INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

A Model for Combining Semantic and Phonetic Term Similarity for Spoken Document and Spoken Query Retrieval

Fabio Crestani*

TR-99-20

December 1999

Abstract

In classical 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 is known as "term mismatch". A similar problem can be found in spoken document retrieval and spoken query processing, where terms misrecognized by the speech recognition process can hinder the retrieval of potentially relevant documents. We will call this problem "term misrecognition", by analogy to the term mismatch problem.

Here we present two classes of retrieval models that attempt to tackle both the term mismatch and the term misrecognition problems at retrieval time using term similarity information. The models assume the availability of complete or partial knowledge of semantic and phonetic term-term similarity in the index term space.

*International Computer Science Institute, 1947 Center St. Suite 600, Berkeley, CA 94704, USA, email: fabioc@icsi.berkeley.edu

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 we are 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 the weighted terms. Document representations, constructed in a similar way, are then matched against the query representation. Documents are ranked according to how well their representation matches the query representation [23].

A fundamental problem of IR is *term mismatch*. A query is usually a short and incomplete description of the user information need. Users and authors of documents often use different terms to refer to the same concepts. This fact produces an incorrect relevance ranking of documents with regards to the information need expressed in the query.

A similar problem can be found in spoken document retrieval and spoken query processing, where terms misrecognized by the speech recognition process are found not matching in query and document representations. Naturally, this hinder the effectiveness of the IR system in a way similar to the term mismatch problem. I will call this problem *term misrecognition*, by analogy to the term mismatch problem.

This report addresses the term mismatch and the term misrecognition problems proposing a family of retrieval models that exploits the knowledge of semantic and phonetic 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 whose semantic and/or phonetic similarity are above a predefined threshold.

The report is structured as follows. In section 2, we show the importance of the term mismatch problem in IR, while in section 3 we introduce the "term misrecognition problem". In section 4, we present a number of solutions to the term mismatch problem that have been proposed in the past. A common graphical interpretation of these solutions is used to help understand their effect on the term space. In section 5, we address the significance of term similarity information on the term space and the cost of this knowledge. In section 6, we present two classes of retrieval models that exploit term similarity knowledge to tackle both the term mismatch and the term misrecognition problem. Section 7 presents some different forms of similarity that could be used with the retrieval models presented, while section 8 explain how these forms of similarity could be combined to tackle both problems at the same time. Section 9 discusses some issues related to the evaluation of the actual effectiveness of the proposed models. Section 10 concludes the report and outlines directions of future work.

2 The Term Mismatch Problem

The problem of representing the user information need and the document informative content is a very difficult one. Attempts to using advanced Natural Language Processing techniques or complex logical models have failed to solve the problem and IR is still using the classical technique of the "bag of terms". So, terms are automatically extracted from or manually assigned to documents or queries. This way of representing documents and queries is common to both the Vector Space model [21] and the Probabilistic model [23], 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 terms 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 [7]. 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 terms co-occurring in the query and the relevant documents [27]. 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 relevant documents, as the following example shows.

Let use 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 following 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 the documents and the 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, and is performed according to the model the IR system uses. The RSV is used to rank document and present them to the user. A IR system using a classical model of IR 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 a term in common with the query. These documents would then be ranked higher that document d_3, which does not have any term in common with the query. However, looking at the documents, we can clearly see that document d_1 is surely not relevant, since it deals with French wine. Moreover,

if we compare the informative content of 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 all of Italy. Document d_3 is fully relevant to the query, while document d_2 is only partially relevant. We are therefore inclined to assigned 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.

3 The Term Misrecognition Problem

The term misrecognition problem is analogous to the term mismatch problem. The problem is caused by the incorrect recognition of a term in a document or in a query. If a term t_j was actually present in both query and document, but was incorrectly recognized for t_k, then a number of potentially relevant documents containing t_j are not going to be retrieved, while a number of documents likely to be non-relevant and containing t_k are going to be retrieved. The incorrect ranking of these retrieved document is in direct relation to how many of these incorrect term recognitions are made, as we can easily imagine even without an example.

