Crestani, F. (2000) *Exploiting the similarity of non-matching terms at retrieval time.* Information Retrieval, 2 (1). pp. 23-43. ISSN 1386-4564

Information Retrieval, ?, 1–14 (?)

Exploiting the Similarity of Non-matching Terms at Retrieval Time*

FABIO CRESTANI
Department of Computing Science
University of Glasgow
Scotland, UK

fabio@dcs.gla.ac.uk

Received August 14, 1998; Revised September 13, 1999

Abstract. In classic Information Retrieval systems a relevant document will not be retrieved in response to a query if the document and query representations do not share at least one term. This problem, known as "term mismatch", has been recognised for a long time by the Information Retrieval community and a number of possible solutions have been proposed. Here I present a preliminary investigation into a new class of retrieval models that attempt to solve the term mismatch problem by exploiting complete or partial knowledge of term similarity in the term space. The use of term similarity enables to enhance classic retrieval models by taking into account non-matching terms. The theoretical advantages and drawbacks of these models are presented and compared with other models tackling the same problem. A preliminary experimental investigation into the performance gain achieved by exploiting term similarity with the proposed models is presented and discussed.

Keywords: Information Retrieval, term mismatch problem, term similarity, retrieval model.

1. Introduction

Information Retrieval (IR) is concerned with finding from a collection of documents those that are relevant to a user information need. The user describes his information need using a query which consists of a set of terms. In Boolean IR systems, terms are chosen by the user and are connected using Boolean operators (e.g., "and", "or", "not") to construct the query. In this paper I am not concerned with Boolean systems, but with systems that extract terms (index terms) from the text of a natural language query to build a query representation consisting of a set of weighted terms. Document representations, constructed in a similar way, are then matched with the query representation. Documents are ranked according to how well their representation matches the query representation [24].

A fundamental problem for IR is *term mismatch*. A query is usually a short and incomplete description of the user information need, and users and authors of documents indexed by the IR system often use different terms to refer to the same concepts.

This paper addresses the term mismatch problem proposing a new class of retrieval models that exploit the knowledge of term similarity in the term space. The term similarity is used at retrieval time to estimate the relevance of a document in response to a query by looking not only at matching terms, but also at non-matching terms.

The paper is structured as follows. In section 2, I discuss the importance of the term mismatch problem in IR. In section 3, I present a number of solutions to the problem that have been proposed in the past. A common graphical interpretation of these solutions is

*This work has been supported by a "Marie Curie" Research Fellowship from the European Commission.

used to help understand their effects on the term space. In section 4, I address the significance of term similarity information on the term space and the cost of this knowledge. In section 5, I present a class of models that exploit term similarity knowledge to tackle the term mismatch problem. The results of a preliminary investigation into the retrieval performance of these models are then described. These results should be read in the context of the evaluation framework presented in section 6. The actual results of the evaluation are presented and analysed in section 7. The paper concludes with section 8 where a discussion on the limitations of the experimentation is reported and directions of future work are examined.

2. The Term Mismatch Problem

Representing the user information need and the document informative content is a very difficult task in IR. Attempts to using advanced Natural Language Processing techniques or complex logical models have failed to solve the problem and IR is still using the classic technique of the "bag of terms" [21]. Terms are automatically extracted from or manually assigned to documents and queries. This way of representing documents and queries is common to both the Vector Space model [19] and the Probabilistic model [24], the two most important models of IR. However, representing documents and queries using a set of terms has a very serious side effect: *the term mismatch problem*.

Users of IR systems often use different term to describe the concepts in their queries than the authors use to describe the same concepts in their documents. It has been observed that two people use the same term to describe the same concept in less than 20% of the cases [8]. It has also been observed that this problem is more severe for short casual queries than for long elaborate ones because, as queries get longer, there is a higher chance of some important term co-occurring in the query and the relevant documents [28]. The term mismatch problem does not have only the effect of hindering the retrieval of relevant documents, it has also the effect of producing bad rankings of retrieved documents, as the following example shows.

Let us assume, for example, that a user would like to find information about "wine of the Tuscany region of Italy". The user submits to the IR system the query:

$$q = (wine,\ Tuscany)$$

Let us consider the following three documents:

$$\begin{aligned} d_1 &= (wine,\ France) \\ d_2 &= (wine,\ Italy) \\ d_3 &= (Florence,\ vineyard) \end{aligned}$$

Leaving aside considerations related to the indexing weights assigned to the terms used to represent documents and query, let us consider the *Retrieval Status Value* (RSV) of these documents in response to the query q. The RSV is an estimate of the relevance of a document with respect to a query that is performed according to the model the IR system uses. The RSV is used by the IR system to rank documents and present them to the user. An IR system using a classic model would assign to documents d_1 and d_2 a very similar RSV (how similar depends on the indexing weights assigned to terms), since both these documents have one term in common with the query. These documents would be ranked higher that document d_3, which has no term in common with the query. However, we can clearly see that document d_1 is surely not relevant, since it deals with French wine. Moreover, if we compare documents d_2 and d_3, we can argue that d_3 is more relevant that d_2, since d_3 deals with wine from Florence, a particular area of Tuscany, while d_2 deals with wine from the whole of Italy. Document d_3 is fully relevant to the query, while document d_2 is only partially relevant. We are therefore inclined to assign a higher RSV to d_3, closely followed by d_2 and then d_1. Such assignment of RSV is almost the opposite of that given by the IR system.

The above example shows the effect of the term mismatch problem. The use of advanced indexing models only partially limits these effects. In the following section I will briefly describe a number of proposed solutions to the term mismatch problem.

3. Approaches to the Term Mismatch Problem

There are a number of approaches to solving the term mismatch problem. In the following of this section I will briefly review some of these approaches showing how they attempt to tackle the problem. In this analysis, instead of describing one or more technique in detail, I will try to generalise the effects on the term space of the different techniques. On the basis of this analysis, I will argue that none of these approaches can completely solve the problem and each approach has its drawbacks.

3.1. Dimensionality Reduction

The most commonly used approach to the term mismatch problem consists in reducing the chances that a query and a document refer to the same concept using different terms. This can be achieved by reducing the number of possible ways a concept can be expressed, or in other words, reducing the "vocabulary" used to represent concepts.

