Exploiting multi-word similarity for retrieval in medical document collections

Citation for published version (APA):
Drymonas, E., Zervanou, K., & Petrakis, E. G. M. (2010). Exploiting multi-word similarity for retrieval in medical document collections: The TSRM approach. *Journal of Digital Information Management*, *8*(5), 315-321.

Document status and date:
Published: 01/10/2010

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Exploiting Multi-Word Similarity for Retrieval in Medical Document Collections: the TSRM Approach

Euthymios Drymonas, Kalliopi Zervanou, Euripides G.M Petrakis
Department of Electronic and Computer Engineering
Technical University of Crete (TUC)
Chania, Greece
edrimon@gmail.com, {kelly, euripides}@intelligence.tuc.gr

Journal of Digital Information Management

ABSTRACT: *In this paper, we investigate on potential improvements to Information Retrieval (IR) models related to document representation and conceptual, topic retrieval, in medical document collections. We propose the TSRM*[1] *(Term Similarity and Retrieval Model) approach, where document representations are based on multi-word domain terms, rather than mere single key-words, typically applied in traditional IR. The proposed representation is semantically compact and more efficient, being reduced to a limited number of meaningful multi-word terms (phrases), rather than large vectors of single-words, part of which may be void of distinctive content semantics. In computing document similarity, contrary to other state-of-the-art methods examined in this work, TSRM adopts a knowledge poor solution, namely an approach which does not require any existing knowledge resources, such as ontologies, or thesauri. The evaluation of TSRM is based on OHSUMED, a standard TREC collection of medical documents and illustrated the efficiency of TSRM over other well established general purpose IR models.*

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Retrieval models; **I.2.7 [Natural Language Processing]:** Language models;

General Terms: Information Retrieval, Term Similarity, Medical information retrieval

Keywords: Term extraction, Term similarity, Information retrieval, Document representation, Medical information retrieval

Received: 8 March 2010, Revised 29 June 2010, Accepted 8 July 2010

1. Introduction

The ever increasing amount of textual information in document repositories, such as digital libraries and the Web, brings about new challenges to information management. The extraction of domain terms plays an important role towards better understanding of the contents of document collections. In particular, the extraction of multi-word domain terms can be used to improve the accuracy of processes, such as document indexing and retrieval, by means of improved document representations. This type of term extraction, when applied for document representation, it may reduce representations into a limited number of meaningful phrases, rather than large vectors of words, part of which may be void of distinctive content semantics, thus allowing for more efficient and accurate content representation of the document collection.

In classical IR [1], document representations are typically based on large vectors of single-word indexing terms. A multi-word term, such as "*carotid artery disease*" would be included in the vector as three different indexing terms, which in turn represent the document content independently. Multi-word or compound terms are not only vested with more compact and distinctive semantics (e.g., the term "*carotid artery disease*" distinguishes the document from any other document referring to other carotid artery information or diseases), but they also present the advantage of lexically revealing their semantic content classificatory information, by means of modifiers [2]. For example, the compound term *"carotid artery disease"* denotes a type of "*artery disease*", which in turn is a type of *"disease"*. In this example, the single-word term *"disease"* has no apparent indication of its respective category, whereas for the multi-word terms, their modifiers, *"carotid artery"* and *"artery"*, provide an indication of their respective reference to a specific disease type. Therefore, an information retrieval model relying on multi-word terms for document representation, would be faster and more efficient, because this kind of representation would reduce the document vector size, while retaining more detailed and meaningful document semantics.

An almost orthogonal issue in IR is term similarity. Plain lexicographic analysis and matching is not always sufficient to determine whether two terms are similar, and consequently, whether two documents are similar. For example, synonymous terms, such as *"heart attack"* and *"myocardial infarction"*, share the same meaning, but completely differ lexically. Furthermore, similar documents may contain conceptually similar terms, but not necessarily the same. For example, two terms referring to arterial diseases, such as *"carotid artery disease"* and *"coronary artery disease"*, are conceptually and lexically similar, since they both refer to arterial diseases, but they are not the same, since they refer to two different subtypes of the disease. Therefore, neither the lack of lexically similar, nor the existence of lexically the same document terms to a query guarantee the relevance of the retrieval results. A solution adopted by many recent contributions in IR suggests discovering semantically similar (single-word) terms in documents using general, or application specific term taxonomies (e.g., WordNet[2] or MeSH[3]) and by associating such terms using semantic similarity methods [3,4,5].

