorithm, because Automatic Term Mapping is not accessible via an API. Moreover, our MeSH term matching algorithm cannot only be applied to queries but also to documents to allow for automatic MeSH term annotation used for document expansion.

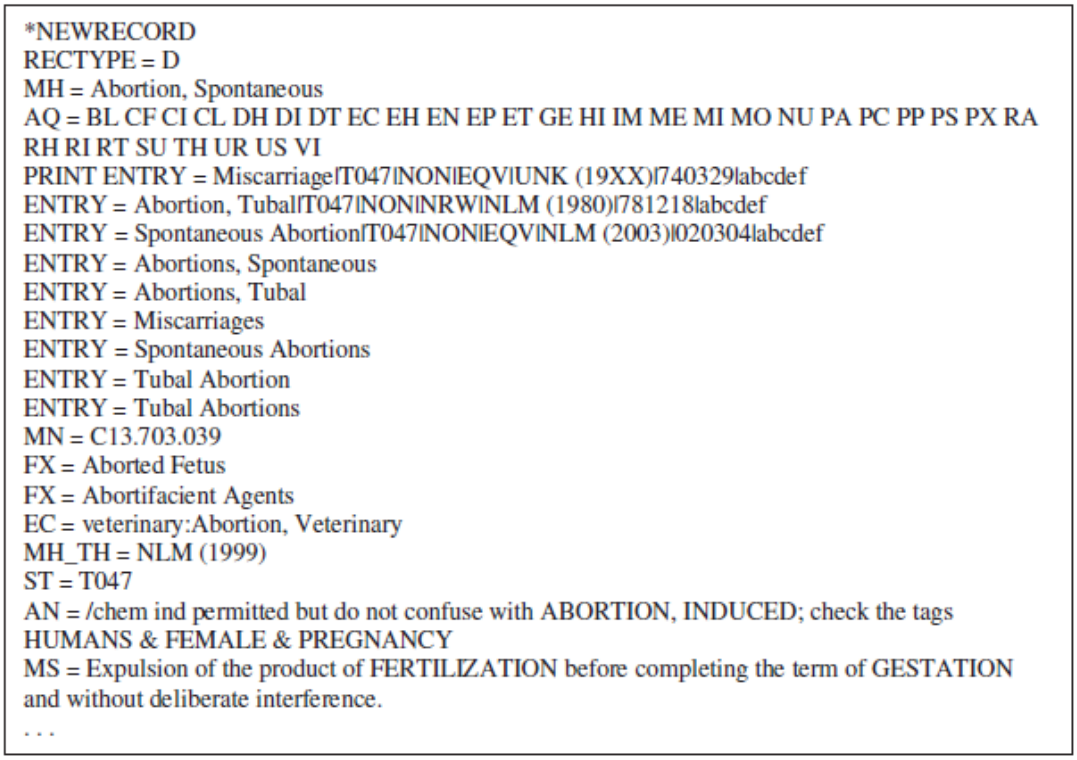
The following sections describe the methods and their combinations used for experimental evaluation in detail. For better understanding of MeSH term matching, we initially provide a brief description of the MeSH thesaurus.

3.1 Medical Subject Headings

Medical Subject Headings[13] (MeSH) are a controlled vocabulary used to index biomedical publications. The MeSH thesaurus consists of *records* representing the nodes of a tree structure. A record describes a *primary MeSH term* and, among other information, a number of *synonyms* (Figure 2). A parent node in the tree represents a more general term than its child nodes. The child nodes of the root node (let us call them *top-level nodes*) are listed in Table 1. Following the approach of Diaz-Galiano et al. [30], we used only 3 top-level nodes for query expansion (nodes A, C, and E). The 3 selected subtrees contain 8,911 primary MeSH terms and 64,201 synonyms.

Every MeSH record is assigned a node identifier given by its *MN* field. The MeSH record shown in Figure 2 has the node identifier *C13.703.039*. The number of dots in the node identifier is an indication of depth of the node in the MeSH tree. We call it *MeSH term specialty*, as deeper nodes refer to more special MeSH terms. Table 2 lists some primary MeSH terms and their specialty values.

[12] `http://www.ncbi.nlm.nih.gov/pubmed`
[13] `http://www.nlm.nih.gov/mesh/`



```
*NEWRECORD
RECTYPE = D
MH = Abortion, Spontaneous
AQ = BL CF CI CL DH DI DT EC EH EN EP ET GE HI IM ME MI MO NU PA PC PP PS PX RA
RH RI RT SU TH UR US VI
PRINT ENTRY = Miscarriage|T047|NON|EQV|UNK (19XX)|740329|abcdef
ENTRY = Abortion, Tubal|T047|NON|NRW|NLM (1980)|781218|abcdef
ENTRY = Spontaneous Abortion|T047|NON|EQV|NLM (2003)|020304|abcdef
ENTRY = Abortions, Spontaneous
ENTRY = Abortions, Tubal
ENTRY = Miscarriages
ENTRY = Spontaneous Abortions
ENTRY = Tubal Abortion
ENTRY = Tubal Abortions
MN = C13.703.039
FX = Aborted Fetus
FX = Abortifacient Agents
EC = veterinary:Abortion, Veterinary
MH_TH = NLM (1999)
ST = T047
AN = /chem ind permitted but do not confuse with ABORTION, INDUCED; check the tags
HUMANS & FEMALE & PREGNANCY
MS = Expulsion of the product of FERTILIZATION before completing the term of GESTATION
and without deliberate interference.
...
```

Figure 2: Example MeSH record [30]. The primary MeSH term is given by the MH field, the ENTRY fields denote synonyms.

Table 1: Top-level nodes of MeSH tree structure. Only the subtrees represented in bold face were used for query expansion.

Anatomy [A]	Anthropology, Education, Sociology and Social Phenomena [I]
Organisms [B]	Technology, Industry, Agriculture [J]
Diseases [C]	Humanities [K]
Chemicals and Drugs [D]	Information Science [L]
Analytical, Diagnostic, Therapeutic Techniques and Equipment [E]	Named Groups [M]
Psychiatry and Psychology [F]	Health Care [N]
Phenomena and Processes [G]	Publication Characteristics [V]
Disciplines and Occupations [H]	Geographicals [Z]

Table 2: Some primary MeSH terms and their associated specialty values (number of dots in node identifier).

Primary MeSH Term	**Node Identifier**	**Specialty**
Abortion, Spontaneous	C13.703.039	2
Pregnancy Complications	C13.703	1
Female Urogenital Diseases and Pregnancy Complications	C13	0
Kidney Pelvis	A05.810.453.537	3
Kidney	A05.810.453	2
Urinary Tract	A05.810	1
Urogenital System	A05	0

3.2 MeSH Term Matching

A naive approach to finding relevant MeSH terms of a given query is to use an existing information retrieval system to index MeSH terms and execute the query to retrieve a ranked list of MeSH terms. However, this method is likely to be ineffective, because retrieval systems have not been designed to index very short documents (i.e. MeSH terms) and to execute queries that may well be longer than the average document length. Moreover, such a method would be clearly too inefficient to retrieve relevant MeSH terms for long documents, as required for document expansion by automatic MeSH annotation.

We therefore developed several MeSH term matching algorithms that enable an efficient generation of a ranked list of MeSH terms supposed to be relevant for a given query or long document. All algorithms work by accumulating MeSH term scores during a single pass through the query or document, followed by score normalization and optional MeSH term specialty boosting. The latter method favors MeSH terms at greater depth in the MeSH tree (i.e. more special terms) as opposed to more general terms. The algorithms are listed below. Their components are described in the following sections.

t0 – BinCov binary coverage

t1 – Dist distance-based match frequency

t2 – BinDist combination of *BinCov* and *Dist* for matching runs

t3 – IdfBinDist *BinDist* with score boosting by maximal IDF of MeSH term words

t4 – IdfCovDist combination of *Dist* with IDF-based run coverage

3.2.1 Basic Algorithm and Data Structures

For the purpose of MeSH term matching, the notion of a *MeSH term* always refers to a single synonym of a MeSH record (see Section 3.1), that is, MeSH term matching is performed on lexical entities, not on semantic concepts.

All algorithms use an inverted index of MeSH term words. Every word of the MeSH thesaurus (or the used part of it) is linked to a list of MeSH terms containing that word. When building the index, words are lower-case-filtered, and punctuation characters are removed. Stop words are not removed, because they may be significant for a MeSH term (as in `Vitamine A`). Since MeSH often contains plural forms of MeSH terms as synonyms, and to favor exact matches, word stemming is not applied.

When processing a query or document, the same preprocessing is applied as for building the inverted index, and for each word of the query or document all MeSH terms containing that word are visited. Visited MeSH terms maintain local statistics depending on the algorithm in use. When query or document processing has finished, all visited MeSH terms are updated to produce final scores by performing score normalization and specialty boosting. Finally, visited MeSH terms are sorted by score, and the list of matching MeSH terms is obtained by thresholding. In fact, the implementation uses a priority queue to assemble the final sorted list of MeSH terms to avoid sorting all visited MeSH terms.

MeSH term matching algorithms differ only in the way they accumulate statistics and compute the final score of visited MeSH terms. The different scoring functions are described in the following sections. To simplify description, we refer to MeSH term matching of documents only, but the algorithms apply to matching queries as well.

3.2.2 Coverage

We define the ratio of words of MeSH term t occurring in a document d as the *coverage* $\mathbf{Cov}(t, d)$ of this MeSH term in the document. Word order and number of occurrences of the same word are ignored. For example, given the document "Abdominal CT scan revealed a large left renal mass with extension into the left renal pelvis and ureter.", the coverage of MeSH term `Pelvis, Renal` is 1.0 and that of MeSH term `Pelvis Cancers` is 0.5.

Obviously, this scoring function makes sense only for queries or very short documents, as longer documents will raise the scores of many irrelevant MeSH terms to 1.0, because their constituent words are spread over the entire document.

An even simpler scoring function that is only used in combination with other functions described below is the *binary coverage* $\mathbf{BinCov}(t, d)$. It is defined as 1 when all words of MeSH term t occur in document d, and 0 otherwise.

3.2.3 Distance-based Match Frequency

To make MeSH term matching sensitive to word order and to the proximity of MeSH term words occurring in the document, we define the score as a function of relative positions of MeSH term words in the document. Let $t = t_1\, t_2\, \ldots\, t_T$ be the constituent words of MeSH term t, $p_1 < p_2 < \cdots < p_N$ the word positions within document d containing MeSH term words t_i, and $r_1, r_2, \ldots, r_N$ the corresponding MeSH term word indexes, i.e. the word at document position p_i is MeSH term word t_{r_i}. (If the MeSH term t contains the same word at multiple positions and this word occurs at position p_i in the document, then we define r_i as the minimum of those positions in t.) The scoring function $\mathbf{Dist}(t, d)$ is then defined as follows:

$$s(p, r) = \begin{cases} (p\, r)^{-1} & \text{if } r > 0, \\ 0 & \text{if } r = 0, \\ (p\, (2 - r))^{-1} & \text{if } r < 0. \end{cases} \tag{4}$$

$$\mathbf{Dist}(t, d) = \begin{cases} \sum_{i=1}^{N-1} s(p_{i+1} - p_i,\, r_{i+1} - r_i) & \text{if } N > 1 \text{ and } T > 1, \\ 0 & \text{if } N = 1 \text{ and } T > 1, \\ N & \text{if } T = 1. \end{cases} \tag{5}$$

Note that $s(p, r)$ is defined for $p > 0$ only, and $s(p, r) > s(p, -r)$ if $r > 0$. The intention behind these formulas is that M exact occurrences of the MeSH term in the document shall give a score of approximately $M\,(T - 1)$ if $T > 1$, but shall allow also for partial matches and word re-orderings with a penalty. The scoring function can therefore be viewed as a distance-based *soft match frequency* of MeSH term words. The score is not normalized with respect to MeSH term length T in order to favor longer MeSH terms.