This problem can be found when documents (or queries) are not in a textual form directly comparable to queries (or document), like for example in spoken document retrieval and spoken query processing, or with OCRed documents of queries. It is plausible to assume that mistakes in speech recognition are related to how close words sound like [12], while in OCR it is related to how similar their shapes are [11]. In this report we will be mainly concerned with the term misrecognition problem found in spoken document retrieval and spoken query processing, since this work is important in the context of the SIRE Project. The main objective of the project is to enable a user to interact via voice (i.e. submit queries, commands, relevance assessments, and receive summaries of retrieved documents) with a probabilistic IR system over a low bandwidth communication line, like for example a telephone line [1].

4 Approaches to the Term Mismatch Problem

There are a number of approaches to solving the term mismatch problem. In the following of this section we will review some of these approaches showing how they attempt to tackle the problem. In this analysis we will look at their effects on the term space. We will argue that none of these approaches can completely solve the problem and each approach has its drawbacks.

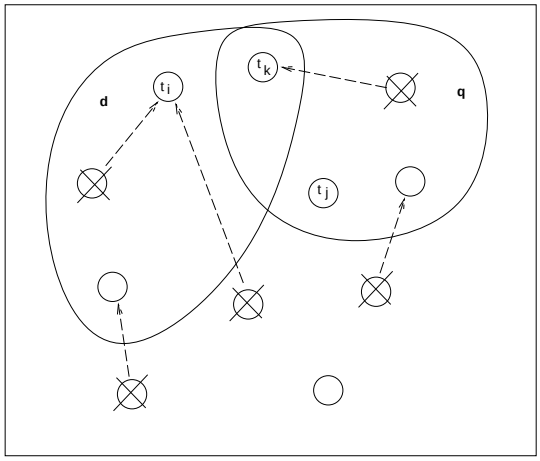

Figure 1: The effects of a dimensionality reduction of the term space.

4.1 Dimensionality Reduction

The most commonly used approach to the term mismatch problem consists of 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.

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

- stemming and conflation;
- manual thesauri;
- clustering or automatic thesauri;
- Latent Semantic Indexing.

The effects of these techniques on the term space are graphically depicted in figure 1. These techniques cause the removal of a number of terms from the term space. These terms are substituted by other terms, like for example, stems, thesauri classes, or cluster representative. Concepts can then be expressed using a limited number of terms, therefore reducing the effects of the term mismatch problem.

Some dimensionality reduction techniques, like for example stemming [9], term clustering [18], and Latent Semantic Indexing [7] have proved to be generally effective and are in use in many IR systems. Nonetheless, dimensionality reduction has a an important drawback: it causes a simplification of the "indexing vocabulary" that limits the expressiveness of the indexing language and can results in incorrect relevance rankings due to the incorrect classification of unrelated terms.

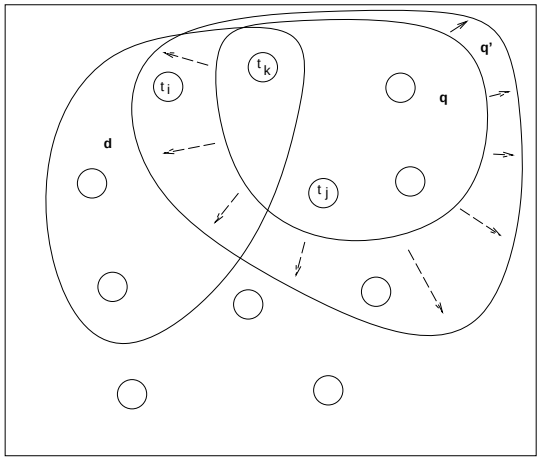

Figure 2: The query expansion process.