We propose *TSRM* (Term Similarity and Retrieval Model), a novel information retrieval model, which estimates relevance based on documents containing conceptually similar but not necessarily the same terms. *TSRM* relies on improved query and document representations by implementing a linguistic term extraction method for the identification of multi-word, rather than single-word domain

[1] A preliminary version of this work appeared in the 13th Intern. Conference on Applications of Natural Language to Information Systems (NLDB 2008), June 24-27, 2008, London, UK.

[2] http://wordnet.princeton.edu

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

terms. For the computation of similarity among multi-word terms, *TSRM* does not require any pre-existing domain knowledge resources (as it is typical in recent IR work [3,4,5]); it exploits current natural language processing research in establishing lexical and contextual term similarity criteria [6], while taking into consideration term variation phenomena [7].

We illustrate the effectiveness of *TSRM* in retrieval of medical documents on OHSUMED, a standard TREC collection of Medline[4] abstracts. The experimental results demonstrate very promising performance improvements over both, classic information retrieval methods and, state-of-the art semantic retrieval methods utilizing ontologies in most cases.

In the following, we present the *TSRM* model, our experiments in establishing term and document similarity measures, and the results evaluation in document retrieval. We start in Sec. 2 with a presentation of related research in term extraction and term similarity. Subsequently, in Sec. 3, the existing approaches exploited by *TSRM* for term extraction, variant recognition and term similarity criteria are presented in more detail and, finally, in Sec. 4 and Sec. 5 our *TSRM* model and our experiments are discussed. We conclude with a discussion on our results and future work in Sec. 6.

2. Related Work

Domain term extraction aims at the identification of linguistic expressions denoting specialized concepts, namely domain or scientific terms. The automatic identification of domain terms is of particular importance in the context of information management applications, because these linguistic expressions are bound to convey the informational content of a document. In early approaches, terms have been sought for indexing purposes, using mostly $tf \cdot idf$ counts [1]. Term extraction approaches largely rely on the identification of term formation patterns [2,8]. Statistical techniques may also be applied to measure the degree of unithood or termhood of the candidate multi-word terms [9]. Later and current approaches tend to follow a hybrid approach combining both statistical and linguistic techniques [10].

A state-of-the art method for extracting multi-word terms is KEA [11]. KEA automatically extracts key-phrases from the full text of documents. The set of all candidate phrases in a document are identified using rudimentary lexical processing, features are computed for each candidate, and machine learning is used to generate a classifier that determines which candidates should be assigned as key-phrases. C/NC-value [10] is a domain-independent method for the automatic extraction of multi-word terms, combining linguistic and statistical information. It enhances the common statistical measure of frequency of occurrence so as to incorporate information on nested terms and term context words for the recognition of domain multi-word terms. Comparative experiments of $tf \cdot idf$, KEA and the C/NC-value term extraction methods by Zhang et al. [12] show that C/NC-value significantly outperforms both $tf \cdot idf$ and KEA, in a narrative text classification task, using the extracted terms.

Since automatic term extraction is primarily based on term form patterns, it inherently suffers from two problems: ambiguity and variation. Ambiguity relates to the semantic interpretation of a given term form and it arises when this form can be interpreted in more than one way. Variation is generally defined as the alteration of the surface term form of a terminological concept. According to Jacquemin [7], variation is more specifically defined as a transformation of a controlled multi-word term and can be of three types: morphological, syntactic or semantic. Many approaches (e.g., [7,8]) attempt to resolve the problems of ambiguity and variation in terminological concepts by combining simple text normalization techniques, statistics, or more elaborate rule-based, linguistic techniques with existing thesaurus and lexicon information.

2.1 Term Similarity

In *TSRM* we attempt to match conceptually similar documents to a query by establishing term similarity to relate similar documents and associate relevant documents to a query. Many approaches attempt to cluster similar terms based on supervised or unsupervised methods. Lexical similarity methods have used multi-word term constituents and syntactic term variants to cluster similar terms [6]. Contextual similarity has been researched in statistical methods taking into consideration the term co-occurring context. Term co-occurrence methods [13,14,15] are based on the assumption that semantically similar terms co-occur in close proximity. Verb selectional restriction methods (i.e., subject-verb and verb-object) are based on the assumption that verbs in specialized document contexts tend to restrict the term type appearing as their subject, or object [14]. For example, in the phrases *“suffered a knee fracture”* and *“suffered a cardiac arrest”* the verb *“suffer”* co-occurs with similar terms (i.e. terms denoting diseases). Other studies consider the context of additional grammatical categories [15], or attempt to establish similarity based on specific syntactic structures, such as conjunctions (e.g., *“peripheral and carotid artery disease”* where *“and”* connects similar terms), enumerations (e.g., *“peripheral, carotid and coronary artery diseases”*), or other patterns [6].