For example, when calculating the score of MeSH term `Pelvis, Renal` for the short document of the previous section, we have $p_1 = 8$, $p_2 = 15$, $p_3 = 16$ and $r_1 = 2$, $r_2 = 2$, $r_3 = 1$, resulting in the score $0 + 1/3 = 0.333$. MeSH term `Pelvis Cancers` has score 0 for the same document, because `pelvis` occurs only once and `cancers` does not occur.

3.2.4 Run Coverage and Match Frequency

The **Dist** scoring function described in the previous section may give rather high values for MeSH terms containing some frequently occurring word groups, although the entire MeSH term is not contained in the document. The most prominent such word group is `of the`, which is part of many MeSH terms (e.g. `Cancer of the Uterus`, `Infarct of the Spleen`, `Exstrophy of the Bladder`). To address this problem, we introduce the notion of *matching runs* and restrict the **BinCov** and **Dist** scoring functions to those runs.

Using the notation of the previous section, we define a *matching run* as a maximal subsequence $(p_i, p_{i+1}, \ldots, p_k)$ of matching positions of a MeSH term in a document, such that $p_{j+1} - p_j \leq \beta$ for all $j \in [i, k - 1]$ and a fixed parameter β (e.g. $\beta = 3$). In other

words, matching runs are groups of consecutive matching positions separated from other such groups by more than β positions. Note that the boundaries between matching runs can be easily determined during a single pass through the document.

The **BinDist** scoring function is computed from products of **BinCov** and **Dist** functions restricted to matching runs $\pi_1, \ldots, \pi_R$ of MeSH term t in document d:

$$\mathbf{BinDist}(t, d) = \sum_{i=1}^{R} \mathbf{BinCov}(t, \pi_i)\ \mathbf{Dist}(t, \pi_i) \tag{6}$$

The restriction of binary coverage to matching runs is called *run coverage*. If run coverage is 1 for all matching runs, the **BinDist** score will approximate the **Dist** score, because the run distance β will limit inter-run contributions of $\mathbf{Dist}(t, d)$ to small values. The **BinDist** scoring function effectively ignores all partial occurrences of a MeSH term in the document, but allows for word permutations and intermixing with other words within matching runs.

For example, considering the short document d given in Section 3.2.2 and MeSH term t = `Pelvis, Renal`, there are two matching runs for $\beta = 3$: $\pi_1 = (8)$, $\pi_2 = (15, 16)$. We have $\mathbf{Dist}(t, \pi_1) = 0$, $\mathbf{Dist}(t, \pi_2) = 1/3$, and $\mathbf{BinCov}(t, \pi_2) = 1$, so $\mathbf{BinDist}(t, d) = 0.333$.

3.2.5 Boosting MeSH Terms by IDF

A major problem with scoring functions based on match frequency is that one-word MeSH terms occurring several times in a document obtain higher scores than multi-word MeSH terms occurring only once. However, the long MeSH term may be equally relevant, because it denotes a medical concept that is rarely mentioned in the document collection. On the other hand, many one-word MeSH terms occur in a large fraction of documents in the collection, so their importance of being relevant for a given document should be decreased. This observation calls for integration of *inverse document frequency* (IDF) of MeSH terms into the scoring function, which takes greater values for MeSH terms occurring less frequently in the document collection.

When defining IDF of MeSH terms, we need to take into account that not all MeSH terms occur in the document collection at hand, and that counting the document frequency of MeSH terms may require automatic MeSH term matching, resulting in a recursive problem. Additionally, the question of how to count synonyms of MeSH terms should be answered. We worked around these problems by defining the *IDF of a MeSH term* as the maximal IDF value of its constituent words. That is, we reduce the global importance of a MeSH term to its most discriminative word with respect to the collection.

The IDF value of a MeSH term word remains to be defined as it may not occur in the document collection at all. Additionally, we have to take care of stop words (e.g. `of` and `the`), which are usually not indexed or counted by the retrieval system. Let w denote a word of a MeSH term, let N be the number of documents in the collection, and n_w the

document frequency of w in the collection (i.e. the number of documents containing w) if w has been indexed by the retrieval system. We call w a *collection stop word* if it is a common English stop word or if it occurs in all N documents of the collection. If w does not occur in the collection (and hence is not a common English stop word with high probability), we call it an *external term*.

$$\mathrm{IDF}(w) = \begin{cases} \varepsilon & \text{if } w \text{ is a collection stop word,} \\ (\log N)/2 & \text{else if } w \text{ is an external term,} \\ \log(N/n_w) & \text{otherwise.} \end{cases} \tag{7}$$

We assign some small positive IDF value $\varepsilon < 1$ (we used $\varepsilon = 0.1$ in our experiments) to collection stop words, for reasons explained in the next section. External terms receive half of the maximal IDF value possible for collection terms. Note that $\mathrm{IDF}(w) > 0$ in all cases. The IDF value of MeSH term $t = t_1\ t_2\ \ldots\ t_T$ is defined as explained earlier and used to boost the **BinDist** score:

$$\mathrm{IDF}(t) = \max_i \mathrm{IDF}(t_i) \tag{8}$$

$$\mathbf{IdfBinDist}(t, d) = \mathrm{IDF}(t) \cdot \mathbf{BinDist}(t, d) \tag{9}$$

3.2.6 IDF-weighted Run Coverage

The binary run coverage used by **BinDist** and **IdfBinDist** scoring functions effectively ignore partial matches of MeSH terms in a document, in the sense that runs missing only one word of a MeSH term do not contribute to the matching score. However, such runs can be regarded as relevant for the MeSH term if the missing word has low discriminative power in the document collection, which is the case for e.g. collection stop words (see Section 3.2.5).

An alternative approach to improving the **BinDist** scoring function is by allowing this kind of partial matches to contribute to the score. This is achieved by replacing the binary run coverage by an *IDF-weighted run coverage* **IdfCov** of matching runs $\pi_1, \ldots, \pi_R$ of MeSH term $t = t_1\ t_2\ \ldots\ t_T$ in document d:

$$\mathbf{IdfCov}(t, \pi) = \frac{\sum_{i=1}^{T} \mathrm{IDF}(t_i) \cdot \mathbf{BinCov}(t_i, \pi)}{\sum_{i=1}^{T} \mathrm{IDF}(t_i)} \tag{10}$$

$$\mathbf{IdfCovDist}(t, d) = \sum_{i=1}^{R} \mathbf{IdfCov}(t, \pi_i) \cdot \mathbf{Dist}(t, \pi_i) \tag{11}$$

The binary coverage $\mathbf{BinCov}(t_i, \pi)$ is 1 if MeSH term word t_i occurs in matching run π, and 0 otherwise. $\mathrm{IDF}(t_i)$ has been defined in Equation (7), and the **Dist** scoring function is the same as in Section 3.2.4. The definition of **IdfCov** also explains why $\mathrm{IDF}(t_i)$ has been

defined to be positive for all MeSH term words t_i: in addition to providing mathematical validity of the fractional expression, it guarantees a penalty for missing MeSH term words in matching runs.

3.2.7 Boosting MeSH Term Specialty

It is reasonable to assume that more special MeSH terms are more relevant to a document, even if they occur less often in the document than more general MeSH terms. We therefore equipped all MeSH term scoring functions described in the previous sections with an optional boost factor based on MeSH term specialty as defined in Section 3.1. So for any scoring function $\mathbf{score}(t, d)$ defined above we also consider a variant $\mathbf{score}_s(t, d)$ boosted by MeSH term specialty $\mathrm{spec}(t)$:

$$\mathbf{score}_s(t, d) = \alpha^{\mathrm{spec}(t)} \cdot \mathbf{score}(t, d) \tag{12}$$

where $\alpha > 1$ is a fixed parameter (we used $\alpha = 1.3$ in our experiments).

3.3 Query Expansion

In order to improve retrieval performance for biomedical document collections, we employ some simple query expansion techniques utilizing two data sources for feature generation: (1) the MeSH thesaurus, and (2) pseudo-relevant (i.e. top-retrieved) documents. The proposed methods fall into the classes *external knowledge models* and *corpus-specific local techniques* described in Section 2.2. The following sections describe the stages of the query expansion process in detail: feature generation (Sections 3.3.1 and 3.3.2), feature selection (Section 3.3.3), and expansion term weighting (Section 3.3.4).

3.3.1 MeSH Term Generation from Query

Using one of the MeSH term matching algorithms described in Section 3.2, a ranked list of MeSH terms (synonyms) supposed to be relevant to a given query can be obtained. As MeSH term matching ignores the synonym relationship between MeSH terms, we propose several *synonym handling methods* to determine the final list of generated features (i.e. MeSH terms):

x0 – direct Only directly matching synonyms are selected.

x1 – primary_replace Each matching synonym is replaced by its corresponding primary MeSH term.

x2 – all_synonyms Each matching synonym is replaced by all synonyms of its corresponding MeSH record.

x3 – primary_filter Directly matching synonyms that are primary MeSH terms are selected. The resulting list is a filtered **direct** list.

In the final list, duplicate synonyms are suppressed, and each MeSH term receives the score of the synonym it has replaced in the original list. For example, given the query "Abdominal CT scan revealed a large left renal mass with extension into the left renal pelvis and ureter", suppose that MeSH term matching results in the scored list (`Ureter`: 1.0; `Pelvis, Renal`: 0.9). `Ureter` is a primary MeSH term, whereas `Pelvis, Renal` is a synonym of the primary MeSH term `Kidney Pelvis`. Here are the final lists resulting from each of the synonym handling methods described above:

x0 (`Ureter`: 1.0; `Pelvis, Renal`: 0.9)

x1 (`Ureter`: 1.0; `Kidney Pelvis`: 0.9)

x2 (`Ureter`: 1.0; `Ureters`: 1.0; `Kidney Pelvis`: 0.9; `Pelvis, Kidney`: 0.9; `Pelvis, Renal`: 0.9)

x3 (`Ureter`: 1.0)

3.3.2 Pseudo-relevance Feedback

The second data source we used for query expansion were top-retrieved documents. The original or MeSH-expanded query is executed by the retrieval system, and the first m documents (called *pseudo-relevant documents*) of the ranked result list are processed to generate another set of expansion features. These are added to the first query to execute the final retrieval run.

For our experiments, we used two types of expansion features generated from pseudo-relevant documents: words ranked by their TF-IDF weight in the collection, and annotated MeSH terms. In addition to single words, we also considered word n-grams (phrases of length n) ranked by TF-IDF weight. MeSH annotations are either available by manual assignment (if available with the dataset) or by automatic MeSH term matching. In fact, we evaluated the following expansion features generated from m pseudo-relevant documents:

rf the first k words (unigrams) ranked by TF-IDF.

rf2 the first k words (unigrams), and the first k_2 bigrams (word 2-grams), both ranked independently by TF-IDF.

rfm all manually annotated MeSH terms.

rfm2 the union of **rf** and **rfm** features.

rfam the first k automatically annotated MeSH terms, ranked by one of the scoring functions described in Section 3.2.

For expansion term weighting, we want all generated features to be associated with a score value. All features mentioned above are already equipped with a score, except for manually annotated MeSH terms. These have been marked by human annotators as *major* or *minor*, expressing whether the MeSH term represents a major topic of the document or not. We used this binary flag to assign different scores to manually annotated MeSH terms: major terms get score 1, minor terms receive a configurable lower fixed score $s_{\min}$. We used $s_{\min} = 0.3$ in our experiments.

3.3.3 Feature Selection

The final expansion features are selected from the ranked lists generated as described in the previous sections by simple thresholds: (1) minimal MeSH term matching score (Section 3.3.1), and (2) number of top-ranked features (parameters k and k_2 in Section 3.3.2). For selecting manually annotated MeSH terms from pseudo-relevant documents (method **rfm**), we also considered reducing the set of MeSH terms to those marked as *major topic* by human annotators, but that resulted in too few or even zero selected terms, because many documents of the dataset have no major topic assigned.