With reference to the example of section 2, dimensionality reduction could be employed to reduce the indexing vocabulary of our system, so that, for example, the terms "wine" and "vineyard" could be replaced by only one term that expresses the concept of wine in all its aspects. The same can be done for the terms "Tuscany", "Florence", and "Italy", that could be replaced by a term that expresses the general geographical concept of Italy.

A number of techniques have been proposed for the dimensionality reduction of the term space. The most important ones are:

- manual thesauri [1];
- stemming and conflation [10];
- clustering or automatic thesauri [17, 23];
- Latent Semantic Indexing [8].

These techniques propose different strategies of term replacement. The strategies can be based on semantical considerations (manual thesauri), morphological rules (stemming and conflation), or term co-occurrence (clustering or Latent Semantic Indexing).

The effectiveness of dimensionality reduction techniques has been debated for long time. Stemming and term clustering, for example, have not proved to be always effective [22, 10]. The effectiveness of these techniques depends very much on the application domain and on the characteristics of the collection. In fact, techniques like stemming, term clustering or Latent Semantic Indexing that have proved to be somewhat effective with collections like TREC, are not so effective if used by Web search engines. These techniques are recall-oriented and, given the very short queries submitted to Web search engines, they would cause the retrieval of a large number of documents to the expenses of precision. While in TREC evaluation (in particular in the "ad hoc" track) the first 1000 documents are used in the evaluation, very few Web search engines' users look beyond the first 20 retrieved documents. Nevertheless, the reduction in the indexing space (and therefore memory space) that they produce has brought such engines as Excite and Infoseek to adopt them.

Another drawback of dimensionality reduction is that it may cause an over-simplification of the term space that may limit the expressiveness of the indexing language and could result in incorrect classification of unrelated terms.

3.2. Query Expansion

Another popular approach to the term mismatch problem is query expansion. This approach considers the query as a tentative definition of the concept the user is interested to find documents about. A number of different techniques can then be used to expand the original query submitted by the user to include other terms related to that concept. Documents are then matched against the new expanded query.

Referring to the example of section 2, query expansion could be used to expand the original query by adding terms related to the concept of wine (adding for example the term "vineyard") or the concept of Italy (adding for example terms that represent geographical regions of Italy). The difficulty in the correct application of query expansion lies in finding the best terms to add and in weighting in a correct way their importance.

The two most important techniques for query expansion are:

- automatic, semi-automatic, or interactive query expansion [9];
- relevance feedback [12].

Query expansion consists in automatically or semi-automatically adding terms to the query by selecting from the term space those that are most similar to the ones used originally by the user. In interactive query expansion some control is left to the user on the choice of terms to be added to the query.

Relevance feedback enables the selection of terms to be added to the original query terms by automatically extracting them from documents marked as relevant by the user.

It is commonly known that query expansion works; in fact it is used quite extensively in top performing TREC participating systems [13]. Nevertheless, theoretically speaking, this approach too has a few drawbacks. The most important one is related to the difficult choice of terms to be added to the original query

terms. Automatic query expansion techniques relies on accurate ways of finding term relations. Most of the approaches currently in use are very much domain and application dependent and require a long tuning process before being applied to a new domain and application. In addition, in interactive query expansion, it has been shown that users cannot always effectively choose the best terms to be added to the query [15]. Finally, terms added to the query should be weighted in such a way that their importance in the context of the query will not modify the original concept expressed by the user.

Complex techniques using local context, like, for example, Local Context Analysis [28], enable to limit some of the drawbacks of automatic query expansion. It is not clear, however, how these techniques would perform on such a heterogeneous and incoherent collection like the Web.

3.3. Imaging

In 1986 Van Rijsbergen proposed the use in IR of a technique called *logical imaging* based on non-classical Conditional Logic [25]. Imaging enables the estimation of the RSV as $P(d \rightarrow q)$, where the semantics of the implication operator $\rightarrow$ does not need to be explicitly defined. In 1995 Crestani and Van Rijsbergen proposed and experimented with a retrieval model based on imaging [5]. This model was later generalised and experimented more thoroughly using a technique called *general logical imaging* [6]. This new technique is generalisation of the imaging technique proposed by Gärdenfors [11] that enables a more general transfer of the indexing weights than logical imaging.

Without entering into the details of this technique (details that can be found in the cited papers), the retrieval by general logical imaging model (RbGLI) uses term-term semantic similarity to direct the transfer of indexing weights at retrieval time from terms non present in the document to terms that are present. RbGLI transfers indexing weights to all terms present in the document with portions that are in decreasing order to the similarity between the "donor term" and the "recipient term". Terms that represent the same or similar concepts can then be accounted for even if they are not present in the document. Therefore, RbGLI attempts to solve the term mismatch problem without explicitly modifying the terms space or the query, but by changing the indexing weights of terms present in the document under consideration to account for terms that are similar and that have not been used to index the document.

Referring to the example of section 2 and looking at document d_2 , RbGLI would transfer the indexing weights of terms "Florence" and "France" to the term "Italy", and the indexing weight of "vineyard" to "wine". The amount of the transfer would be determined by the similarity between these terms. The value of $RSV(d_2, q)$ would then be determined using the classical way, but the term "wine" would now hold also the weight of term "vineyard", although this term is not present in the document. Unfortunately, such small example do not give full credit to the complexity of RbGLI. With a much larger term space and longer documents and queries the effects of the kinematics of indexing weights produced by RbGLI on the ranking of retrieved documents is much larger.

The major problem with RbGLI is that it is computationally very expensive [4]. Term similarity information is used at retrieval time to find for every term not present in the document those terms in the document to which its probability needs to be transfered and the relative amount involved in the transfer. This computation needs to be done for every document in the collection in order to produce a ranking.

4. Term Similarity

Most of the approaches to the term mismatch problem presented in the previous section assume the availability of a measure of the similarity between terms. A similarity measure between pairs of terms is necessary in order to build an automatic thesaurus or expand the query.