2.2 Term Mismatch in Information Retrieval

The establishment of term similarity in *TSRM* aims not only at the identification of similar concepts expressed by different terms across documents, but also it aims at the resolution of the so called term mismatch problem in IR, by allowing for query expansion using similar terms.

Query expansion with potentially related (e.g., similar) terms has long been considered a means for dealing with the term mismatch problem in IR. Term expansion attempts to automate the manual or semi-automatic query re-formulation process based on feedback information from the user. There are also approaches which attempt to improve the query with terms obtained from a similarity thesaurus. This thesaurus is usually computed by automatic or semi-automatic corpus analysis (global analysis) and may not only introduce new terms, but also reveal term relationships and estimate the degree of relevance between terms. Possas et.al. [13] exploit the intuition that co-occurrent terms occur close to each other and propose a method for extracting patterns of co-occurrent terms and their weights by data mining. The work referred to above is complementary to methods which expand the query with co-occurrent terms (e.g., “*heart*”, “*attack*”) in retrieved documents [16] (local analysis). Dimensionality reduction (e.g., by Latent Semantic Indexing [17]) has been also proposed for dealing with the term mismatch problem.

In Semantic Similarity Retrieval Model (SSRM), Hliaoutakis et.al. [3] show how to handle more relationship types (hyponyms and hypernyms in an ontology or term taxonomy, such as WordNet[5] or MeSH[6] and how to compute good relevance weights given the $tf \cdot idf$ weights of the initial query terms. They focus on semantic relationships only and demonstrate that it

[4] http://www.nlm.nih.gov/databases/databases_medline.html

[5] http://wordnet.princeton.edu

[6] http://www.nlm.nih.gov/mesh

is possible to enhance the performance of retrievals using this information alone. Voorhees [5] proposed expanding query terms with synonyms, hyponyms and hypernyms in WordNet but did not propose an analytic method for setting the weights of these terms. Voorhees reported some improvement for short queries, but little or no improvement for long queries. Along the same lines, Mihalcea [4] proposed associating only the most semantically similar terms in two documents. Query terms are not expanded, nor re-weighted.

3. The *TSRM* method resources

In the following, we present the *TSRM* resources in more detail namely, a natural language processing approach for multi-word term extraction, the FASTR tool for term variants detection [7] and, finally, the term similarity measures which are inspired by the lexical and contextual term similarity criteria defined by the work of Nenadic et al. [6].

3.1 Term Extraction

We apply C/NC-value [10], a domain-independent natural language processing method for the extraction of multi-word and nested terms. The text is first tokenized and tagged by a part-of-speech tagger (i.e., a grammatical category, such as noun, verb, adjective, etc. is assigned to each word). This is implemented using tools from the OpenNLP suite[7]. As an enhancement to the method, we applied the morphological processor from Java WordNet Library (JWNL)[8]. JWNL is an API for accessing WordNet-style relational dictionaries. It also provides relationship discovery and morphological processing. The morphological processor attempts to match the form of a word or phrase to its respective lemma (i.e., its base form in WordNet). We decided to incorporate this enhancement in our approach, because it allows forhandling of morphological variants of terms, such as *"blood cell"* — *"blood cell"*, or *"blood clotting"* — *"blood clot"*. Subsequently, a set of linguistic filters is used to identify in text candidate term phrases, such as the following:

- Noun+ Noun
- (Adj | Noun)+ Noun
- ((Adj | Noun)+ | ((Adj | Noun)* (NounPrep)?)(Adj | Noun)*) Noun

In our current implementation the selection of all three filters is available. However, the last filter has been applied, because it is an open filter which reveals more terms.

The subsequent statistical component defines the candidate noun phrase termhood by two measures: C-value and NC-value. The first measure, the C-value, is based on the hypothesis that multi-word terms tend to consist of other terms (nested in the compound term). For example, the terms "*coronary artery*" and "*artery disease*" are nested within the term "*coronary artery disease*". Thus, C-value is defined as the relation of the cumulative frequency of occurrence of a word sequence in the text, with the frequency of occurrence of this sequence as part of larger proposed terms in the same text. The second measure, the NC-value, is based on the hypothesis that terms tend to appear in specific context and often co-occur with other terms. Thus, NC-value refines C-value by assigning additional weights to candidate terms which tend to co-occur with specific context words.

3.2 The FASTR Method for Term Variants Extraction

The FASt Term Recognizer (FASTR) [7] has been designed for the recognition and expansion of indexing terms. It is based on a combination of linguistic filtering (for term extraction) and morpho-syntactic rules (for term variants detection). The term extraction phase in FASTR identifies noun phrases which are subsequently associated with their respective variants in the term variant recognition phase.