3.3.4 Expansion Term Weighting

The final stage of query expansion is query reformulation (see Section 2.2.1). As we simply add the selected expansion features to the original query, the reformulation problem reduces to choosing expansion term weights. Because all generated features are associated with a score value, we used a variant of Rocchio's reweighting formula (see Equation (2)) to weight expansion terms relative to original query terms:

$$w'_{t,q'} = w_{t,q} + \mu \cdot \frac{s_t}{s_{\max}} \cdot w_{t,Q} \tag{13}$$

where μ is a parameter controlling the relative importance of expansion terms with respect to original query terms, and $s_{\max}$ is the maximum of expansion term scores (assumed to be positive). As in Equation (2), $w_{t,q}$ and $w_{t,Q}$ are the weights assigned by the underlying retrieval system to term t within the original query q and within the sequence Q of expansion terms, respectively. The normalization by $s_{\max}$ allows for unified handling of scoring functions with different scales.

Since some of the pseudo-relevance feedback methods described in Section 3.3.2 combine expansion features generated by two different scoring functions s' and s'' — namely the **rf2** and **rfm2** methods —, we normalized their scores before applying Equation (13) by using a parameter κ to control the relative importance of the two scoring functions:

$$s_t = \begin{cases} s'_t \,/\, s'_{\max} & \text{if } t \text{ was generated by } s', \\ \kappa \cdot s''_t \,/\, s''_{\max} & \text{if } t \text{ was generated by } s''. \end{cases} \tag{14}$$

Table 3: Score thresholds used to select MeSH terms for automatic document annotation. MeSH term matching algorithms are described in Section 3.2.

Document expansion	MeSH term matching	Score threshold
plus1	**t1 – Dist**	0.05
plus2	**t2 – BinDist**	0.001
plus3	**t3 – IdfBinDist**	0.002
plus4	**t4 – IdfCovDist**	0.004

3.4 Document Expansion

Another opportunity to address the vocabulary problem is to add terms to documents describing the topic of a document at indexing time. This may improve retrieval effectiveness if the added terms do not already occur in the original document, or occur only infrequently — provided that those terms occur in the query. This method is known as *document expansion*.

For biomedical datasets external knowledge models containing medical terms are a promising source of features for document expansion, because those terms are likely to occur in queries expressed by users. In our experiments, we expanded biomedical publications by MeSH terms supposed to capture the topic of the publication, adding these terms to the indexed *fulltext* field. In analogy to query expansion, the expansion features were identified by several methods:

- **plus** all manually annotated MeSH terms (whether marked as *major topic* or not) provided with the dataset were used for document expansion.
- **plusN** automatically annotated MeSH terms generated by algorithm **tN** described in Section 3.2 were used for document expansion ($1 \leq N \leq 4$). The score thresholds for MeSH term selection were determined manually by inspecting a few documents of the dataset. They are shown in Table 3. MeSH term matching algorithm **t0** (binary coverage) was excluded as it does not make sense for long documents.

4 Parameter Optimization

The query expansion methods described in Section 3.3 introduce a number of free parameters that need to be chosen carefully to optimize retrieval performance on a given dataset. As there are many combinations of methods to be evaluated and optimal parameter settings are sensitive to the particular method combination in use, an automatic parameter optimization algorithm was applied. Moreover, the use of automatic parameter optimization facilitates evaluation in a cross-validation setting, where only part of the

Table 4: Parameters to be optimized for query expansion methods described in Section 3.3. Not all parameters are relevant for every expansion method.

Parameter	**Type**	**Range**	**Description**
$s_{\min}$	real	0.2 – 2.0	minimal matching score for MeSH term selection
μ_M	real	0.1 – 1.0	weighting factor of MeSH expansion terms relative to original query terms
m	integer	1 – 20	number of pseudo-relevant documents
k	integer	1 – 150	number of expansion terms to use for pseudo-relevance feedback
k_2	integer	1 – 50	number of bigrams to use for expansion for **rf2** method
μ_F	real	0.1 – 2.0	weighting factor of feedback terms relative to original query terms
κ	real	0.1 – 2.0	relative importance of the two scoring functions for **rf2** and **rfm2** methods

Table 5: Statistics about applying SPSA to parameter optimization during 5-fold cross-validation of 546 retrieval configurations (see Section 5). The total number of optimization runs is $5 * 546 = 2730$.

Number of optimization runs	2730	100%
Number of improved runs	2424	89%
Number of converged runs	635	23%
Number of runs yielding optimum in last iteration	270	10%

```
For k = 1:n
        ak = a/(k+A)^alpha;
        ck = c/k^gamma;
        delta = 2*round(rand(p,1)) - 1;
        thetaplus = theta + ck*delta;
        thetaminus = theta - ck*delta;
        yplus = loss(thetaplus);
        yminus = loss(thetaminus);
        g = (yplus - yminus) ./ (2*ck*delta);
        theta = theta - ak*g;
        theta = min(theta, thetamin);
        theta = max(theta, thetamax);
end
theta
```

Figure 3: MATLAB code of SPSA algorithm [76]. Initialization and stopping criterion are not shown.

dataset is used to optimize parameters and the remaining part is used to assess retrieval performance.

The parameters to be optimized for each query expansion method are listed in Table 4. The objective function to be maximized is mean average precision (MAP) of a retrieval run on the dataset (or part of it). Because evaluation of the objective function at a single point in parameter space is a costly operation, we chose an optimization algorithm that tries to keep the number of objective function evaluations low: Simultaneous Perturbation Stochastic Approximation (SPSA) [75, 76]. It has been designed to find a local optimum of continuous-variable problems with smooth objective functions, even if objective function measurements include added noise.

Although sufficient conditions for convergence of SPSA cannot be established for our parameter optimization problem – some parameters take discrete values, and the objective function is not continuous –, we can use SPSA as a vehicle for heuristic optimization of parameters: the algorithm performs a "random walk" in parameter space guided by objective function differences, and we consider the best of visited points as an "optimal" parameter setting. By choosing manually tuned parameter settings as a starting point, we ensure that the result of parameter optimization will not be worse than a previously known "best" parameter configuration. The usefulness of this heuristic application of SPSA becomes evident after the fact when looking at some statistical results of parameter optimization during 5-fold cross-validation of 546 retrieval configurations described in Section 5, as given in Table 5. In 89% of optimization runs, SPSA found better parameter settings, although only 10% of optimizations obtained the best setting in the last iteration (no matter whether SPSA converged or not).

The SPSA algorithm is easy to implement and is shown in Figure 3. It is formulated to minimize a *loss function* y by finding an optimal value of p-dimensional vector $\vec{\theta}$. Starting

Table 6: SPSA parameters used in our experiments.

Parameter	Value	Description
a	1.0	used to compute a_k
A	0	used to compute a_k
α	1.0	used to compute a_k
c	0.1	used to compute c_k
γ	0.5	used to compute c_k
ε	0.001	equality threshold for stopping criterion
K	3	number of stationary iterations for stopping criterion
n	20	maximal iteration count

with an initial guess $\vec{\theta}_1$ and non-negative parameters a, c, A, α, and γ, each iteration k computes an approximation $\vec{g}_k$ of the unknown gradient of y at $\vec{\theta}_k$. The gradient computation (17) requires only two evaluations of the loss function at points $\vec{\theta}_k^+$ and $\vec{\theta}_k^-$ according to Equations (15) and (16). (c_k) is a decreasing sequence of positive numbers and $\vec{\Delta}_k$ is a random perturbation vector whose elements are ± 1, sampled independently from a Bernoulli distribution with probability 1/2. $\vec{\theta}_k$ is then updated to a new value $\vec{\theta}_{k+1}$ (supposed to be closer to the minimum) by adding the negative gradient approximation scaled by a positive number a_k that decreases with k (Equation (18)).

$$\vec{\theta}_k^+ = \vec{\theta}_k + c_k \, \vec{\Delta}_k \tag{15}$$

$$\vec{\theta}_k^- = \vec{\theta}_k - c_k \, \vec{\Delta}_k \tag{16}$$

$$\vec{g}_k = \frac{y(\vec{\theta}_k^+) - y(\vec{\theta}_k^-)}{2\, c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{17}$$

$$\vec{\theta}_{k+1} = \vec{\theta}_k - a_k \, \vec{g}_k \tag{18}$$

To apply the SPSA algorithm to parameter optimization for query expansion we normalized every parameter domain to the interval $[0, 1]$ by linear transformation and used the negative MAP of retrieval runs as loss function. Prior to evaluating the loss function, the inverse linear transform needs to be applied to normalized parameter values, followed by rounding for originally integer-valued parameters. Normalized parameter values were clipped to the $[0, 1]$ range when applying the update step (18). The algorithm terminates when $y(\vec{\theta}_k^+)$ and $y(\vec{\theta}_k^-)$ differ by less than ε for K successive iterations, or when a maximal iteration count n is reached. The SPSA parameter values used in our experiments are shown in Table 6.

To determine the result $\vec{\theta}_{\min}$ of optimization we consider all parameter vectors $\vec{\theta}_k^+$ and $\vec{\theta}_k^-$ as well as the initial vector $\vec{\theta}_1$ and the final vector $\vec{\theta}_{n+1}$ when the algorithm terminates after n iterations. The most recently computed one of these parameter vectors with minimal loss value is selected as $\vec{\theta}_{\min}$.

$$\vec{\theta}_{\min} = \underset{1 \le k \le n}{\operatorname{argmin}} \left\{ y(\vec{\theta}_1),\ y(\vec{\theta}_k^+),\ y(\vec{\theta}_k^-),\ y(\vec{\theta}_{n+1}) \right\} \tag{19}$$

5 Experiments

5.1 Dataset and Cross-Validation

The information retrieval methods described in Section 3 were evaluated on the MCR dataset of ImageCLEF 2013 [28]. It consists of 74,654 full-text publications in English language drawn from the MEDLINE[14] database of biomedical literature and freely available in the PubMed Central[15] repository. Publications are provided as XML documents with separate fields for title, abstract, image captions, and fulltext (see Figure 4). Additionally, the dataset includes about 300,000 image files referenced by documents, but they were ignored in our experiments.

Most of MEDLINE publication records are annotated with MeSH terms, which can be retrieved using the Entrez search system API[16] [25]. We were able to retrieve MeSH terms for 73,584 documents (98.6%) of the MCR dataset. They have been used as *manually annotated MeSH terms* in our experiments.

The ImageCLEF 2013 MCR dataset comes with 35 query topics represented in XML (see Figure 5). Each topic consists of a few English sentences describing patients' symptoms, and one or more diagnostic images, which were again ignored in our experiments. Relevance judgments have been produced by medical experts for pooled results submitted by ImageCLEF 2013 participants [28], according to common practice in TREC-type retrieval evaluation [81]. Relevance judgments were then published[17] by the organizers of ImageCLEF 2013 medical tasks. Table 7 gives some statistical information.

ImageCLEF 2013 participants had access to the 2012 MCR dataset and relevance judgments. In 2012, the same document collection had been used as in 2013, but there were only 26 query topics, which were re-used in 2013. That is, the 2013 dataset added another 9 query topics. However, relevance judgments of the 2012 dataset failed to provide any relevant documents for 3 topics, so we removed them for our cross-validation experiments. This corrected set of 23 queries is denoted by *2012corr* (Table 7).