Measures of similarity have been studied in IR for a long time. They have been studied in the context of clustering [17], ranked retrieval [24], thesauri construction [23], and other applications. Although no single similarity measure has proved to be the best for any kind of application, most research in this area agrees on the fact that the estimate of a complete measure of similarity on the term space (i.e. for each pair of terms in the term space) is a very expensive business. However, the availability of very fast computers and more efficient algorithms for the evaluation of similarity in large term space is making this problem less and less serious.

In the context of this paper I will assume that a measure of similarity on the terms space can be evaluated. This is not an unreasonable assumption. First of all term similarity can be evaluated off-line and efficiently stored to be used at retrieval time. Second, the retrieval models presented in the following of this paper can also work with partial term similarity information, making it possible to tailor the evaluation of similarity to the available means (see section 5.5).

Let us suppose we have a measure of similarity that enables us to evaluate for each pair of terms in the term space T a real value which estimates how semantically close the terms are.

$$\forall(t_i, t_j) \in T, 0 \leq Sim(t_i, t_j) \leq 1$$

Such function Sim, should have the following properties:

1. $Sim(t_i, t_j) = 1$ iff $t_i = t_j$;
2. $Sim(t_i, t_j) \rightsquigarrow 1$ if t_i and t_j are semantically close, that is if t_i and t_j can be (and have been) used to express the same concepts;
3. $Sim(t_i, t_j) \rightsquigarrow 0$ if t_i and t_j are not semantically close, that is if t_i and t_j cannot be (and have not been) used to express the same concepts.

Of the above properties, property 1 is obvious, while properties 2 and 3 although intuitive, are difficult to verify for a given measure Sim. In fact, most measures of similarity developed in the field of IR attempt to follow these properties, but the information available for the estimate of the semantic similarity between terms is quite poor [14].

Most similarity measures used in IR attempt to estimate the semantic similarity between terms by looking at their pattern of occurrence in documents. Two terms are considered semantically similar if they tend to co-occur in the same context (i.e. a document, a paragraph, or a phrase). There are many recognised drawbacks to this assumption, but no one has been able to propose a better and still implementable approach[1]. I will not enter into a discussion about the plausibility of this approach. In the future we may have better ways of estimating the semantic similarity between terms, but for the time being I will make use of the state of the art in this area. Although the effectiveness of the models presented in the paper depends very much on the quality of the similarity measures, the proposed models could make use of any available similarity information of the term space.

5. Exploiting Term Similarity at Retrieval Time

Most classic IR models evaluate the RSV of a document with regard to a query using some variant of the following formula:

$$RSV(d, q) = \sum_{t \in q} w_d(t)\ w_q(t) \qquad (1)$$

where $w_d(t)$ is the indexing weight assigned to term t in the context of document d, and $w_q(t)$ is the indexing weight assigned to term t in the context of query q. The sum of the product of the indexing weights is performed over all terms occurring in the query, but since some query terms may not occur in the document (for which $w_d(t) = 0$), the sum is actually performed over all terms occurring in both the query and the document. Normalisation factors are often employed to take into consideration different document and query lengths.

Classic IR models fall into the term mismatch problem since they do not take into account that the same concept could be expressed using different terms in document and query.

Supposing we had similarity information on the term space, we could use this information to account for the term mismatch problem, exploiting such information at retrieval time for the evaluation of the RSV. In fact, if we take for example a query term for which we cannot find a matching document term, we could use similarity information to identify the semantically closest document terms. Alternatively, if we take a document term for which we cannot find a matching query term, we could use similarity information to identify the semantically closest query terms. In the following two sections I present a class of models that exploit term similarity information at retrieval time in this fashion. These models do not have most of the drawbacks of dimensionality reduction and query expansion, since they do not modify the term space or the original query, but they exploit at retrieval time the information provided by the term similarity.

5.1. The $q \triangleright d$ Models

If we consider the point of view of a query, then we could take a query term for which we cannot find a matching document term and look for semantically close document terms. In the presence of complete similarity information on the term space, we can easily

determine the closest document term, that is the document term for which we have the maximum value of similarity with the query term[2]. Supposing the similarity measure has been normalised in the range $[0, 1]$, we could introduce the similarity value in the computation of the RSV as follows:

$$RSV_{max(q \rhd d)}(d, q) = \sum_{t \in q} Sim(t, t^*)\ w_d(t^*)\ w_q(t) \tag{2}$$

where $t^* \in T$ is a document term for which $Sim(t, t^*) = max$, $w_d(t^*)$ is the indexing weight assigned to term t^* in the context of document d, $w_q(t)$ is the indexing weight assigned to term t in the context of query q, and $Sim(t, t^*)$ is the similarity value between t and t^*.

Formula 2 enables to consider non-matching terms in the evaluation of the RSV, since it considers all terms in the query even if they are not present in the document. Non-matching terms for which the similarity measure is maximum will contribute to the RSV in a measure that is proportional to their similarity value. The formula is a generalisation of formula 1, as it can be easily proved if we assume $Sim(t, t^*) = 1$ for $t = t^*$ and $Sim(t, t^*) = 0$ otherwise.

We can also consider the total value of the contribution of all non-matching terms in the evaluation of the RSV. In this case the formula needs to be changed into the following:

$$RSV_{tot(q \rhd d)}(d, q) = \\ = \sum_{t_k \in q} (\sum_{t_j \in d} Sim(t_k, t_j)\ w_d(t_j))\ w_q(t_k) \tag{3}$$

where symbols are defined as in formula 2. Again, formula 3 is a generalisation of formula 1 for the classic evaluation of the RSV.

Let us suppose, for example, that we have the following query and document:

$$\begin{aligned} q &= (t_1,\ t_3) \\ d &= (t_1,\ t_2,\ t_4) \end{aligned}$$

Let us also suppose that we have the following similarity information:

	t_1	t_2	t_3	t_4
t_1	1	.5	0	.1
t_2	.6	1	.2	.5
t_3	.8	.6	1	0
t_4	0	.4	.2	1

The above matrix S provides the similarity between each pair of terms in the term space, that is $Sim(t_i, t_j)$, where t_i is a term in row i and t_j is a term in column j. Notice that S satisfy the properties described in section 4, and it is also complete (i.e., there is a value for each pair (t_i, t_j)) and is non-symmetric (i.e., $Sim(t_i, t_j) \neq Sim(t_j, t_i)$).