The principal types of term variation phenomena FASTR deals with are three: morphological, semantic and syntactic. Term morphological variation is due to grammatical, or derivational affixation (e.g., *"artery wall"*, *"arterial wall"* and *"artery walls"*), whereas term semantic variation is due to the use of synonyms (e.g., *"heart attack"* and *"myocardial infarction"*). For the detection of these morphological and semantic variants FASTR requires external knowledge resources, such as the CELEX dictionary[9]. Syntactic term variants are due to modifications of the syntactic structure of the term, due to designator insertions, expansions and permutations (e.g., ``*human clones*'', ``*human DNA clones*'' and ``*clones of human DNA*''). For the detection of these syntactic variants FASTR applies morpho-syntactic rules.

3.3 The *CLS* Term Similarity method

In a method proposed by Nenadic et al. [6] term similarity, *CLS*, is defined as a linear combination measure of three similarity criteria, Lexical Similarity (LS), Contextual Similarity (CS) and Syntactic Similarity (SS). We have implemented in our *TSRM* model the Lexical Similarity (LS) and the Contextual Similarity (CS) measures in our process for term similarity computation. The syntactic similarity criterion (i.e., the third criterion which we did not use) is principally based on detecting enumeration and coordination structures (e.g., "*such as . . .*", "*. . . and . . .* '', "*either. . . or. . .* '' etc.). This criterion has been considered constraint to rare patterns and is expected to have very low recall in many corpora. It is corpus-dependent: As mentioned in [6], the size of the corpus and the frequency with which the concurrent lexico-syntactic patterns are realized in it, affect the syntactical similarity.

"Lexical Similarity" between terms is based on identifying their respective common subsequences. By comparing all their non-empty subsequences, it is possible to give more credit to pairs of terms sharing longer nested constituents. An additional credit is given to terms having common heads. The definition of LS attempts to combine previous research on term clustering and term similarity based on term heads, modifiers and prepositional phrase modifiers in a single measure, computed according to a Dice-like coefficient formula, as shown in Eq. 1:

$$LS(t_1,t_2)=\frac{|P(t_1)\cap P(t_2)|}{|P(t_1)+P(t_2)|}+\frac{|P(h_1)\cap P(h_2)|}{|P(h_1)+P(h_2)|} \quad (1)$$

where the LS between terms t_1 and t_2 (whose heads are denoted by h_1 and h_2 respectively) is computed as the set of their shared constituents divided by the set of their total constituents and where potentially common head constituents increase term similarity. In this formula, $P(t_1)$, $P(t_2)$ and $P(h_1)$, $P(h_2)$ denote the sets of terms and heads of terms *t*1 and *t*2 respectively.

The rationale behind lexical similarity involves the following hypotheses: (a) terms sharing a head are assumed to be (in)direct hyponyms of the same term (e.g., "progesterone receptor" and

[7] http://opennlp.sourceforge.net
[8] http://sourceforge.net/projects/jwordnet
[9] http://www.ru.nl/celex

"oestrogen receptor" are both receptors); (b) when a term is nested inside another term, we assume that the terms are related (e.g., "retinoic acid receptor" and "retinoic acid" should be associated).

Contextual Similarity (CS) is based on the assumption that similar terms appear in similar contexts. If two terms can substitute each other in similar contexts, then they can be deemed similar. In [6] there are two main categories of generalized context patterns: morpho-syntactic (e.g., noun phrases, verb phrases, prepositional phrases) and terminological (i.e., term occurrences). Left and right term context are treated as separate term features and the CS is computed as a Dice-like coefficient formula:

$$CS(t_1,t_2) = \frac{|P(CL_1) \cap P(CL_2)|}{|P(CL_1) + P(CL_2)|} + \frac{|P(CR_1) \cap P(CR_2)|}{|P(CR_1) + P(CR_2)|} \quad (2)$$

where $P(CL_1)$, $P(CR_1)$, $P(CL_2)$, $P(CL_2)$, $P(CR_2)$ are sets of left and right context patterns associated with terms t_1 and t_2 respectively.

We generated all possible "linearly nested" patterns for each given context. In particular, when considering left contexts, contexts of the maximal length (without crossing the sentence boundary) are initially selected and subsequently they are iteratively trimmed on the left side until the minimal length is reached. Right contexts are treated analogously. The following example illustrates the left linear pattern generation process:

V	PREP	TERM	NP	PREP	(the maximal pattern)
	PREP	TERM	NP	PREP	
		TERM	NP	PREP	
			NP	PREP	(the minimal pattern)

"NP" stands for "Noun Phrase", a basic syntactic structure. During our implementation we experimented with various maximal pattern lengths. Based on our results, we decided to set the minimum pattern length to 2 and the maximum to 8.