In order to assess the robustness of retrieval methods with respect to parameter optimization, we divided the 2013 query set into 5 subsets of equal size and used 4 subsets

[14] `http://www.nlm.nih.gov/pubs/factsheets/medline.html`
[15] `http://www.ncbi.nlm.nih.gov/pmc/`
[16] `https://www.ncbi.nlm.nih.gov/books/NBK21081/`
[17] `http://www.imageclef.org/2013/medical`. The dataset is available after registration only.

```
−<article pmcid="100321" pmid="11882251" doi="10.1186/1471-2105-3-9"
 pmc-article-url="http://www.ncbi.nlm.nih.gov/pmc/articles/PMC100321" original-
 article-url="">
  −<title>
     An algorithm and program for finding sequence specific oligo-nucleotide probes for
     species identification
   </title>
  −<authors>
     <author>Pozhitkov, Alexander E</author>
     <author>Tautz, Diethard</author>
   </authors>
  −<abstract>
     Background The identification of species or species groups with specific oligo-
     nucleotides as molecular signatures is becoming increasingly popular for bacterial
     samples. ...
   </abstract>
  −<fulltext>
     Background Identification of species with molecular probes is likely to revolutionize
     taxonomy, at least for taxa with morphological characters that are difficult to
     determine otherwise. ...
   </fulltext>
  −<figures>
    −<figure iri="1471-2105-3-9-1">
      −<caption>
         Scheme of the probe finding algorithm. Details are explained in the text.
       </caption>
     </figure>
    −<figure iri="1471-2105-3-9-3">
      −<caption>
         Comparison of specific oligos suggested by ARB and PROBE for Thermotoga
         maritima, ...
       </caption>
     </figure>
   </figures>
 </article>
```

Figure 4: Sample XML document of ImageCLEF 2013 MCR dataset.

```
−<TOPIC>
    <ID>1</ID>
    <TYPE>case-based</TYPE>
  −<EN-DESCRIPTION>
     A 43-year-old man with painless, gross hematuria. Abdominal CT scan revealed a
     large left renal mass with extension into the left renal pelvis and ureter.
    </EN-DESCRIPTION>
    <image>CaseQueryImages2012/1_1.jpg</image>
    <image>CaseQueryImages2012/1_2.jpg</image>
    <image>CaseQueryImages2012/1_3.gif</image>
 </TOPIC>
```

Figure 5: Sample query topic of ImageCLEF 2013 MCR dataset.

Table 7: Summary statistics of relevance judgments (RJ) for ImageCLEF MCR datasets used in our experiments. Q_k denotes the k-th quartile, Q_2 is the median.

Property	**2012corr dataset**	**2013 dataset**
number of queries	23	35
number of RJ	12,327	15,028
RJ per query (min/avg/max)	394/536/589	372/429/480
relevant documents per query ($\min/Q_1/Q_2/Q_3/\max$)	2/4/6/17/46	1/3/10/33/101

for parameter optimization and the remaining subset for testing. This procedure was repeated 4 times such that each subset was used once for testing, and finally the retrieval metric (MAP) was averaged over the 5 test runs. This evaluation method is known as *5-fold cross-validation* [12, 73].

5.2 Evaluated Method Combinations

The proposed query and document expansion methods described in Section 3 are listed in Table 8, together with their acronyms used to identify method combinations. Every MeSH query expansion method uses both a MeSH matching algorithm and a synonym handling method, amounting to $5 * 4 = 20$ MeSH query expansion methods. The other two method groups, pseudo-relevance feedback and document expansion, consist of single alternative methods, resulting in 8 and 5 methods, respectively. To compute the total number of possible method combinations, we need to take into account that every method combination must include either fulltext search or MeSH query expansion ($1 + 20 = 21$ possibilities), and that a pseudo-relevance feedback or document expansion method may be used or not (resulting in $8 + 1 = 9$ and $5 + 1 = 6$ possibilities, respectively). Thus, the total number of proposed query and document expansion method combinations is $21 * 9 * 6 = 1134$.

To evaluate a single method combination by applying the 5-fold cross-validation approach described in the previous section, we need to optimize parameters on each of five validation sets and evaluate on five test sets. An optimization run is limited to 20 iterations, each computing mean average precision (MAP) on all queries in the validation set twice (with different parameter settings). Evaluation on the test set requires computing a single MAP value. We end up with a maximum of $5 * (20 + 1) = 105$ MAP computations (and the associated retrieval runs) to evaluate a single method combination.

To reduce overall computation time[18] and to simplify analysis and presentation of results, we chose to evaluate only "interesting" method combinations, not all possible ones. Preliminary experiments showed that methods employing pseudo-relevance feedback gave

[18]Evaluating 546 method combinations concurrently on a 24-core machine with 96 GB RAM took about 36 hours.

Table 8: Query and document expansion methods proposed in Section 3, divided into four classes (typeset in *italics*). The *Count* column gives the number of different methods corresponding to each line. The combination rules leading to a total number of 1134 method combinations are explained in the main text.

Acronym	**Method**	**Count**
F	*fulltext search* (no MeSH query expansion)	1
M	*MeSH query expansion*	20
tN	MeSH term matching algorithm, $0 \leq N \leq 4$	5
xN	synonym selection method, $0 \leq N \leq 3$	4
r*	*pseudo-relevance feedback*	8
r	unigrams ranked by TF-IDF	1
r2	unigrams and bigrams ranked by TF-IDF	1
rm	manually annotated MeSH terms	1
rm2	union of r and rm features	1
raN	automatically annotated MeSH terms ranked by score tN, $1 \leq N \leq 4$	4
+*	*document expansion*	5
+	manually annotated MeSH terms	1
+N	automatically annotated MeSH terms ranked by score tN, $1 \leq N \leq 4$	4

clearly better results than other method combinations, so we emphasized feedback methods when selecting combinations for evaluation. Moreover, we were interested in MeSH query expansion alone, and in combinations of document expansion with feedback methods. The selected set of 546 method combinations is presented in Table 9, grouped by the number of parameters that need to be optimized. There are two method groups with 5 parameters, because they use different parameter sets (cf. Table 4). The acronym *raN+N* denotes all method combinations using pseudo-relevance feedback of automatically annotated MeSH terms (*raN*) and document expansion (*+N*) using the *same* method N ($1 \leq N \leq 4$) for automatic MeSH term annotation (cf. Table 8). We expect that these combinations perform better than cross-combinations *raN+K* with $N \neq K$, because MeSH terms chosen for query expansion from pseudo-relevant documents are more likely to be found in expanded documents if MeSH terms of both expansions have been generated by the same algorithm.

For purposes of presentation and analysis of results, selected method combinations have been arranged into eight groups corresponding to combinations of three classes of techniques: MeSH query expansion (M), pseudo-relevance feedback (r*), and document expansion (+*). These groups are listed in Table 10.

Table 9: Combinations of query and document expansion methods selected for evaluation and grouped by number of parameters to be optimized (cf. Table 4). Acronyms of methods are given in Table 8.

Combinations	**Parameters**	**Count**
F, F+	0	2
M, M+	2	40
Fr, Frm, FraN, Fr+, Frm+, FraN+N	3	12
Frm2, Frm2+*	4	6
Fr2, Fr2+*	5	6
Mr, Mrm, MraN, Mr+, Mrm+, MraN+N	5	240
Mrm2, Mrm2+*	6	120
Mr2, Mr2+*	7	120
Total count		546

Table 10: Query and document expansion methods selected for evaluation and grouped by combination of techniques (MeSH query expansion, pseudo-relevance feedback, and document expansion). The 546 individual method combinations are the same as in Table 9.

Acronym	**Group of methods**	**Count**
F	fulltext search (without query expansion)	1
M	MeSH query expansion	20
F+	fulltext search with document expansion (manual MeSH annotation)	1
M+	MeSH query expansion with document expansion (manual MeSH annotation)	20
Fr*	fulltext search with pseudo-relevance feedback	8
Mr*	MeSH query expansion followed by pseudo-relevance feedback	160
Fr*+*	fulltext search with pseudo-relevance feedback and document expansion Fr+, Frm+, FraN+N, Frm2+*, Fr2+*	16
Mr*+*	MeSH query expansion followed by pseudo-relevance feedback with document expansion Mr+, Mrm+, MraN+N, Mrm2+*, Mr2+*	320
Total count		546

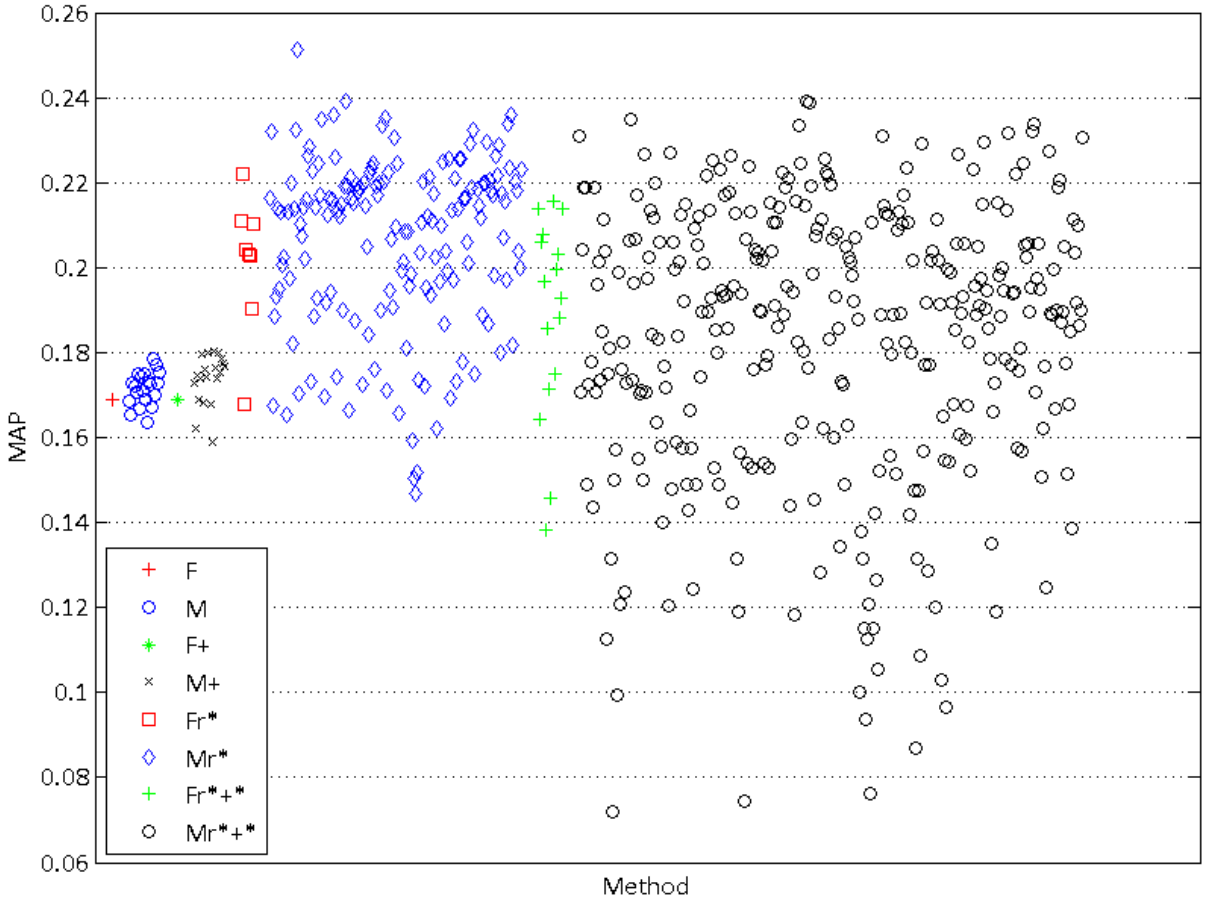


Figure 6: Scatter plot of 546 combinations of query and document expansion methods with optimized parameters obtained by 5-fold cross validation on the ImageCLEF 2013 MCR dataset. Method combinations are grouped according to Table 10.

5.3 Cross-validation Results

We evaluated the selected 546 combinations of query and document expansion methods by 5-fold cross-validation on the ImageCLEF 2013 MCR dataset, as explained in the previous sections. As retrieval performance metric we used *mean average precision* (MAP), which is commonly applied to TREC-style evaluations [7]. Note that the same metric served as objective function for parameter optimization (cf. Section 4).