4.2 Query Expansion

Another approach to the term mismatch problem is query expansion. This approach consists of considering the query as a tentative definition of the concept the user in 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. The difficulty lies in finding the best terms to add and in weighting in a correct way their importance. A detailed treatment of this approach can be found in [8].

The two most important techniques for query expansion are:

- automatic query expansion;
- relevance feedback.

The first technique consists in automatically adding terms to the query by selecting those that are most similar to the ones used originally by the user. Some control can be left to the user on the choice of terms. The second technique 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.

The effect of query expansion on the the term space is graphically depicted in figure 2. The figure shows how the original query is expanded by adding terms that are similar to those originally present (similarity is represented in the figure as closeness in the space). Documents are then matched against the new expanded query.

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. Moreover, terms added to the query should be weighted in such a way that their importance

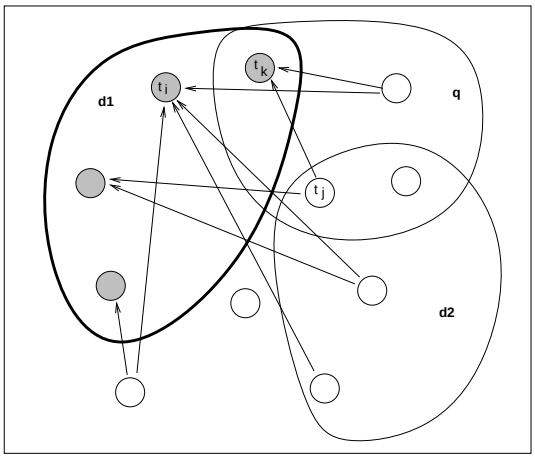

Figure 3: The imaging process on document d_1.

in the context of the query will not modify the original concept expressed by the user. Many different algorithms for automatic and interactive query expansion and relevance feedback have been proposed, but it is not clear yet which one is the most effective.

4.3 Imaging

In 1986 Van Rijsbergen proposed the use in IR of a technique called *logical imaging* based on non-classical Conditional Logic [24]. 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 not explicitly defined. In 1995 Crestani and Van Rijsbergen proposed and experimented with a retrieval model based on imaging [4]. This model was later generalized and experimented more thoroughly using a technique called *general logical imaging* [5]. This new technique is generalization of the imaging technique proposed by Gärdenfors [10] that enables a more general transfer of the indexing weights than logical imaging.

Without entering into the details of these techniques (details that can be found in the cited papers), the retrieval by general logical imaging model (RbGLI) use term semantic similarity to direct the transfer of indexing weights at retrieval time from terms not 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 in relation 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. Figure 3 depicts an example of the indexing weights kinematics produced by RbGLI.

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.

The major problem with RbGLI is that it is computationally very expensive [3]. It is in fact necessary to have a similarity value for every pair of terms in the term space. These values need to be used at retrieval time to find for every term not present in the document those terms to which its probability needs to be transfered and the relative amount involved in the transfer. We should also remember that this computation needs to be done for every document in the collection.

5 Term Similarity

All 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 in necessary in order to build an automatic thesaurus, expand the query, or perform indexing weights transfer.

Measures of similarity have been studied in IR for long time. They have been studied in the context of clustering [18], ranked retrieval [23], thesauri construction [22], and other areas of IR research. Although no single similarity measure has proved to be the best for any kind of application, most IR research 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 computationally 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 report we 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 then efficiently stored to be used at retrieval time. Second, the retrieval models presented in this report can also work with partial term similarity information, making it possible to tailor the evaluation of similarity to the available means (see section 6.4).

Let suppose we have a measure of similarity that enables us to evaluate for each pair of terms a real value which estimates how semantically close the terms are. We can normalize this values so that:

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

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, in this case t_i and t_j can be (and often have been) used to express the same concept;
3. $Sim(t_i, t_j) \rightsquigarrow 0$ if t_i and t_j are not semantically close, that is t_i and t_j cannot be (and have not been) used to express the same concept.