Then:

$$RSV(d, q) = w_d(t_1) \cdot w_q(t_1)$$

while:

$$\begin{aligned} RSV_{max(q \rhd d)}(d, q) &= 1 \cdot w_q(t_1) \cdot w_d(t_1) + \\ &\quad +0.8 \cdot w_q(t_3) \cdot w_d(t_1) \end{aligned}$$

and

$$\begin{aligned} RSV_{tot(q \rhd d)}(d, q) &= 1 \cdot w_d(t_1) \cdot w_q(t_1) + \\ &\quad +0.5 \cdot w_q(t_1) \cdot w_d(t_2) + \\ &\quad +0.1 \cdot w_q(t_1) \cdot w_d(t_4) + \\ &\quad +0.8 \cdot w_q(t_3) \cdot w_d(t_1) + \\ &\quad +0.6 \cdot w_q(t_3) \cdot w_d(t_2) + \\ &\quad +0 \cdot w_q(t_3) \cdot w_d(t_4) \end{aligned}$$

As we can see from the example above, $RSV_{max(q \rhd d)}(d, q)$ and $RSV_{tot(q \rhd d)}(d, q)$ extend the classic $RSV(d, q)$ by considering also non-matching terms.

5.2. The $d \rhd q$ Models

If we consider the point of view of a document, then we have the $d \rhd q$ models. The evaluation of the RSV can then be obtained using the following formula for the evaluation of $RSV_{max(d \rhd q)}(d, q)$:

$$RSV_{max(d \rhd q)}(d, q) = \sum_{t \in d} Sim(t, t^*)\ w_d(t)\ w_q(t^*) \tag{4}$$

where $t^* \in T$ is a query term for which $Sim(t, t^*) = max$, $w_d(t)$ is the indexing weight assigned to term t

in the context of document d, $w_q(t^*)$ is the indexing weight assigned to term t^* in the context of query q, and $Sim(t, t^*)$ is the similarity value between t and t^*.

Alternatively, we can evaluate $RSV_{tot(d \rhd q)}(d, q)$ in a way similar to the one reported in formula 3:

$$RSV_{tot(d \rhd q)}(d, q) = \\ = \sum_{t_j \in d} (\sum_{t_k \in q} Sim(t_k, t_j)\ w_q(t_k))\ w_d(t_j) \tag{5}$$

Considering the same data of the example reported in section 5.1, here we would have the following results:

$$RSV(d, q) = w_d(t_1) \cdot w_q(t_1)$$

Notice that the classic $RSV(d, q)$ remains the same whatever the point of view taken.

$$\begin{aligned} RSV_{max(d \rhd q)}(d, q) \ &= 1 \cdot w_d(t_1) \cdot w_q(t_1) + \\ &+ 0.6 \cdot w_d(t_2) \cdot w_q(t_1) + \\ &+ 0.2 \cdot w_d(t_4) \cdot w_q(t_3) \end{aligned}$$

and

$$\begin{aligned} RSV_{tot(d \rhd q)}(d, q) \ &= 1 \cdot w_d(t_1) \cdot w_q(t_1) + \\ &+ 0 \cdot w_d(t_1) \cdot w_q(t_3) + \\ &+ 0.6 \cdot w_d(t_2) \cdot w_q(t_1) + \\ &+ 0.2 \cdot w_d(t_2) \cdot w_q(t_3) + \\ &+ 0 \cdot w_d(t_4) \cdot w_q(t_1) + \\ &+ 0.2 \cdot w_d(t_4) \cdot w_q(t_3) \end{aligned}$$

Notice that both $RSV_{tot(d \rhd q)}$ and $RSV_{max(d \rhd q)}(d, q)$ hold different results from the ones reported in section 5.1. However, it can easily be proved that with a symmetric similarity measure, i.e., with $Sim(t_i, t_j) = Sim(t_j, t_i)$, $forall(t_i, t_j) \in T$, we have the following equility: $RSV_{tot(q \rhd d)} = RSV_{tot(d \rhd q)}$. On the other hand, even with a symmetric similarity measure, we generally have $RSV_{max(q \rhd d)} \neq RSV_{max(d \rhd q)}$.

5.3. *Relation Between the $q \rhd d$ and $d \rhd q$ Models*

In a related area of research, aimed at modelling the IR retrieval process as logical model, Nie showed that the two conditionals $d \rightarrow q$ and $q \rightarrow d$ have a very interesting interpretation [16]. The conditional $d \rightarrow q$ expresses the *exhaustivity* of the document to a query, i.e. how much of a document content is specified by the query content. In fact $d \rightarrow q$ is intuitively equivalent to $d \subseteq q$. The conditional $q \rightarrow d$, instead, expresses the *specificity* of a document to a query, i.e. how much of a query content is specified in the document content. In fact, $q \rightarrow d$ is intuitively equivalent to $q \subseteq d$.

The models proposed in this paper can be interpreted in this way too. In fact, the $q \rhd d$ models, by taking the query point of view, measure how much of the query content is specified in the document. This is done in a complete way by $tot(q \rhd d)$, or in a partial way by $max(q \rhd d)$, considering only the most important contributions. This enables to measure the specificity of the document to the query. On the other hand, the $d \rhd q$ models, by taking the document point of view, measure how much of the document content is required by the query. Again, this is done in a complete way by $tot(d \rhd q)$, or in a partial way by $max(d \rhd q)$. This enables to measure the exhaustivity of the document to the query.

Nie proposed to combine the two measures to produce a "correspondence" measure between query and document. This measure should estimate the relevance of a document to a query. In this paper I did not follow this approach (yet), since at this stage I am mainly interested in analysing from an effectiveness perspective the difference between the models. A study of the possible combination of the $q \rhd d$ and $d \rhd q$ models will be carried out in the future.