Certain grammatical categories were removed from context patterns, since not all of them are equally significant in providing useful contextual information. Adjectives (that are not part of terms), adverbs and determiners can be filtered from context patterns, for they rarely bare information. In addition, the so-called "linking words" (e.g., "*however", "moreover",* etc.), or more generally, "linking constructs" (e.g., verb phrases, such as "*result in", "lead to", "entail"*, etc.) were also considered non-informative and were filtered.

3.4 Term Similarity Estimation

For the combination of lexical and contextual similarities into a single measure and in order to establish the relative importance of each in determining term similarity, we conducted a machine learning experiment: We randomly selected 200 term pairs and we asked domain experts to provide a similarity estimate for each pair, in a range between 0 (not similar) and 3 (perfect similarity). To estimate the relative importance of the two similarity criteria (lexical and contextual), we used the above training set as input to a Decision Tree [19]. The evaluation method was stratified cross validation.

We have thus obtained the following Term Similarity (TS) measure:

$$TS = 0.8 \cdot LS + 0.2 \cdot CS \quad (3)$$

In order to evaluate the performance of the TS similarity formula above, we compared the similarity scores computed by this formula (for the same terms) with the human relevance results as in the experiment conducted by Miller and Charles [20]: The similarity values obtained by Eq. 3 are correlated with the average scores obtained by humans. The higher the correlation of a method, the better the method is (i.e., the more it approaches the results of human judgments). The experimental results indicate that *TS* approximates algorithmically the human notion of similarity, reaching correlation (with human judgment of similarity) up to 72%.

Table 1 illustrates examples of terms pairs along with their computed term similarity. Notice that, for term pairs sharing no common lexical constituents, the method is still able to compute a valid similarity value based on context.

Term Pair	LS	CS	TS
mast cell - tumor cell	0.67	0.39	0.61
surgical technique - operative technique	0.67	0.23	0.58
blood flow - coronary artery	0	0.45	0.09
male patient - female patient	0.67	0.65	0.66
pericardial effusion - stress test	0	0.38	0.07
endoscopic examination - melanoma cell	0	0.41	0.08
blood pressure - systolic blood pressure	0.83	0.65	0.79
phosphatidic acid - medical practice	0	0.32	0.06
arterial pressure - systolic blood pressure	0.61	0.59	0.60

Table 1. Example of term pairs and of their computed Lexical Similarity (LS).

4. Term-based Document Similarity: the *TSRM* Approach

Queries and documents are first syntactically analyzed and reduced into multi-word term vectors. Each term in this vector is represented by its *tf · idf* weight. In traditional IR approaches, very infrequent or very frequent terms are eliminated. However, with *TSRM* there is no need to omit terms in our vectors, given that document vectors are usually very small (consisting of less than 20-30 terms).

The approach adopted by *TSRM* for the document retrieval application can be viewed as a three phase process: the *"Corpus Processing phase"*, the *"Query Processing phase"* and, finally, the *"Document Retrieval phase"* which are described below.

4.1 Corpus Processing

Pre-processing and Term Extraction: During this phase, the term extraction method of Sec. 3.1 is applied. The processing and storage of term vectors and their respective context patterns were implemented using BerkeleyDB[10]. To facilitate searching, inverted files were created containing multi-word terms.

Term Variants Detection: In this stage we use the FASTR tool[11] modified to work in pipeline with our linguistic pre-processing tools. The role of FASTR in our approach primarily consists in the identification of morpho-syntactic variants for terms discovered by the previous, term extraction process. In *TSRM* we

[10] http://www.oracle.com/database/berkeley-db/je/index.html

[11] http://www.limsi.fr/Individu/jacquemi/FASTR

opted for a knowledge-poor approach, therefore domain specific external resources, such as domain lexica and thesauri, were not applied. For this reason, we did not apply semantic variant detection using FASTR. Simple text normalization techniques, such as stemming, could be applied for the resolution of morphological term variation phenomena. Given that stemming may lead to erroneous suffix removal (overstemming and/or understemming problems) and given that currently we can easily access existing general language dictionaries, such as the WordNet, we have opted to deal with morphological variation by incorporating the morphological processor from Java WordNet Library[12]. For the purposes of syntactic term variation resolution, we have decided to exploit existing research in automatic term extraction by applying a rule-based, linguistic method for term variants detection of FASTR. Our approach to syntactic and morphological variation is expected to have a positive impact on our results because it is founded on existing research in automatic domain term extraction and on linguistic principals, rather than heuristics.