Figure 6 presents a scatter plot of obtained results, grouped by the eight classes of method combinations listed in Table 10. Every data point represents the final MAP value of one method combination X, i.e. the average over five test runs, where each test run corresponds to parameter settings optimized independently for X on one of the five validation sets.

The two best method combinations of each group are listed in Table 11, revealing the actual algorithms employed. In particular, the overall best method combination was Mt2x0r2, which used MeSH term matching algorithm **BinDist** (t2) with direct synonym

Table 11: Best and second-to-best combinations of query and document expansion methods depicted in Figure 6. Best MAP values of each column are marked in boldface.

Group	Best Method	MAP	Second Method	MAP
F	F	0.1689	–	–
M	Mt0x3	0.1784	Mt2x3	0.1771
F+	F+	0.1688	–	–
M+*	Mt2x2+	0.1802	Mt0x3+	0.1801
Fr*	Fr2	0.2219	Fr	0.2109
Mr*	Mt2x0r2	**0.2511**	Mt1x1r	**0.2390**
Fr*+*	Frm2+	0.2155	Fr2+	0.2139
Mr*+*	Mt4x1r2+	0.2393	Mt4x1r2+2	0.2389

handling (x0) for MeSH query expansion, followed by pseudo-relevance feedback with unigrams and bigrams (r2) to further expand the query. Refer to Table 8 and Section 3 to interpret acronyms of other method combinations.

5.3.1 Comparison of Feedback Methods

As all method combinations exceeding 0.2 MAP employ pseudo-relevance feedback, we would like to know if some feedback methods are consistently better than others within a given group of combinations. We focused on the best performing group Mr* and grouped their methods by employed pseudo-relevance feedback algorithm. The scatter plot (Figure 7) reveals that point clouds pertaining to different feedback algorithms form clusters with rather small intra-class variance (with respect to MAP), and some classes clearly perform better than others, indicated by large inter-class distances. In particular, feedback methods ranking unigrams (words) of pseudo-relevant documents by TF-IDF, namely methods r, r2, and rm2, perform consistently better than other feedback methods. Although the overall best method combination uses unigrams and bigrams for feedback (r2), this feedback method cannot be claimed to be better than feedback using unigrams only (r), because the best data point appears to be an outlier in the group of tested r2 method combinations.

Another interesting conclusion drawn from Figure 7 is that method combinations using manually annotated MeSH terms for feedback consistently perform worse than feedback methods ra2, ra3, and ra4, which all use automatically annotated MeSH terms for feedback. This may be unexpected to some extent, because manually annotated MeSH terms are assumed to be more accurately related to document semantics than automatically annotated ones and hence should provide a more effective data source for query expansion. But in the light of successful feedback methods using words from pseudo-relevant documents directly (r, r2, and rm2 methods), the relatively better performance of ra2, ra3, and ra4 methods becomes intelligible, as they basically extract MeSH terms already present in documents.

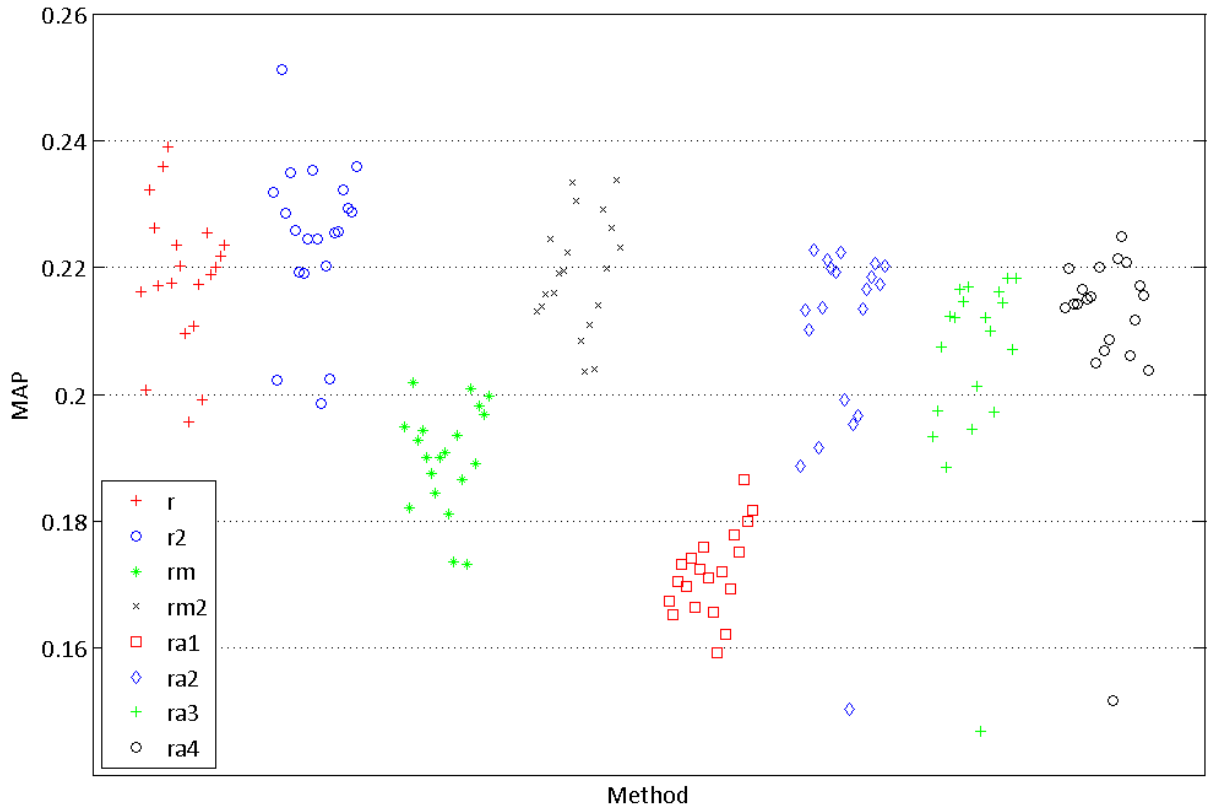


Figure 7: Scatter plot of 160 query expansion methods employing MeSH query expansion followed by pseudo-relevance feedback, grouped by feedback method. The data points correspond to the Mr* group of Figure 6. Acronyms of feedback methods are explained in Table 8.

Method combinations employing feedback by MeSH terms extracted using distance-based match frequency (**Dist** MeSH term matching, see Section 3.2) perform consistently worse than ra2, ra3, and ra4 methods. This is a strong indication that the concept of matching runs (used by ra2, ra3, and ra4 methods) is important to apply the proposed MeSH term matching approach to longer documents. The **Dist** scoring function may assign a high score to a MeSH term for a document just because words of the MeSH term occur sufficiently often in the document, although not all words of the MeSH term are present or they occur in distant locations in the document.

5.3.2 MeSH Query Expansion Methods

The 20 tested MeSH query expansion methods use different MeSH term matching algorithms (**t0** – **t4**) and synonym handling methods (**x0** – **x3**). The data points of group M in Figure 6 suggest that the effect of MeSH query expansion methods on retrieval performance is small. The comparison of different MeSH term matching and synonym handling methods is therefore likely to give no clear results, but is pursued here in the interest of completeness.

Let us look at scatterplots of method combinations M and Mr*, grouped by MeSH term matching algorithms. Although the plot for group M (Figure 8) suggests that the **BinDist** algorithm (t2) performs better than **Dist** (t1), the difference in terms of MAP is small enough to be swallowed by the dominating variance of feedback methods in group Mr* (Figure 9). A similar observation can be made for scatterplots grouped by MeSH synonym handling methods (Figures 10 and 11).

5.3.3 Document Expansion Methods

When comparing the point clouds of method groups Mr* and Mr*+* in Figure 6, it is obvious that document expansion (used by Mr*+* methods) did not improve retrieval performance in our experiments. It even deteriorated results substantially for many method combinations. However, for the sake of comparing the usefulness of our automatic MeSH annotation algorithms with that of manual MeSH annotations, it may be interesting to take a closer look at the performance of different tested document expansion methods.

Figure 12 presents all data points corresponding to query expansion methods using pseudo-relevance feedback combined with document expansion (groups Fr*+* and Mr*+* of Figure 6), grouped by document expansion method. In contrast to their use for pseudo-relevance feedback (see Section 5.3.1), but as expected, manually annotated MeSH terms perform consistently better than automatically annotated ones for document expansion. However, MeSH terms annotated by the **BinDist** algorithm (t2) yield a comparable performance for many method combinations, including the top-performing ones. Somewhat disappointing is the fact that the more sophisticated MeSH term matching algorithms **IdfBinDist** (t3) and **IdfCovDist** (t4) did not improve retrieval performance over **BinDist** (t2), although they have been designed to give more meaningful MeSH annotations.

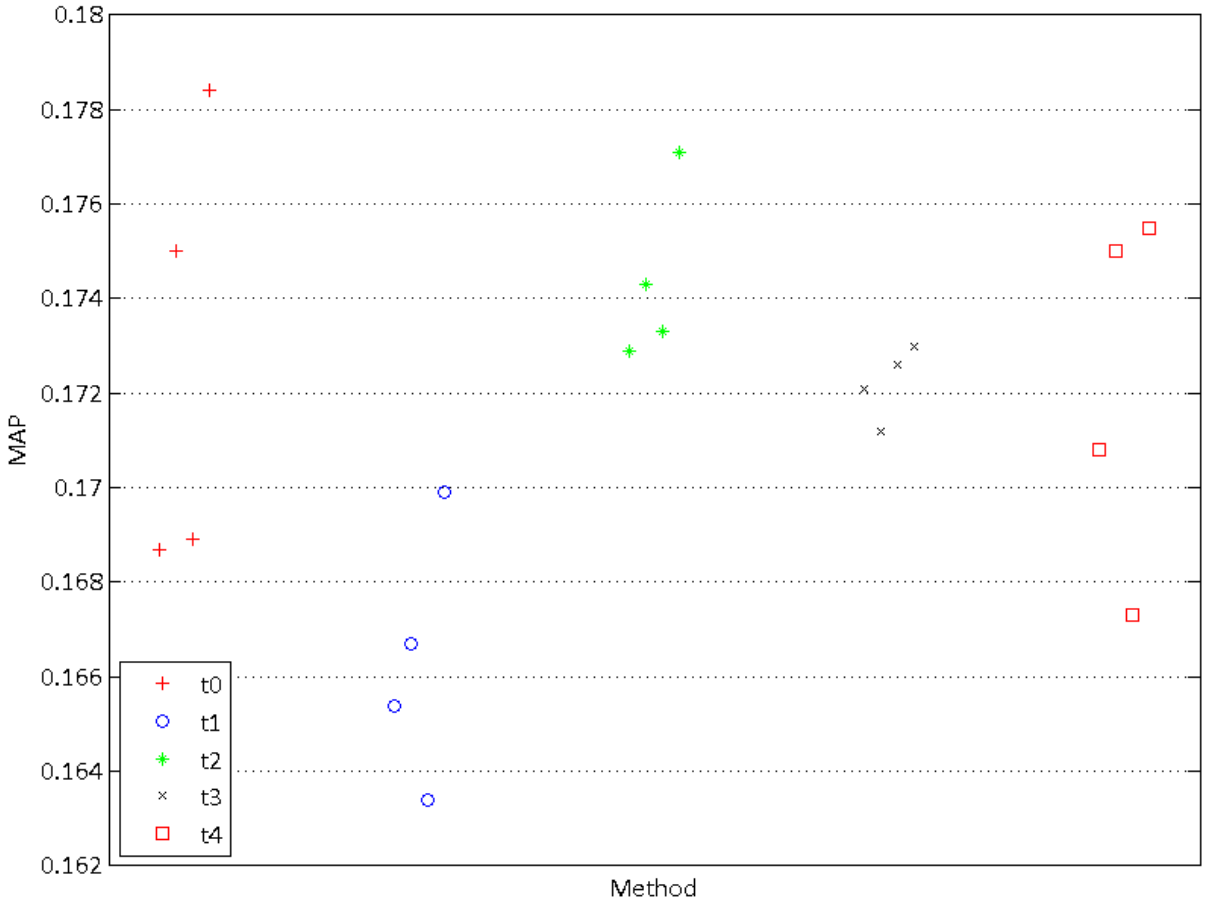


Figure 8: Scatter plot of MeSH query expansion methods, grouped by MeSH term matching algorithm. The data points correspond to group M of Figure 6. MeSH term matching algorithms (t0 – t4) are explained in Section 3.2.