Models:	f. sim	maximum sim.	total sim.
$q \rhd d$	$f(q \rhd d)$	$max(q \rhd d)$	$tot(q \rhd d)$
$d \rhd q$	$f(d \rhd q)$	$max(d \rhd q)$	$tot(d \rhd q)$

Table 1: Examples of $q \rhd d$ and $d \rhd q$ models.

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 attempts to follow these properties, but the information available for the estimate of the semantic similarity between terms is quite poor. 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 recognized drawbacks to this assumption, but no one has been able to propose a better and still implementable approach. We 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 we will make use of the state of the art in this area. Although the effectiveness of the models presented in the report 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.

6 Exploiting Term Similarity

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

$$RSV(d,q)_c = \sum_{t \in (d \cap q)} w_d(t) \cdot w_q(t) \tag{1}$$

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 both the document and the query. Classical 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.

Table 1 presents an overview of the two classes of models one could envisage, depending on the perspective taken in 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

and use the similarity information in the evaluation of the RSV. 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 and use the similarity information in the evaluation of the RSV. The function f specifies the way the similarity information is use. Table 1 reports two examples of functions that have been successfully tested [2]. Other more complex uses of the similarity information or more complex combination of indexing weights can be devised.

In the following two sections we will present in more detail two classes of retrieval models that exploit term similarity information at retrieval time in this fashion.

6.1 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. We could then evaluate the RSV using the following general formula:

$$RSV_{f(q \triangleright d)}(d,q) = \phi_{t_k \in q}(w_d(t_j), w_q(t_k), f(Sim(t_j,t_k),t_k)) \tag{2}$$

where t_j and t_k are respectively a document and a query term, and f indicates the use we can make of the known similarity between the two terms.

Let us take, for example, the following f, indicated by max:

$$f(Sim(t_j,t_k),t_k) = max_{t_j} Sim(t_j,t_k) \cdot w_d(t_j)$$

The rationale behind this formula is that 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. Supposing the similarity measure has been normalized in the range $[0,1]$, we could introduce the similarity value in the computation of the RSV as follows:

$$RSV_{max(q \triangleright d)}(d,q) = \sum_{t \in q} Sim(t,t^*) \cdot w_d(t^*) \cdot w_q(t) \tag{3}$$

where t^* is a document term for which the value of $Sim(t,t^*)$ is maximum given the query term t, $w_d(t^*)$ 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 3 enables to consider non-matching terms in the evaluation of the RSV. Two non-matching terms for which the similarity measure is maximum, will contribute to the RSV in a way that is proportional to their similarity value. Formula 3 is a generalization of formula 1, as it can be easily proved if we assume $Sim(t_j,t_k) = 1$ if $t_k = t_j$ and $Sim(t_k,t_j) = 0$ otherwise.

Another possibility is, for example, to consider the total value of the contribution of all non-matching terms in the evaluation of the RSV. We can the use the following f, indicated hereby with tot:

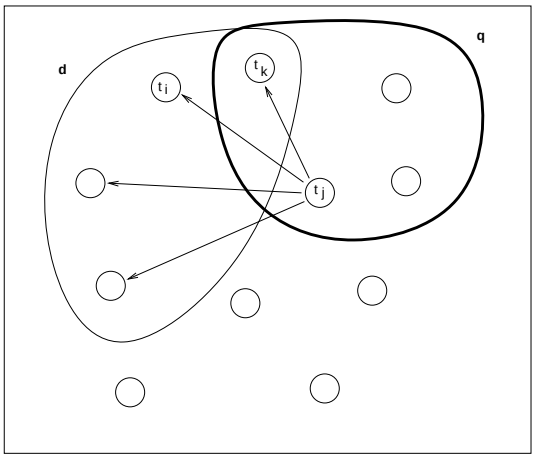

Figure 4: Graphical interpretation of $q \rhd d$.

$$f(Sim(t_j, t_k), t_k) = \sum_{t_j} Sim(t_j, t_k) \cdot w_d(t_j)$$

In this case we could evaluate the RSV using the following formula:

$$RSV_{tot(q \rhd d)}(d, q) = \sum_{t_k \in q} (\sum_{t_j \in d} Sim(t_k, t_j) \cdot w_d(t_j)) \cdot w_q(t_k) \tag{4}$$

where symbols are defined as in formula 3. Again, formula 4 is a generalization of formula 1.