Term Similarity Estimation: We use the C4.5 Decision Trees classifier and Eq. 3 to estimate initial similarities among terms in our document collection. The results of this process consist of term pairs with respective similarity estimations and are stored in a database.

4.2 Query Processing

Query Pre-processing and Term Extraction: In a similar manner to the respective stages for the Corpus processing phase, the queries and respective query descriptions are linguistically pre-processed and their respective terms are extracted.

Query Expansion: The query terms identified by the previous, Query Term Extraction process is expanded to include identified term variants found in the documents during the Term Variants Detection stage. Thus, subsequent document similarity computation is to be based on both query term types (actual and variants).

4.3 Document Retrieval

Document Similarity Computation and Ranking: The similarity between a query *q* and a document *d* can be computed invarious ways. Typically, document similarity can be computed as in the Vector Space Model (VSM) [1]:

$$Document - Similarity(q,d) = \frac{\sum q_i d_i}{\sqrt{\sum q_i^2}\sqrt{\sum d_i^2}} \qquad (4)$$

where *i* and *j* denote terms in the query and the document and *qi* and *di* are their weights in their respective vector representations. Typically, the weight *di* of a term *i* in a document is computed as $d_i = tf_i \cdot idf_i$, where tf_i is the frequency of term *i* in the document, and *idfi* is the inverse document frequency of *i* in the whole document collection. All weights are normalized by document length. According to VSM, two documents are similar, only if they have (at least some) common terms. Consequently, VSM will fail to retrieve documents which contain conceptually similar, but not lexically similar terms. To deal with this problem, we have modified the formula in [4] (which works only for single-word terms) to take into consideration lexical and contextual term similarity and we compute document similarity as follows:

$$Document - Similarity = \frac{1}{2}\{\frac{\sum_{i \in q} idf_i \max_j TS(i,j)}{\sum_{i \in q} idf_i} + \frac{\sum_{j \in d} idf_j \max_i TS(j,i)}{\sum_{j \in d} idf_i}\} (5)$$

The weight of a term is computed as its inverse document similarity *idf* as in VSM. Eq. 5 takes into account dependencies between non-identical terms. Their dependence is expressed quantitatively by virtue of their term similarity *TS* and this information is taken explicitly into account in the computation of document similarity. Eq. 5 proposes the association of only the most similar terms in two documents by summing up their similarities (weighted by the inverse document frequency *idf*). Notice however the quadratic time complexity of Eq. 5 as opposed to the linear time complexity of Eq. 4 of VSM. To speed up similarity computations, the semantic similarities between term pairs are stored in a hash table. A computation complexity analysis of the methods used in this work can be found in [3]. All document similarity measures above (VSM, TSRM) are normalized in the range [0,1].

5. Evaluation

We conducted a series of comparative experiments, so as to investigate the potential efficiency of *TSRM* over classic IR Models (such as VSM) and, most importantly, the relative performance of *TSRM* compared to state-of-the-art IR methods using external knowledge resources (such as ontologies, or term taxonomies) for discovering term associations (e.g., [5,3]): Query expansion by semantically similar terms is applied as a means for capturing similarities between terms of different degrees of generality in documents and queries (e.g., ``*human*”, ``*man*”). Queries are augmented with conceptually similar terms (i.e., hyponyms and hypernyms) which are retrieved from a taxonomy, or ontology.

The following methods are implemented and evaluated:

TSRM: the proposed method with document similarity computed by Eq. 5. This method works by associating only the most semantically similar terms in two documents and by summing up their similarities (weighted by the inverse document frequency, *idf*).

VSM Single Word Vectors [1]: the classic Vector Space Model (VSM) approach, using vectors of single-word terms as it is typical in IR applications.

VSM Multi Word Vectors: the same method as above, using vectors of multi-word terms.

SSRM [3]: A knowledge-based IR method. Queries are expanded with semantically similar terms from the MeSH[13] medical terms thesaurus.

Voorhees [5]: A knowledge-based IR method, using MeSH for discovering term similarities as above. The query terms are always expanded with hyponyms one level higher or lower in the taxonomy, and synonyms.

TSRM and its competitors have been tested on OHSUMED[14], a standard TREC collection of 348,566 medical document abstracts from Medline, published between 1988-1991. OHSUMED is commonly used in benchmark evaluations of IR applications. OHSUMED provides queries and the relevant answer set (documents) for each query. These correct answers were compiled by the editors of OHSUMED and are also available from TREC. For the evaluations, we applied all 64 queries available.