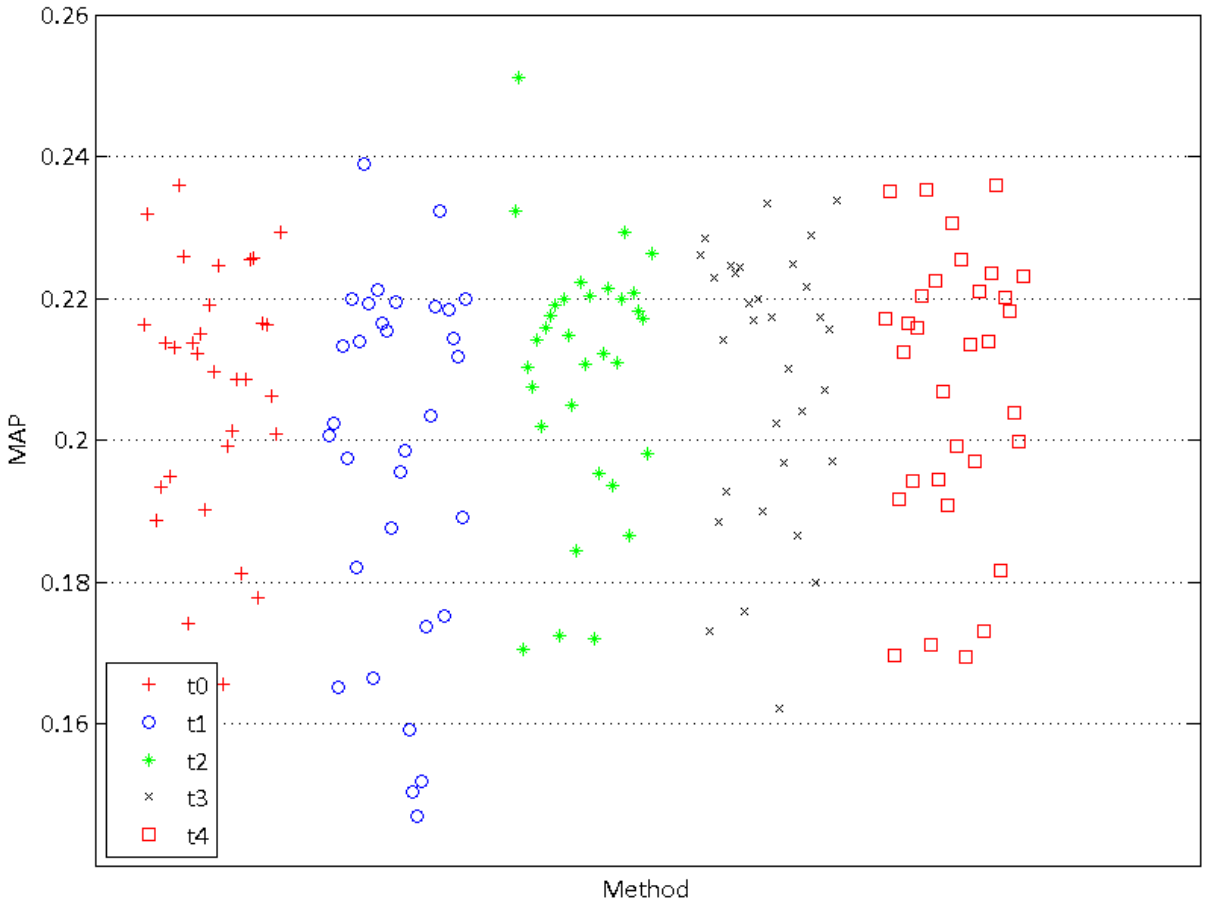


Figure 9: Scatter plot of MeSH query expansion methods combined with pseudo-relevance feedback, grouped by MeSH term matching algorithm. The data points correspond to group Mr* of Figure 6. MeSH term matching algorithms (t0 – t4) are explained in Section 3.2.

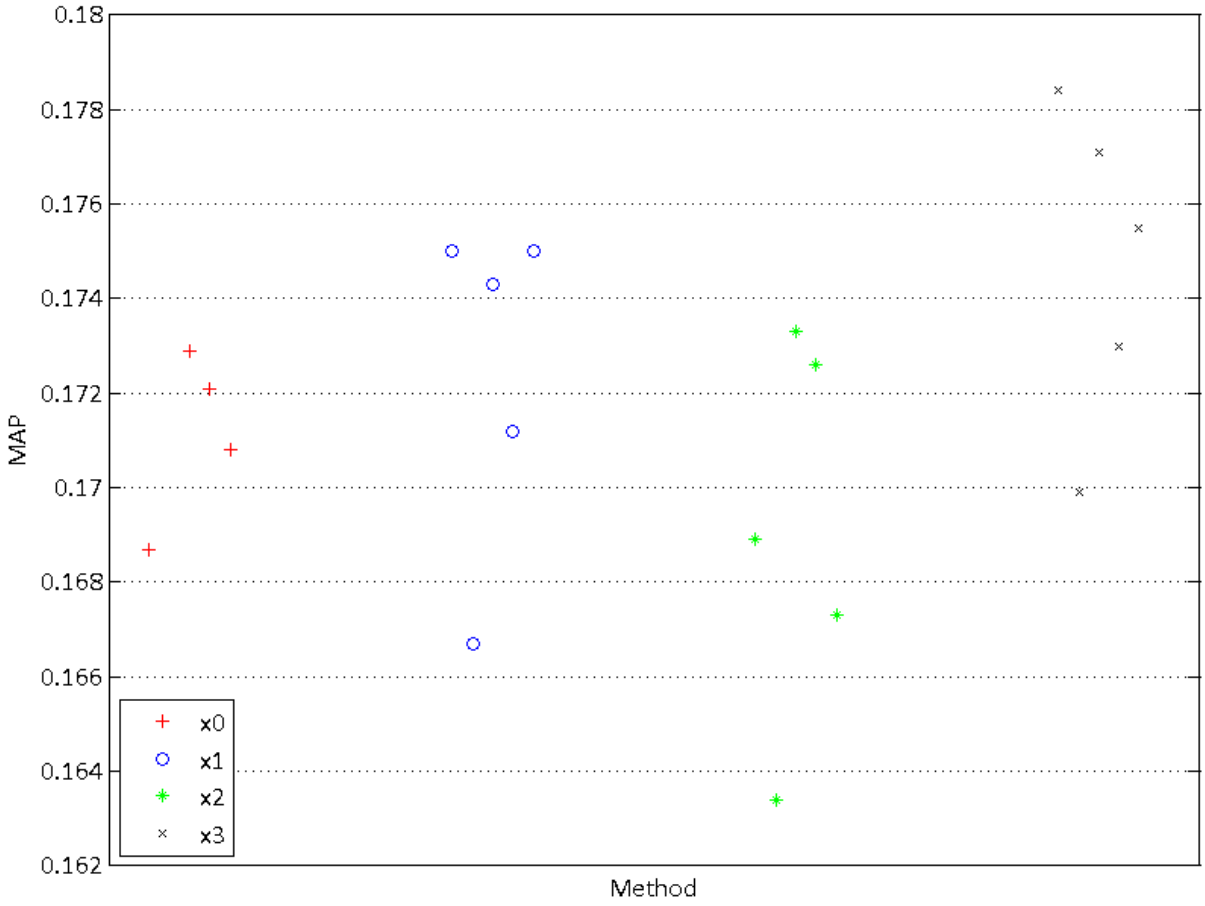


Figure 10: Scatter plot of MeSH query expansion methods, grouped by MeSH synonym handling method. The data points correspond to group M of Figure 6. MeSH synonym handling methods (x0 – x3) are explained in Section 3.3.1.

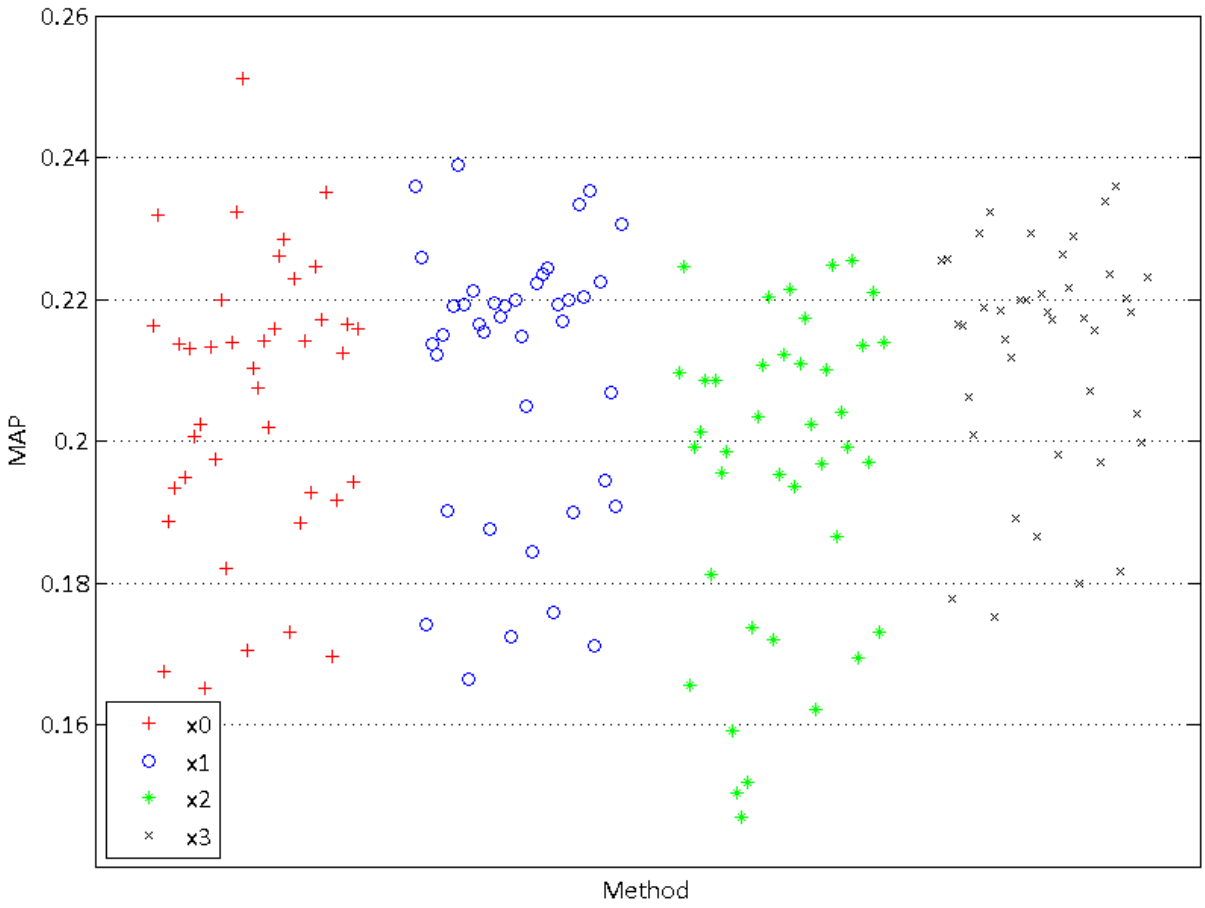


Figure 11: Scatter plot of MeSH query expansion methods combined with pseudo-relevance feedback, grouped by MeSH synonym handling method. The data points correspond to group Mr* of Figure 6. MeSH synonym handling methods (x0 – x3) are explained in Section 3.3.1.