The process of evaluation of the RSV in the $q \rhd d$ models is schematically depicted in figure 4.

Notice that the process depicted in figure 4 is just one of the possible ways of considering the contribution of non-matching terms in the evaluation of the RSV using term similarity. Other, more complex strategies can be considered, like for example the use of a term representative of the document content (e.g. the centroid) as depicted in figure 5, or using terms representative of topical concepts or structural sections of a document, as depicted in figure 6. We will not address these more complex models in this report.

Some of the approaches that this model makes available have been tried already (see for example [16], which resembles the models depicted in figure 5), but non-matching terms have never been considered in the evaluation of the RSV.

6.2 The d $\rhd$ q Models

If we consider the point of view of a document, then we have the $d \rhd q$ models defined in general terms as follows:

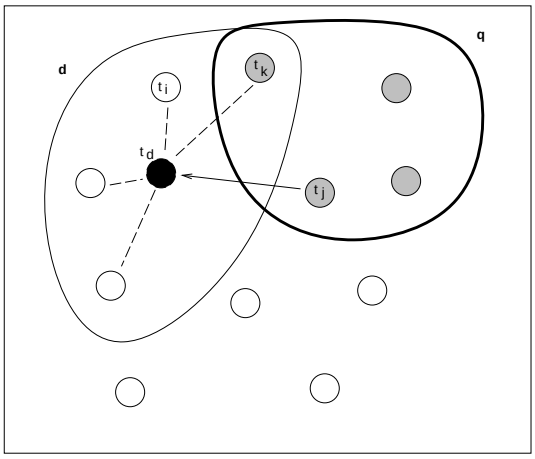

Figure 5: The $q \rhd d$ using a term representative of the document content.

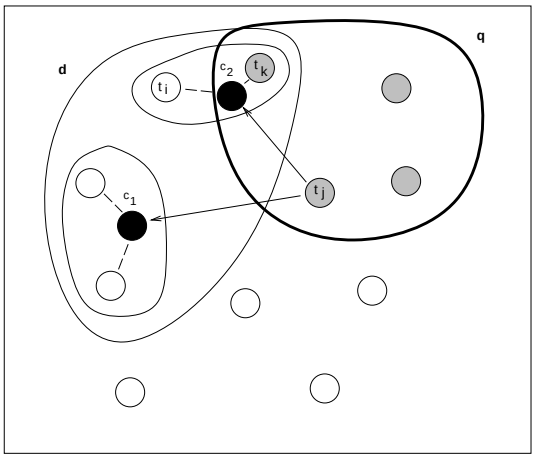

Figure 6: The $q \rhd d$ using terms representative of topical concepts or structural sections of a document.

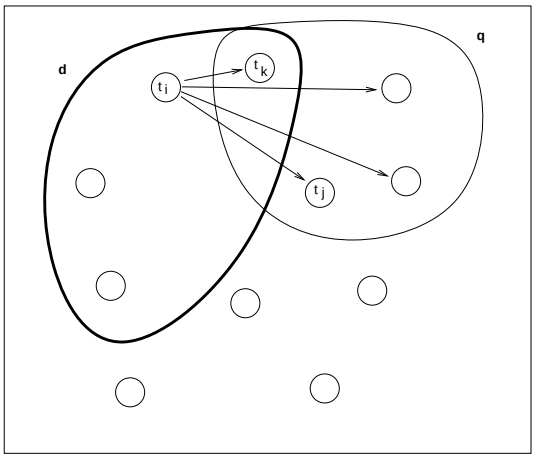

Figure 7: Graphical interpretation of $d \triangleright q$.