5.4. *Relation Between the $q \rhd d$ and $d \rhd q$ Models and Other Approaches to the Term Mismatch Problem*

It is worth noticing that the proposed models differ in principle from other approaches to the term mismatch problem like, for example, the dimensionality reduction and query expansion techniques presented in section 3. While dimensionality reduction and query expansion techniques reshape the term space or the query, the models proposed in this paper do not alter the original term space or query. Instead, they use in the evaluation of the RSV additional information derived from term similarity of non-matching terms. Because of this fundamental differences, the effectiveness of the models here presented should not be compared with that of models of IR employing dimensionality reduction and query expansion techniques, but with that of classical models of IR. In fact, the $q \rhd d$ and $d \rhd q$ models could be used on top or in conjunction with

dimensionality reduction and query expansion techniques, and do not constitute an alternative to them.

5.5. Partial Similarity Information

In the above discussion I have supposed the availability of full similarity information, i.e. the availability of $Sim(t_i, t_j)$ for every pair of terms (t_i, t_j) in the term space T. This case is often unrealistic, especially for large term spaces, given the computational burden of the evaluation of $Sim(t_i, t_j)$. The evaluation and storing of complete similarity information is in fact a very expensive process. In most practical cases it makes sense to evaluate and store similarity information only for pair of terms that are most similar. These often are a very small subset of all terms in the term space. Moreover, this makes it possible to use a thesaurus for the evaluation of $Sim(t_i, t_j)$, since a thesaurus contains information regarding only the most similar pairs of terms. The formulas presented in the two previous sections do not need to be modified in case of availability of only partial similarity information. They can be used as their are.

In fact, let suppose, for example, that we have the following matrix S':

	t_1	t_2	t_3	t_4
t_1	1	.5	na	na
t_2	.6	1	.2	na
t_3	.8	.6	1	na
t_4	na	na	na	1

where na indicates a non-available similarity value. We can still evaluate the different RSVs by not considering the contributions of those pairs of terms for which we do not have similarity information. For example, using S' instead of S, we have:

$$\begin{aligned} RSV_{max(d \rhd q)}(d,q) &= 1 \cdot w_d(t_1) \cdot w_q(t_1) + \\ &\quad + 0.6 \cdot w_d(t_2) \cdot w_q(t_1) \end{aligned}$$

and

$$\begin{aligned} RSV_{tot(d \rhd q)}(d,q) &= 1 \cdot w_d(t_1) \cdot w_q(t_1) + \\ &\quad + 0.6 \cdot w_d(t_2) \cdot w_q(t_1) + \\ &\quad + 0.2 \cdot w_d(t_2) \cdot w_q(t_3) \end{aligned}$$

It should be noticed that the difference between the RSVs of the $q \rhd d$ and $d \rhd q$ models and the classical IR models become smaller the smaller is the amount of similarity information used. In fact, if the similarity information is completely non-available, the $q \rhd d$ and $d \rhd q$ models produce the same RSVs as with the classical IR models.

Moreover, in case of use of incomplete similarity information, the availability of a symmetric similarity measure does not assure that the equality $RSV_{tot(q \rhd d)} = RSV_{tot(d \rhd q)}$ holds.

6. Evaluation Framework

In order to carry out a preliminary evaluation of the retrieval effectiveness of the proposed models, a suitable evaluation framework needs to be devised. In such a framework it should be possible to evaluate the contribution of the use of term similarity information of non-matching terms to the retrieval effectiveness of classic IR models. To this purpose, three classic retrieval models have been compared with the equivalent $q \rhd d$ and $d \rhd q$ models, i.e. with models using the same weighting schemes. Any effectiveness improvements should then be attributed to the use of similarity between non-matching terms in the evaluation of the RSV.

6.1. Classic Retrieval Models

The computation of the classic $RSV(d,q)$ has been carried out according to the following three classic IR models:

- coordination level matching model;
- coordination level matching with idf weighting model;
- tf-idf model.

The *coordination level matching* simply counts the number of terms that are present in both document and query, that is:

$$RSV_{c.l.}(d,q) = \sum_{t \in q} I_d(t)$$

where $I_d(t)$ is 1 if t is present in d and 0 otherwise.

In the *coordination level matching with idf* the importance of the term t in the document collection, estimated by the inverse document frequency weight (idf), is added in the evaluation of the RSV:

$$RSV_{idf}(d,q) = \sum_{t \in q} idf(t)$$

where:

$$idf(t) = -log\ \frac{n}{N}$$

with n as the number of documents in which t occurs, and N as the number of documents in the collection.

Finally, we have the *tf-idf model*, one of the most classic model of IR, where the importance of the term t in the context of the document d is additionally taken into consideration:

$$RSV_{tf-idf}(d,q) = \sum_{t \in q} tf_d(t)\ idf(t)$$

where idf is defined as in the previous model and

$$tf_d(t) = \frac{\log(freq_d(t)+1)}{\log(length_d)}$$

with $freq_d(t)$ as the frequency of term t in document d, and $length_d$ as the number of unique terms in document d.

Each one of the above models makes use of additional information relative to term distribution (in the collection as a whole and/or in a document). Past studies of the retrieval effectiveness of the above models have shown that the full use of information about the term distribution, as achieved by the tf-idf model, gives higher retrieval effectiveness [24].

6.2. Similarity Measure

The models proposed in this paper make use of an additional form of information about terms in comparison to classic models: term similarity. The similarity between terms needs then to be evaluated for the term spaces of the document collections used. The problem of defining a measure of similarity between terms has been addressed by many researchers in the past, both in IR [27, 26, 23] and Natural Language Processing [3, 2]. It is very important to chose a good measure since much of effectiveness of the models proposed here depends on it. In the experiments reported in this paper I decided to use the *Expected Mutual Information Measure* ($EMIM$), because it has been used with success in the past by the author [7], and because it is a well accepted measure in Lexicography [3].

In Information Theory $EMIM(t_i,t_j)$ is often interpreted as a measure of the statistical information contained in t_i about t_j (or vice versa, it being a symmetric measure). The EMIM measure is defined as follows:

$$EMIM(t_i,t_j) = \\ = \sum_{t_i,t_j} P(t_i \in d, t_j \in d)\ log \frac{P(t_i \in d, t_j \in d)}{P(t_i \in d)P(t_j \in d)} \quad (6)$$

where t_i and t_j are any two terms of the term space T.