[12] http://sourceforge.net/projects/jwordnet

[13] http://www.nlm.nih.gov/mesh

[14] http://trec.nist.gov/data/t9_filtering.html

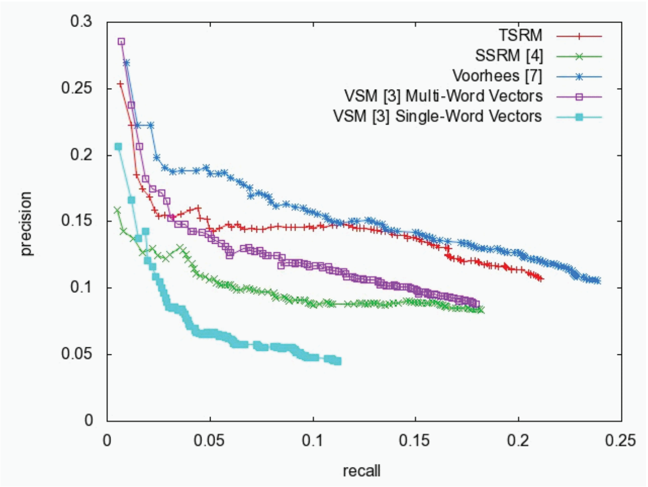

Figure 1. Precision-recall diagram for retrieval on OHSUMED

In the experiments illustrated in Fig. 1 each method is represented by a precision/recall curve. For each query, the best 50 answers were retrieved (the precision/recall plot of each method contains exactly 50 points). The top-left point of the precision/recall curve corresponds to the precision/recall values for the best answer, or best match (which has rank 1), while the bottom right point corresponds to the precision/recall values for the entire answer set. Notice that it is possible for two precision-recall curves to cross-over. This means that one of the two methods performs better for small answer sets (containing less answers than the number of points up to the cross-section), while the other performs better for larger answer sets.

The results in Fig. 1 demonstrate that *TSRM* and the knowledge-based IR method by Voorhees [5] clearly outperform all other methods. This method performs clearly better for small answer sets with up to 25 answers. This is an important result on itself, showing that it is possible for *TSRM* to approximate the performance of methods making use of external knowledge resources (i.e., the MeSH taxonomy of medical terms in our case) for enhancing the performance of retrieval. However, knowledge-based methods, such as [5,3] are only applicable to corpora for which such external resources (ontology, thesauri, or term hierarchy) are available. Contrary to knowledge-based methods, *TSRM* does not rely on external resources and can be applied on any corpus at the cost of an initial exhaustive corpus pre-processing step for discovering term similarities.

Some problems related to the processing of OHSUMED were first identified in our pilot experiments, reported in [21]. These relate to small document size (i.e., OHSUMED is a collection of MEDLINE document abstracts), which poses constraints to the application of statistical methods, such as *TSRM*. The performance of *TSRM* may improve for full documents and larger data sets. Such corpora were not available to us.

We observe that VSM achieves a good performance with multi-word terms. This method performed equally well with *TSRM* and Voorhees [5] for small answer sets. This result confirms that domain term extraction can be used to improve the accuracy of IR processes by means of improved document representations. Using multi-word, domain term extraction, document representations may be reduced to a limited number of meaningful phrases, rather than large vectors of words, part of which may be void of distinctive content semantics.

A closer look into the results reveals that the efficiency of *TSRM* is mostly due to the contribution of non-identical but conceptually similar terms. VSM (similar to most classical retrieval models relying on lexical term matching) ignores this information. Most queries have relatively few relevant answers, most of them containing exactly the query words. These are easily recognized by plain lexical matching and are retrieved by VSM.

The addition of semantic relations in SSRM [3] didn't actually improve the performance of retrievals. In SSRM, query terms are expanded by more than one level up in the MeSH taxonomy. The reason for performance deterioration in SSRM lies in query expansion which resulted in topic drift. Nevertheless, this should not be regarded as a failure of these methods but rather as a failure of MeSH to provide terms conceptually similar to the topic. MeSH is a medical term thesaurus (rather than an ontology of medical terms), where not all linked terms are in fact similar, so as to be appropriate for query expansion purposes. For this reason, the application of a rigorously built domain term taxonomy, or ontology is expected to improve results. Such a knowledge resource was not available to us for these experiments. In *TSRM*, term similarity is aligned and therefore optimized for the corpus and query context.

6. Conclusions

The focus of this work is not on term extraction but on showing that extraction of multi-word domain terms may improve the accuracy of document representations, while reducing their size, and, consequently, improve the quality and efficiency of retrievals in text collections. Our experiment with TSRM confirms this assumption. *TSRM* relies on C/NC-value, a well established, domain independent method for the extraction multi-word domain terms. In our attempt to improve on topic and conceptual retrieval, we researched on approaches, which aim at establishing conceptual similarity among terms in documents, using internal (lexical) and external (contextual) criteria, while taking into consideration term variation (morphological and syntactic). In our approach, we opted for a knowledge-poor solution, namely an approach which does not require any existing knowledge resources, such as ontologies, or thesauri.