$$RSV_{f(d \triangleright q)}(d, q) = \phi_{t_j \in d}(w_d(t_j), w_q(t_k), f(Sim(t_j, t_k), t_j)) \tag{5}$$

The evaluation of the RSV could then be obtained, for example and mutatis mutandis, using the following formula for the evaluation of $RSV_{max(d \triangleright q)}(d, q)$:

$$RSV_{max(d \triangleright q)}(d, q) = \sum_{t \in d} Sim(t, t^*)\ w_d(t) \cdot w_q(t^*) \tag{6}$$

where t^* is a query term for which the value of $Sim(t, t^*)$ is maximum given the document term t, $w_d(t)$ 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 could evaluate $RSV_{tot(d \triangleright q)}(d, q)$ in a way similar to the one reported in formula 4. As can be noticed, the only difference between the $q \triangleright d$ and the $d \triangleright q$ models is the point of view taken:

$$RSV_{tot(d \triangleright q)}(d, q) = \sum_{t_j \in d} (\sum_{t_k \in q} Sim(t_k, t_j) \cdot w_q(t_k)) \cdot w_d(t_j) \tag{7}$$

The process evaluation of the RSV in the $d \triangleright q$ class of models is schematically depicted in figure 7. As can be noticed the only difference between the $q \triangleright d$ and the $d \triangleright q$ models is the point of view taken.

For this class of models more complex ways of considering the contribution of non-matching terms in the evaluation of the RSV can also be devised, in ways similar to those already discussed for the $q \triangleright d$ class of models. These will not be addressed here.

6.3 Relation Between q $\triangleright$ d and d $\triangleright$ q Models

In a related area of research, aimed at modeling the IR retrieval process as logical model, Wong and Yao demonstrated that most current IR models can be explained in terms of the formula $P(E \rightarrow H)$ that is evaluated as $P(H \mid E)$ [25, 26]. The latter formula evaluates the degree of confirmation (or belief, according to the view taken) of the sentence H given evidence E. Conventional IR models can be obtained by associating either d or q to H or E, and by defining different ways of evaluating the probabilities via probabilistic inference on a concept space. Earlier, Nie showed that the two conditionals $d \rightarrow q$ and $q \rightarrow d$ have a very interesting interpretation in the context of IR [14]. 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 report can be interpreted in this way too.

In fact, the $q \triangleright d$ 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 $tot(q \triangleright d)$, or in a partial way by $max(q \triangleright d)$, considering only the most important contributions. So, $q \triangleright d$ enables to measure the specificity of the document to the query. On the other hand, the $d \triangleright q$ 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 $tot(d \triangleright q)$, or in a partial way by $max(d \triangleright q)$. So, $d \triangleright q$ enables to measure the exhaustivity of the document to the query.

To summarize:

- $(q \triangleright d) \overset{P.W.S.}{\equiv} (q \rightarrow d) \overset{C.L.}{\equiv} (q \subseteq d)$ = *specificity of d to q* = how much of the query content is specified in the document content;
- $(d \triangleright q) \overset{P.W.S.}{\equiv} (d \rightarrow q) \overset{C.L.}{\equiv} (d \subseteq q)$ = *exhaustivity of d to q* = how much of the document content is specified by the query content.

Different applications require different levels of specificity and exhaustivity. Specificity is precision oriented, while exhaustivity is recall oriented.

Nie proposed to combine the two measures to produce a "correspondence" measure between query and document. This measure should estimate in a more complete way the relevance of a document to a query. In this report we did not follow this approach (yet). A study of the possible combination of the $q \triangleright d$ and $d \triangleright q$ models will be carried out in the future.