We can efficiently estimate EMIM between two terms using the technique proposed by Van Rijsbergen in [24]. This technique makes use of co-occurrence data that can be derived by a statistical analysis of the term occurrences in the collection. Some thresholding on the estimates has been used to make the computation fast and efficient [4], causing the similarity information to be incomplete.

6.3. Implementation of the $q \rhd d$ and $d \rhd q$ Models

Variants of the $max(d \rhd q)$ and $tot(q \rhd d)$ models taking into consideration different term distribution information were implemented to be comparable with the classic retrieval models. In particular, I defined and implemented the following models.

Max and *Tot*, defined respectively as:

$$RSV_{max(q \rhd d)}(d,q) = \sum_{t \in q} EMIM(t,t^*)\ I_d(t^*)$$

where t^* has been defined in section 5.1, and

$$RSV_{tot(q \rhd d)}(d,q) = \\ = \sum_{t_k \in q} (\sum_{t_j \in d} EMIM(t_k,t_j)\ I_d(t_j))\ I_q(t_k)$$

These models make no use of term distribution information and are comparable to the coordination level matching.

Max idf and *Tot idf*, defined respectively as:

$$RSV_{max(q \rhd d)}(d,q) = \sum_{t \in q} EMIM(t,t^*)\ idf(t^*)$$

and

$$RSV_{tot(q \rhd d)}(d,q) = \\ = \sum_{t_k \in q} (\sum_{t_j \in d} EMIM(t_k, t_j)\ idf(t_j))$$

Max idf and tot idf use only collection-wide term distribution information (the idf weight) and are comparable with the coordination level matching with idf.

Max tf-idf and *Tot tf-idf*, defined respectively as:

$$RSV_{max(q \rhd d)}(d,q) = \\ = \sum_{t \in q} EMIM(t, t^*)\ tf_d(t^*)\ idf(t^*)$$

and

$$RSV_{tot(q \rhd d)}(d,q) = \\ = \sum_{t_k \in q} (\sum_{t_j \in d} EMIM(t_k, t_j)\ tf_d(t_j)\ idf(t_j))$$

These last two models use all the available term distribution information, both at document level and at collection level. This makes them comparable to the tf-idf model.

6.4. Test Collections and Effectiveness Measure

The evaluation of the retrieval effectiveness of both classic and new models was carried out according to the traditional IR methodology. The retrieval performance were evaluated making use of some standard test collections (see table 1) and computed using the standard measure of precision and recall as defined in [24].

One clear drawback of this evaluation is the rather small size of the collections used, compared with those currently used in the context of TREC [13]. The main reason for the use of small collections, at this stage of the experimentation of the proposed models, is due to their computational burden. The examples reported in sections 5.1 and 5.2 give a clear indication of the size of the task. Moreover, the current implementation of the models has not been engineered for efficiency[3], since the main interest, at this stage of this research, is centred on the analysis of the behaviour of the models. Evaluations with larger test collections will be carried in the future.

Table 1. Test collections data

Data	*Cranfield*	*CACM*	*NPL*
documents	1400	3204	11429
queries	225	52	93
terms in doc.	5000	7121	7492
terms in query	274	356	337
avg. doc. length	53.61	24.26	19.96
avg. query length	8.95	11.5	7.15
avg. rel. doc.	7.28	15.31	22.41

7. Results and Analysis

The models presented in section 5 were tested and compared with classic IR models using the framework described in section 6. The experimental analysis carried out is too extensive to be reported in full in this paper. I will only report the most interesting findings.

Tables 2 and 3 report the results of the retrieval effectiveness of the $max(q \rhd d)$ and $tot(q \rhd d)$ models, in their various implementations, compared with those of classic IR models. The collection used is the CACM document collection. As it can be noticed, each $q \rhd d$ model performs almost always more effectively that the respective classic model. The same can be said for the $d \rhd q$ models, whose results are reported in Tables 4 and 5. Similar results were obtained for the other test collections, with bigger differences in effectiveness between new and classic models for the Cranfield collection, while smaller differences where found for the NPL collection. Since the $max(q \rhd d)$ and $tot(q \rhd d)$ models use the same term distribution information of the classic models, any increase in effectiveness can be attributed solely to the use contribution of the non-matching terms, ceteris paribus. Observe, also, how the improvement in average precision increases with the use of more and more term distribution information.

Table 6 reports a comparison of the effectiveness between the tf-idf model and the new models making use of all the available information on the terms distribution. Two things can be observed: (1) all the $q \rhd d$ and $d \rhd q$ models perform significantly better than the tf-idf model; (2) all the $q \rhd d$ and $d \rhd q$ models perform in a almost identical way.

The first observation is an experimental proof of the advantages of exploiting the similarity of non-matching terms at retrieval time. However, this result needs to be tested with a larger collection before taking it for certain. In IR there have been many instances

Table 2. Performance of $max(q \rhd d)$ using the CACM and different weighting schemas.

Rec. %	*Prec. %*					
	c.l.	max	idf	max idf	tf-idf	max tf-idf
10	44	46	41	53	56	70
20	33	32	34	43	49	59
30	21	25	27	38	41	53
40	16	21	22	30	33	46
50	13	16	18	21	26	39
60	11	13	15	16	21	33
70	08	08	09	13	14	26
80	07	07	07	11	11	19
90	04	04	05	08	06	10
100	03	04	04	05	03	07
Avg.	16.0	17.6	18.2	23.8	26.0	36.2
Impr.		01.6		05.6		10.2

Table 3. Performance of $tot(q \rhd d)$ using the CACM and different weighting schemas.

Rec. %	*Prec. %*					
	c.l.	tot	idf	tot idf	tf-idf	tot tf-idf
10	44	43	41	51	56	67
20	33	30	34	43	49	58
30	21	21	27	36	41	53
40	16	17	22	28	33	47
50	13	16	18	22	26	39
60	11	13	15	17	21	33
70	08	08	09	13	14	27
80	07	06	07	10	11	20
90	04	04	05	06	06	09
100	03	03	04	05	03	07
Avg.	16.0	16.1	18.2	23.1	26.0	36.0
Impr.		00.1		04.9		10.0

Table 4. Performance of $max(d \rhd q)$ using the CACM and different weighting schemas.