TSRM has been designed to rely on domain term recognition typically expressed in multi-word phrases. Consequently, the user must be familiar with documents content and application terminology and the method assumes documents originating from a single application domain and of sufficient size to allow for the statistic part of the term extraction method to extract statistically significant terms. This is the main limitation of *TSRM*. Because OHSUMED is the only corpus available to us satisfying these restrictions (all other available test corpus are either too small or consist of documents from multiple domains) we have decided to limit the scope of *TSRM* to medical documents. Future developments on *TSRM* method include: experimentation with more data sets in different application domains; investigation of extensive syntactic similarity patterns based on text mining and information extraction techniques; and extending the term similarity methods for handling semantic relationships between common words (or phrases), using ontologies.

References

[1] Baeza-Yates, R., Ribeiro-Neto, B (1999). Modern Information Retrieval. Addison Wesley Longman.

[2] Bourigault, D., Gonzalez-Mullier, I., Gros, C (1996). LEXTER, a Natural Language Tool for Terminology Extraction. *In:* 7th EURALEX Intl. Congress on Lexicography, Part II, G¨oteborg University, Goteborg, Sweden 771–779.

[3] Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E., Milios, E (2006). Information Retrieval by Semantic Similarity. *Intl.*

Journal on Semantic Web and Information Systems (IJSWIS) 3 (3) (July/September) 55–73.

[4] Mihalcea, R., Corley, C., Strapparava, C (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *In:* AAAI 2006, Boston (July) 530–536.

[5] Voorhees, E (1994). Query Expansion Using Lexical-Semantic Relations. *In*: 17th ACM SIGIR Conf., Dublin, Ireland 61–69.

[6] Nenadic, G., Spasic, I., Ananiadou, S (2004). Automatic Discovery of Term Similarities Using Pattern Mining. *Intl. Journal of Terminology* 10 (1) 55–80.

[7] Jacquemin, C (2001). Spotting and Discovering Terms through Natural Language Processing. MIT Press, Cambridge, MA, USA.

[8] Gaizauskas, R., Demetriou, G., Humphreys, K (2000). Term Recognition in Biological Science Journal Articles. *In*: Workshop on Computational Terminology for Medical and Biological Applications, (NLP 2000), Patras, Greece 37–44.

[9] Daille, B., Gaussier, E., Lange, J (1994). Towards Automatic Extraction of Monolingual and Bilingual Terminology. *In:* COLING-94, Kyoto, Japan 515–521.

[10] Frantzi, K., Ananiadou, S., Mima, H (2000). Automatic recognition of multi-word terms: The C-Value/NC-value Method. *Intl. Journal of Digital Libraries* 3 (2) 117–132.

[11] Witten, I., Paynter, G., Frank, E., Gutwin, C., Nevill-Manning, C (1999). KEA: Practical Automatic Keyphrase Extraction. In: 4th ACM Conf. on Digital Libraries, Berkeley, CA, USA (Aug) 254–255

[12] Zhang, Y., Milios, E., Zincir-Heywood, N (2005). Narrative Text Classification and Automatic Key Phrase Extraction in Web Document Corpora. *In*: 7th ACM Intl. Workshop on Web Information and Data Management (WIDM 2005), Bremen, Germany (Nov. 5) 51–58.

[13] Possas, B., Ziviani, N., Meira, W., Ribeiro-Neto, B (2005). Set-Based Vector Model: An Efficient Approach for Correlation-Based Ranking. *ACM Trans. on Info. Systems* 23 (4) (Oct.) 397–429.

[14] Hindle, D (1990). Noun Classification from Predicate-Argument Structures. *In*: 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, PA, USA 268–275.

[15] Grefenstette, G (1994). Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers.

[16] Attar, R., Fraenkel, A. (1977). Local Feedback in Full Text Retrieval Systems. *Journal of the ACM* 24 (3) 397–417The TSRM Approach for Information Retrieval 21.

[17] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41 (6) 391–407.

[19] Witten, I., Frank, E (2000). Data Minning. Morgan Kaufmann.

[20] Miller, G., Charles, W (1991). Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes* 6 1–28.

[21] Hliaoutakis, A., Zervanou, K., Petrakis, E, (2007). Medical document indexing and retrieval: Amtex vs. NLM MMTx *In*: Intern. Symposium for Health Information Management Research, Sheffield, UK.