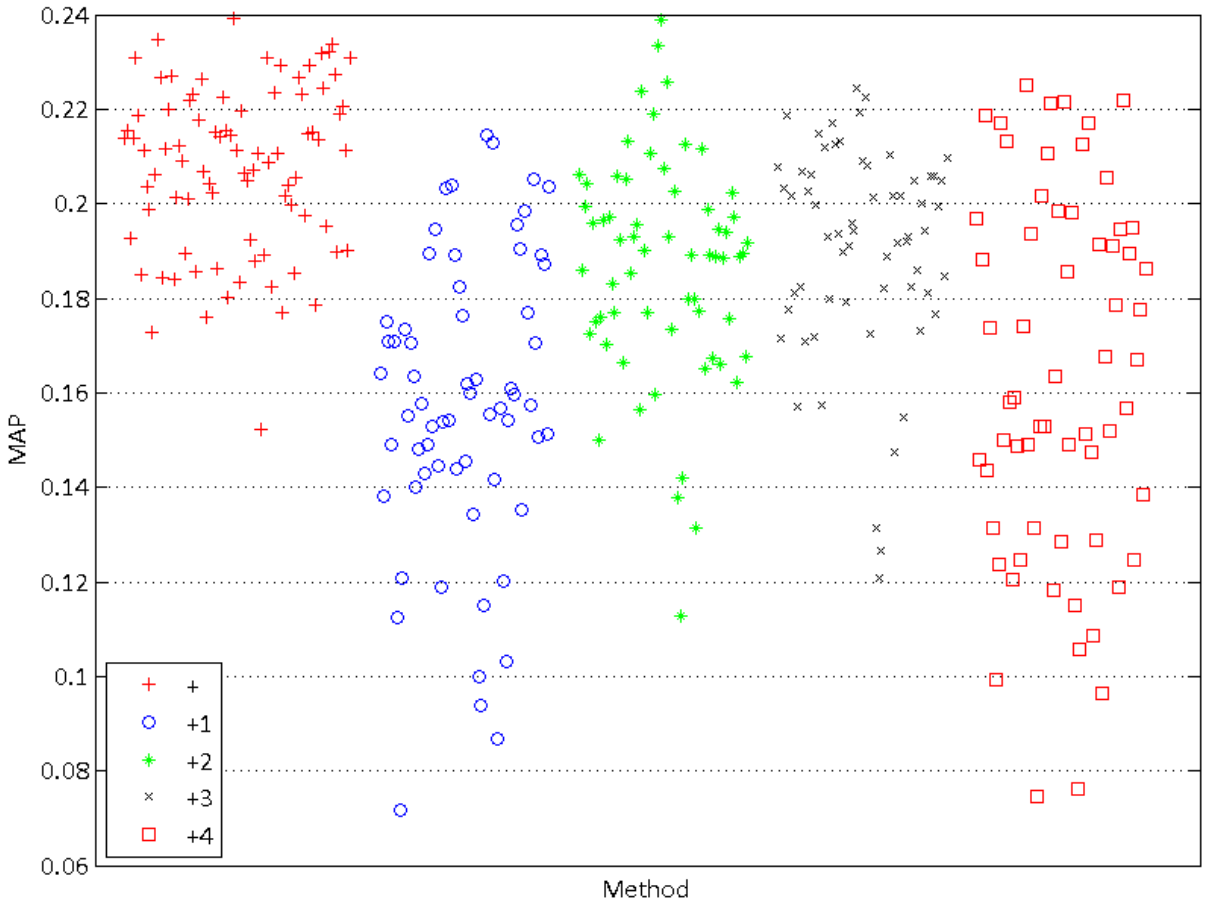


Figure 12: Scatter plot of query expansion methods using pseudo-relevance feedback combined with document expansion, grouped by document expansion method. The data points correspond to groups Fr*+* and Mr*+* of Figure 6. Document expansion methods (+, +1, +2, +3, +4) are explained in Section 3.4.

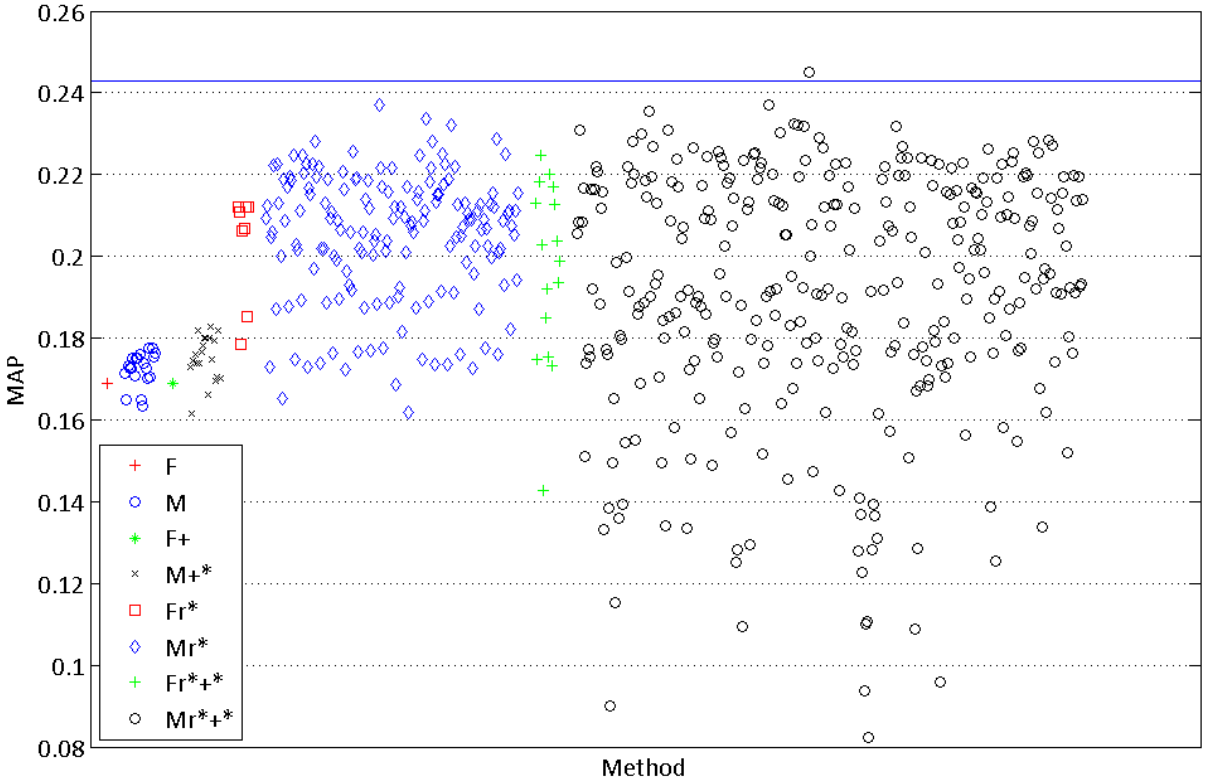


Figure 13: Scatter plot of 546 combinations of query and document expansion methods with parameters optimized on corrected ImageCLEF 2012 dataset and evaluated on the ImageCLEF 2013 MCR dataset. Method combinations are grouped according to acronyms listed in the legend, which are explained in Table 8. The horizontal line at MAP 0.2429 corresponds to the best run submitted to ImageCLEF 2013 [28].

5.4 ImageCLEF Evaluation Results

In order to compare the methods evaluated here to retrieval runs originally submitted to the ImageCLEF 2013 MCR task [28], we repeated the experiments described in the previous section using the official ImageCLEF 2013 MCR evaluation technique. The corrected ImageCLEF 2012 dataset (same document collection as 2013, but only 23 queries with different relevance judgments, cf. Table 7) was used as validation set for parameter optimization, because the relevance judgments of the 2013 dataset were not available to participants before submission. Retrieval performance was then evaluated for each optimized method combination on the entire 2013 dataset (35 queries, a superset of the 2012 queries). No cross-validation technique was involved. As in the previous section, we present results for the mean average precision (MAP) metric.

In analogy to the previous section, a scatter plot of all 546 tested method combinations is shown in Figure 13. The details of the best two method combinations of each method group are listed in Table 12. The best MCR run submitted to ImageCLEF 2013 achieved 0.2429 MAP using an external corpus of 22 million MEDLINE citations to generate MeSH

Table 12: Best and second-to-best combinations of query and document expansion methods, optimized on corrected ImageCLEF 2012 dataset and evaluated on ImageCLEF 2013 MCR dataset. Acronyms of method combinations are explained in Table 8. Best MAP values of each column are marked in boldface.

Group	Best Method	MAP	Second Method	MAP
F	F	0.1689	–	–
M	Mt0x3	0.1774	Mt2x3	0.1774
F+	F+	0.1688	–	–
M+*	Mt3x2+	0.1827	Mt0x1+	0.1820
Fr*	Fra4	0.2122	Fr	0.2121
Mr*	Mt3x1rm2	0.2369	Mt2x2ra4	0.2335
Fr*+*	Fr2+3	0.2247	Frm2+	0.2201
Mr*+*	Mt4x1r2+2	**0.2450**	Mt2x1rm2+	**0.2370**

terms for query expansion by local feedback [21]. This run is indicated by a horizontal line in Figure 13. Although all our methods rely on the dataset corpus only, one method combination (Mt4x1r2+2) achieved an even better result.

When comparing the scatter plots of Figures 6 and 13, they give a very similar picture of the performance of different method groups. Even the absolute MAP values achieved by the vast majority of method combinations within a group coincide (e.g. most Mr* achieve a MAP between 0.16 and 0.24 for both evaluation techniques). Outliers, however, both in high- and low-performing ranges, differ remarkably in several method groups (Fr*, Mr*, and Mr*+*). We attribute that to randomness inherent to parameter optimization, limiting the robustness of affected method groups.

Based on the correspondence between ImageCLEF-type evaluation and that based on cross-validation, the main findings of Section 5.3 remain valid, and we do not repeat the analysis here. In particular, query expansion methods employing MeSH query expansion followed by pseudo-relevance feedback (group Mr*) seem to be the best choice (out of tested methods), and combining them with document expansion (group Mr*+*) has no further benefit.

6 Conclusion

This work investigated the benefit of selected known query expansion and document expansion techniques to textual methods for medical case retrieval (MCR). We proposed new algorithms to automatically map queries or documents to Medical Subject Headings (MeSH), a thesaurus of biomedical terms, and used these MeSH terms for query and document expansion. Additionally, query-specific local feedback methods based on Rocchio's pseudo-relevance feedback were used to determine expansion terms from top-retrieved documents. Several variants of these query and document expansion methods were combined in different ways and evaluated on the ImageCLEF 2013 MCR dataset.

More precisely, 546 method combinations were evaluated independently by 5-fold cross-validation to avoid overfitting by parameter optimization. Another set of experiments applied the official ImageCLEF 2013 MCR evaluation procedure to these method combinations to allow for comparison with retrieval runs submitted to ImageCLEF 2013.

Experimental results show that query expansion methods using MeSH terms derived from the query (MeSH query expansion) and local feedback can substantially improve MCR performance over fulltext-only retrieval. The improvement is mainly due to local feedback using unigrams (words) and bigrams from pseudo-relevant documents, local feedback by MeSH terms is less effective. However, combining MeSH query expansion with local feedback may result in a higher performance gain (in terms of mean average precision) than combining it with fulltext-only retrieval.

On the other hand, combining MeSH query expansion and/or local feedback with document expansion does not improve retrieval performance. There is no consistent best method within the set of proposed MeSH term matching algorithms and MeSH synonym handling methods used for query and document expansion.