Rec. %	*Prec. %*					
	c.l.	max	idf	max idf	tf-idf	max tf-idf
10	44	44	41	51	56	70
20	33	30	34	42	49	59
30	21	21	27	37	41	52
40	16	16	22	29	33	47
50	13	14	18	22	26	40
60	11	11	15	17	21	34
70	08	08	09	13	14	26
80	07	06	07	10	11	19
90	04	04	05	07	06	09
100	03	03	04	05	03	07
Avg.	16.0	15.7	18.2	23.3	26.0	36.3
Impr.		-00.3		05.1		10.3

Table 5. Performance of $tot(d \rhd q)$ using the CACM and different weighting schemas.

Rec. %	*Prec. %*					
	c.l.	tot	idf	tot idf	tf-idf	tot tf-idf
10	44	43	41	52	56	70
20	33	30	34	43	49	59
30	21	21	27	36	41	52
40	16	17	22	28	33	47
50	13	16	18	22	26	40
60	11	13	15	17	21	34
70	08	08	09	13	14	26
80	07	06	07	10	11	19
90	04	04	05	07	06	09
100	03	03	04	05	03	07
Avg.	16.0	16.1	18.2	23.3	26.0	36.3
Impr.		00.1		05.1		10.3

of good experimental results that have not been confirmed by experimentation on larger test collections; the RbGLI model presented in section 3.3 is one of these instances [7].

The second observation can be explained considering the use of a symmetric and almost complete term similarity information. In this case, in fact, we have: $RSV_{tot(d \rhd q)}(d,q) = RSV_{tot(q \rhd d)}(d,q)$. The thresholding used in the estimate of EMIM [4] explains the little differences in the data observed. However, this does not explain why also the two models $max(q \rhd d)$ and $max(d \rhd q)$ performs also in an almost identical way. An explanation of this result can be found by looking at the EMIM values for the CACM collection. Given the size of the collection, the close domain of the documents, and the rather flat term distribution, many pairs of terms tend to have the same EMIM values. This results in a rather flat contribution of the similarity of the non-matching terms to the RSVs that flattened the differences between $max(q \rhd d)$ and $max(d \rhd q)$. This effect was also worsen by the availability of an almost complete term similarity information.

The availability of complete similarity information, apart from being unrealistic for larger term spaces, makes the situation uninteresting. It is more interesting to consider the case of availability of only partial or incomplete term similarity information. To simu-

late such a case in the current experimentation I used a thresholding function on the stored EMIM values. Only the n most similar terms where stored for each term, with $n \ll k$, where k is the total number of terms in the terms space T. In this case, if we consider, for example, the $max(q \rhd d)$ model and a query term t not present in the document, if we could not find a document term among the n most similar terms to t, we would not consider its contribution to the RSV.

Table 7 and 8 report the effectiveness figures for the different $q \rhd d$ and $d \rhd q$ models using respectively $n = 10$ and $n = 5$. It can be noticed that the effectiveness of the different models differ more largely, and the performance of $max(q \rhd d)$ and $max(d \rhd q)$ are different now. The fact that $max(q \rhd d)$ perform slightly better than $max(d \rhd q)$ can be explained by the relatively large size of queries compared to that of documents and by the fact that the term space is relatively small. In fact it is easier for a query term to find a similar document term than vice versa. I expect this result to reverse in larger test collections.

Notice also that $tot(q \rhd d)$ and $tot(q \rhd d)$ tend to perform better than $max(d \rhd q)$ and $max(q \rhd d)$ the larger the value of n, since it is easier to find more than one similar terms. Similar results were obtained with the other test collections. These results should also be valid for larger test collections, although the difference in performance may become smaller the larger the collection. It is hoped that experiments on a larger collection that are currently in progress will provide further evidence for some of the above experimental conclusions.

Table 6. Performance on the CACM document collection.

Rec. %	*Prec. %*				
	tf-idf	max (q ▷ d) tf-idf	tot (q ▷ d) tf-idf	max (d ▷ q) tf-idf	tot (d ▷ q) tf-idf
10	56	70	67	70	70
20	49	59	58	59	59
30	41	53	53	52	52
40	33	46	47	47	47
50	26	39	39	40	40
60	21	33	33	34	34
70	14	26	27	26	26
80	11	19	20	19	19
90	06	10	09	09	09
100	03	07	07	07	07
Avg.	26.0	36.2	36.0	36.3	36.3

Table 7. Performance on the CACM document collection with n=10.

Rec. %	*Prec. %*				
	tf-idf	max (q ▷ d) tf-idf	tot (q ▷ d) tf-idf	max (d ▷ q) tf-idf	tot (d ▷ q) tf-idf
10	56	63	65	60	66
20	49	56	56	51	57
30	41	50	52	49	50
40	33	44	45	40	45
50	26	37	37	33	38
60	21	32	31	29	32
70	14	25	25	21	24
80	11	17	18	12	17
90	06	08	07	06	07
100	03	05	05	04	05
Avg.	26.0	33.7	34.1	30.5	34.1

Table 8. Performance on the CACM document collection with n=5.

Rec. %	*Prec. %*				
	tf-idf	max (q ▷ d) tf-idf	tot (q ▷ d) tf-idf	max (d ▷ q) tf-idf	tot (d ▷ q) tf-idf
10	56	59	63	57	63
20	49	52	54	49	54
30	41	47	49	45	49
40	33	41	43	36	42
50	26	33	35	28	36
60	21	28	30	24	30
70	14	22	23	17	22
80	11	15	16	12	15
90	06	06	06	06	06
100	03	03	03	03	03
Avg.	26.0	30.6	32.2	27.7	32.0

8. Conclusions and Future Work

I presented a new class of IR models that exploit an available measure of similarity on the term space to include in the evaluation of the RSV also the contribution of non-matching terms. Although similar work has been carried out by other researchers, and in particular by Richardson and Smeaton [18], the exploitation of term similarity information at retrieval time has never been formalised into a class of models and experimented in the way presented in this paper.

The first experimental results on the effectiveness of the proposed models, also presented in this paper, seem

to prove that the proposed models are significantly more effective than classic IR models. Nevertheless, the experimentation carried out so far suffers from the following limitations:

- the term space is small;
- the similarity measure used (EMIM) is collection dependent, and symmetric.

To prove in a more definite way the effectiveness of the proposed models, the experimentation needs to be carried out using larger collections (and therefore larger terms spaces) and different measures of similarity, more representative of semantic similarity between terms than EMIM. To this purpose, I am currently:

- testing the model using a subset of the TREC collection composed of the full text of articles of the Wall Street Journal (1990-93);
- using EMIM similarity values obtained from the full TREC collection (applying a threshold on the minimum level of similarity to be considered) so that the values will be only partially collection dependent;
- studying the possibility of using a non-symmetric measure of similarity, like for example a thesaurus.

Once this further experimentation will be concluded it will be possible to assess the potentials of the proposed models in a more complete way.

Acknowledgements

I would like to thank Keith van Rijsbergen and Gianni Amati for discussing with me at length some of the ideas reported in this paper. I am also grateful to the anonymous reviewers for their useful comments on how to improve the presentation of these ideas.

Notes

1. Approaches using manually constructed thesauri to produce similarity information, like for example [18], rely on the availability of these sources of semantic similarity information, that cannot be always assured.
2. Ties at this stage, if they occur, are broken at random, to ensure the uniqueness of the term.
3. The models are currently implemented in Perl on top of an experimental IR system developed in house at Glasgow University for pure research purposes [20].

References

1. J. Aitchison and A. Gilchrist. *Thesaurus construction. A practical manual.* ASLIB, London, 2nd edition, 1987.
2. P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, and R.L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
3. K.W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *Proccedings of ACL 27*, pages 76–83, Vancouver, Canada, 1989.
4. F. Crestani, I. Ruthven, M. Sanderson, and C.J. van Rijsbergen. The troubles with using a logical model of IR on a large collection of documents. Experimenting retrieval by logical imaging on TREC. In *Proceedings of the TREC Conference*, pages 509–525, Washington D.C., USA, November 1995.
5. F. Crestani and C.J. van Rijsbergen. Information Retrieval by Logical Imaging. *Journal of Documentation*, 51(1):1–15, 1995.
6. F. Crestani and C.J. van Rijsbergen. Probability kinematics in Information Retrieval. In *Proceedings of ACM SIGIR*, pages 291–299, Seattle, WA, USA, July 1995.
7. F. Crestani and C.J. van Rijsbergen. A study of probability kinematics in rnformation retrieval. *ACM Transactions on Information Systems*, 16(3):225–255, 1998.
8. S. Deerwester, S.T. Dumais, G.W. Furnas, T. Landauer, and Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
9. E. Efthimiadis. Query expansion. *Annual Review of Information Science and Technology*, 31:121–187, 1996.
10. W.B. Frakes. Stemming algorithms. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms*, chapter 8. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.
11. P. Gärdenfors. *Knowledge in flux: modelling the dynamics of epistemic states*. The MIT Press, Cambridge, Massachusetts, USA, 1988.
12. D. Harman. Relevance feedback and other query modification techniques. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms*, chapter 11. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.
13. D. Harman, editor. *Proceedings of the Sixth Text Retrieval Conference (TREC-6)*, Gaithersburg, MD, USA, November 1997.
14. Tague-Sutcliffe. J. *Measuring information*. Academic Press, San Diego, CA, USA, 1995.
15. M. Magennis and C.J. van Rijsbergen. The potential and actual effectiveness of interactive query expansion. In *Proceedings of ACM SIGIR*, pages 324–332, Philadelphia, PA, USA, July 1997.
16. J.Y. Nie. An outline of a general model for Information Retrieval. In *Proceedings of ACM SIGIR*, pages 495–506, Grenoble, France, June 1988.
17. E. Rasmussen. Clustering algorithms. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms.*, chapter 16. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.
18. R. Richardson and A.F. Smeaton. Using wordnet in a knowledge-based approach to Information Retrieval. Technical Report CA-0395, School of Computer Applications, Dublin City University, Dublin, Ireland, 1995.
19. G. Salton. *Automatic information organization and retrieval*. McGraw Hill, New York, 1968.

20. M. Sanderson. System for Information Retrieval experiments (SIRE). Unpublished paper, November 1996.
21. A.F. Smeaton. Progress in the application of Natural Language Processing to Information Retrieval tasks. *The Computer Journal*, 35(3):268–278, 1992.
22. K. Sparck Jones. *Information Retrieval Experiments*. Butterworth, London, 1981.
23. P. Srinivadsan. Thesaurus construction. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms.*, chapter 9, pages 161–218. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.
24. C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.
25. C.J. van Rijsbergen. A non-classical logic for Information Retrieval. *The Computer Journal*, 29(6):481–485, 1986.
26. E.M. Voorhees. On expanding query vectors with lexically related words. In *Proceedings of the TREC Conference*, pages 223–232, Gaithersburg, MD, USA, November 1993.
27. S.K.M. Wong, Y.J. Cai, and Y.Y. Yao. Computation of term association by a Neural Network. In *Proceedings of ACM SIGIR*, Pittsburgh, PA, USA, July 1993.
28. J. Xu. *Solving the word mismatch problem through automatic text analysis*. Ph.D. Thesis, Department of Computer Science, University of Massachusetts, Amherst, MA, USA, May 1997.

Fabio Crestani is currently a visiting research fellow at the International Computer Science Institute in Berkeley, CA, USA. He was a “Marie Curie” research fellow at the Department of Computing Science of the University of Glasgow from 1997 to 1999, the time this work was carried out. Previously, he was assistant professor at the University of Padua, Italy, for five years. He holds a degree in Statistics from the University of Padua, and an MSc and PhD in Computing Science from the University of Glasgow.