The contributions of this work include (1) the design of novel efficient algorithms to associate queries or documents with MeSH terms, that do not rely on natural language processing or machine learning; and (2) a comprehensive evaluation of query and document expansion methods based on MeSH terms and pseudo-relevance feedback that achieve state-of-the-art retrieval performance on the ImageCLEF 2013 MCR dataset.

Although care has been taken to avoid overfitting effects when performing experiments, the generalization power of results is still limited by the facts that (1) evaluation is based on a single dataset, and (2) results depend on the effectiveness of parameter optimization. So further work could improve evaluation by searching for or developing a second dataset, and by cross-validating parameter optimization using a different (e.g. genetic) algorithm. The proposed methods for automatic MeSH annotation of documents could be evaluated separately by comparing them to manual MeSH annotation.

Other promising avenues for future work on textual MCR techniques include utilizing document structure (title, abstract, image captions), applying more sophisticated query expansion methods (cf. Section 2.2.2), or using external corpora or text categorization based on machine learning [73] to expand queries or annotate documents with additional biomedical terms.

References

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[2] S. Abdou and J. Savoy. Searching in Medline: Query expansion and manual indexing evaluation. *Inf. Process. Manage.*, 44(2):781–789, Mar. 2008.

[3] E. Agirre, G. M. D. Nunzio, T. Mandl, and A. Otegi. CLEF 2009 ad hoc track overview: Robust-WSD task. LNCS 6241, pages 36–49. Springer, 2010.

[4] G. Amati, C. Carpineto, and G. Romano. Comparing weighting models for monolingual information retrieval. In *Comparative Evaluation of Multilingual Information Access Systems, Proc. 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003*, pages 310–318. Springer, 2004.

[5] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, Oct. 2002.

[6] J. Arguello, J. L. Elsas, J. Callan, and J. G. Carbonell. Document representation and query expansion models for blog recommendation. In E. Adar, M. Hurst, T. Finin, N. Glance, N. Nicolov, and B. Tseng, editors, *Proc. 2nd Int. Conf. Weblogs and Social Media*, pages 10–18. AAAI Press, 2008.

[7] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, USA, 2nd edition, 2011.

[8] H. Bast, D. Majumdar, and I. Weber. Efficient interactive query expansion with complete search. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 857–860, New York, NY, USA, 2007. ACM.

[9] M. Batet, D. Sánchez, and A. Valls. An ontology-based measure to compute semantic similarity in biomedicine. *Journal of Biomedical Informatics*, 44(1):118–125, 2011.

[10] S. Begum, M. Ahmed, P. Funk, N. Xiong, and M. Folke. Case-based reasoning systems in the health sciences: A survey of recent trends and developments. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(4):421–434, July 2011.

[11] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Inf. Process. Manag.*, 43(4):866–886, July 2007.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl. 1):D267–D270, 2004.

[14] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM.

[15] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27, Jan. 2001.

[16] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):17:1–17:38, July 2009.

[17] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, Jan. 2012.

[18] C. Carpineto, G. Romano, and V. Giannini. Improving retrieval feedback with multiple term-ranking function combination. *ACM Trans. Inf. Syst.*, 20(3):259–290, July 2002.

[19] Y. Chang, I. Ounis, and M. Kim. Query reformulation using automatically generated query concepts from a document space. *Inf. Process. Manage.*, 42(2):453–468, Mar. 2006.

[20] P. A. Chirita, C. S. Firan, and W. Nejdl. Personalized query expansion for the web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 7–14, New York, NY, USA, 2007. ACM.

[21] S. Choi, J. Lee, and J. Choi. SNUMedinfo at ImageCLEF 2013: Medical retrieval task. In P. Forner, R. Navigli, and D. Tufis, editors, *CLEF 2013 Evaluation Labs and Workshop, Online Working Notes*, 2013.

[22] K. Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 837–846, New York, NY, USA, 2009. ACM.

[23] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 704–711, New York, NY, USA, 2005. ACM.

[24] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 303–310, New York, NY, USA, 2007. ACM.

[25] N. R. Coordinators. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 41(D1):D8–D20, 2013.

[26] M. Crespo, J. Mata, and M. Maña. Improving image retrieval effectiveness via query expansion using MeSH hierarchical structure. *Journal of the American Medical Informatics Association [online]*, Sept. 2012. DOI 10.1136/amiajnl-2012-000943.

[27] C. J. Crouch and B. Yang. Experiments in automatic statistical thesaurus construction. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 77–88, New York, NY, USA, 1992. ACM.

[28] A. G. S. de Herrera, J. Kalpathy-Cramer, D. Demner-Fushman, S. Antani, and H. Müller. Overview of the ImageCLEF 2013 medical tasks. In *Working notes of CLEF 2013*, Valencia, Spain, 2013.

[29] A. P. Dempster, N. M. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc.*, 39(1):1–38, 1977.

[30] M. C. Díaz-Galiano, M. Martín-Valdivia, and L. A. Ureña López. Query expansion with a medical ontology to improve a multimodal information retrieval system. *Comput. Biol. Med.*, 39(4):396–403, Apr. 2009.

[31] T. E. Doszkocs. AID, an associative interactive dictionary for online searching. *Online Information Review*, 2(2):163–173, 1978.

[32] Z. Gong, C. Cheang, and L. Hou U. Multi-term web query expansion using WordNet. In S. Bressan, J. Küng, and R. Wagner, editors, *Database and Expert Systems Applications*, volume 4080 of *Lecture Notes in Computer Science*, pages 379–388. Springer Berlin Heidelberg, 2006.

[33] J. Gonzalo, F. Verdejo, I. Chugur, and J. M. Cigarrän. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 647–678. Association for Computational Linguistics, 1998.

[34] J. Graupmann, J. Cai, and R. Schenkel. Automatic query refinement using mined semantic relations. In *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration*, WIRI '05, pages 205–213, Washington, DC, USA, 2005. IEEE Computer Society.

[35] J. Hu, W. Deng, and J. Guo. Improving retrieval performance by global analysis. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 02*, ICPR '06, pages 703–706, Washington, DC, USA, 2006. IEEE Computer Society.

[36] D. A. Hull. Stemming algorithms: A case study for detailed evaluation. *J. Am. Soc. Inf. Sci.*, 47(1):70–84, Jan. 1996.

[37] F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Proceedings of the Workshop on Pattern Recognition in Practice, Amsterdam, May 21-23, 1980*, pages 381–397. North-Holland, 1980.

[38] S. Jones, M. Gatford, S. Robertson, M. Hancock-Beaulieu, J. Secker, and S. Walker. Interactive thesaurus navigation: Intelligence rules ok? *Journal of the American Society for Information Science*, 46(1):52–59, 1995.

[39] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *British Medical Journal*, 330(7494):765, 2005.

[40] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 191–202, New York, NY, USA, 1993. ACM.

[41] A. M. Lam-Adesina and G. J. F. Jones. Applying summarization techniques for term selection in relevance feedback. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 1–9, New York, NY, USA, 2001. ACM.

[42] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[43] S. Liu, F. Liu, C. Yu, and W. Meng. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 266–272, New York, NY, USA, 2004. ACM.

[44] X. Liu and W. B. Croft. Statistical language modeling for information retrieval. *Annual Review of Information Science and Technology*, 39(1):1–31, 2005.

[45] Y. Liu, C. Li, P. Zhang, and Z. Xiong. A query expansion algorithm based on phrases semantic similarity. In *Proceedings of the 2008 International Symposiums on Information Processing*, ISIP '08, pages 31–35, Washington, DC, USA, 2008. IEEE Computer Society.

[46] Z. Lu, W. Kim, and W. J. Wilbur. Evaluation of query expansion using MeSH in PubMed. *Inf. Retr.*, 12(1):69–80, Feb. 2009.

[47] Y. Lv and C. Zhai. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 255–264, New York, NY, USA, 2009. ACM.

[48] R. Mandala, T. Takenobu, and T. Hozumi. The use of WordNet in information retrieval. In *Proceedings of the ACL Workshop on the Usage of WordNet in Information Retrieval*, pages 31–37. Association for Computational Linguistics, 1998.

[49] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[50] J. Mata, M. Crespo, and M. J. Maña. Using MeSH to expand queries in medical image retrieval. In *Proc. MICCAI, Medical Content-Based Retrieval for Clinical Decision Support*, MCBR-CDS'11, pages 36–46. Springer, 2012.

[51] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing & Management*, 40(5):735 – 750, 2004. Special Issue on Bayesian Networks and Information Retrieval.

[52] D. Metzler and W. B. Croft. Latent concept expansion using Markov random fields. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 311–318, New York, NY, USA, 2007. ACM.

[53] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[54] J. Minker, G. A. Wilson, and B. H. Zimmerman. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, 8(6):329 – 348, 1972.

[55] H. Müller, P. Clough, T. Deselaers, and B. Caputo, editors. *ImageCLEF – Experimental Evaluation in Visual Information Retrieval*, volume 32 of *The Information Retrieval Series*. Springer Berlin Heidelberg, 2010.

[56] R. Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, Feb. 2009.

[57] R. Navigli and P. Velardi. An analysis of ontology-based query expansion strategies. In *Proceedings of the 14th European Conference on Machine Learning, Workshop on Adaptive Text Extraction and Mining*, pages 42–49, 2003.

[58] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[59] Y. Qiu and H.-P. Frei. Concept-based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 160–169, New York, NY, USA, 1993. ACM.

[60] S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, ACL-07, pages 464–471. Association for Computational Linguistics, 2007.

[61] C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[62] S. Robertson. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 2004.

[63] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, Apr. 2009.

[64] S. E. Robertson. On term selection for query expansion. *J. Doc.*, 46(4):359–364, Dec. 1990.

[65] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[66] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System*, pages 313–323, Englewood Cliffs, NJ, 1971. Prentice-Hall.

[67] R. Rosenfeld. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, Aug 2000.

[68] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(02):95–145, 2003.

[69] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *J. Amer. Soc. Info. Science*, 41(4):288–297, 1990.

[70] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.

[71] S. Schulz, H. Stenzhorn, M. Boeker, and B. Smith. Strengths and limitations of formal ontologies in the biomedical domain. *Revista electronica de comunicacao, informacao & inovacao em saude: RECIIS*, 3(1):31–45, Mar. 2009.

[72] H. Schütze and J. O. Pedersen. A co-occurrence-based thesaurus and two applications to information retrieval. *Inf. Process. Manage.*, 33(3):307–318, May 1997.

[73] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[74] M. Song, I.-Y. Song, X. Hu, and R. B. Allen. Integration of association rules and ontologies for semantic query expansion. *Data Knowl. Eng.*, 63(1):63–75, Oct. 2007.

[75] J. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on*, 37(3):332–341, Mar 1992.

[76] J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–491, Dec. 1998.

[77] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[78] R. Sun, C.-H. Ong, and T.-S. Chua. Mining dependency relations for query expansion in passage retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 382–389, New York, NY, USA, 2006. ACM.

[79] D. Tudhope, C. Binding, D. Blocks, and D. Cunliffe. Query expansion via conceptual distance in thesaurus indexed collections. *Journal of Documentation*, 62(4):509–533, 2006.

[80] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[81] E. M. Voorhees and D. Harman. Overview of the sixth Text REtrieval Conference (TREC-6). *Inf. Process. Manage.*, 36(1):3–35, Jan. 2000.

[82] W. S. Wong, R. W. P. Luk, H. V. Leong, K. S. Ho, and D. L. Lee. Re-examining the effects of adding relevance information in a relevance feedback environment. *Inf. Process. Manage.*, 44(3):1086–1116, May 2008.

[83] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):13:1–13:37, June 2008.

[84] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, Jan. 2000.

[85] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, CIKM '01, pages 403–410, New York, NY, USA, 2001. ACM.

[86] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.