6.4 Partial Similarity Information

In the above discussion we have supposed the availability of full similarity information. This case is often unrealistic, especially for large term spaces, given the computational

burden of the evaluation of $Sim(t_i, t_j)$ for every pair of terms in the term space. The evaluation and the storing of complete similarity information is in fact a very expensive process. In most practical cases it makes more sense to evaluate and store similarity information only for pairs of terms that are most similar. These often are a very small subset of all terms in the term space. 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.

Moreover, it can be easily proved that, for most f:

$$RSV_{f(q \triangleright d)} \rightarrow RSV_c \quad \text{for} \quad \mathbf{SS} \rightarrow \mathbf{n.a.}$$

and

$$RSV_{f(d \triangleright q)} \rightarrow RSV_c \quad \text{for} \quad \mathbf{SS} \rightarrow \mathbf{n.a.}$$

where **SS** is the similarity matrix, and **n.a.** is the matrix on all $n.a.$ values, where with $n.a.$ we indicate the non-availability of similarity information for a pair of terms.

7 Different Forms of Similarity

So far we have always talked about similarity from a generic point to view, meaning any possible way of establishing a metric in the term space that has the properties discussed in section 5. However, there are two forms of similarity that it would be very important to combine for spoken document and spoken queries retrieval:

- *semantic similarity*;
- *phonetic similarity*.

These two type of similarity are related to the two main types of uncertainty present in spoken document retrieval and spoken query processing and that may cause the term mismatch problem: "information need uncertainty" and "recognition uncertainty".

Information need uncertainty relates to the word sense ambiguity present in the natural language and to the difficulties that the user has in expressing his information need. This is the typical form of textual IR uncertainty, which the term mismatch problem usually refers to. Recognition uncertainty can be related to a term misrecognition problem, due to the misrecognition of a term in the speech recognition process. Both types of uncertainty are present in spoken document and spoken queries retrieval.

The next sections will examine how we can evaluate semantic and phonetic term similarity. Later I will explain how they can be combined together in a retrieval model.

7.1 Semantic Similarity

There are many different techniques for estimating semantic similarity between terms (henceforth indicated with $SSim$). Semantic similarity may be estimated from external knowledge, like for example a thesaurus [20] or a dictionary [19]. It can also be estimated from the document collection itself, given a large enough corpus. Most of these techniques are based on statistical analysis of the patterns of occurrence of terms in the documents [22, 17, 15, 6].

One of the most often used measure of semantic similarity is the Expected Mutual Information Measure (EMIM), a well accepted measure in Lexicography. EMIM is defined as follows:

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

where t_i and t_j are any two terms of the term space T. The EMIM between two index terms is often interpreted as a measure of the statistical information contained in one term about the other (and vice versa, it being a symmetric measure). For our purposes we can estimate EMIM using the technique proposed by Van Rijsbergen in [23, p.130], which rely on the availability of co-occurrence data that can be derived from a statistical analysis of the term occurrences in the collection.

$SSim(t_i, t_j)$ can easily be normalized in $[0, 1]$ once its maximum and minimum values for the available data have been found. The important point is that any measure of semantic similarity, can be used with the models proposed in this report. The better the measure, the better the performance of the models. We will indicate semantic similarity with $SSim$ in the rest of this report.

7.2 Phonetic Similarity

Phonetic similarity (henceforth indicated with $PSim$) can be estimated using a error recognition confusion matrix (ERCM). A ERCM is a matrix that reports for each element in row and column the number of times one has been mistaken for the other [13]. In other words, if we call reference (r) the real value of the element being observed and hypothesis (h) the value actually observed for that element, $ERCM(r, h)$ gives the number of times r is confused with h.

When elements of the ERCM are terms, the matrix can be more easily done using at a phonetic level, instead of with terms [13]. The number of terms in the term space is too large to produce a ERCM matrix. Moreover, the matrix would be very sparse. On the other hand, there exist a limited number of phones (the exact number depends on the phonetic system used), making it easier to build such a matrix. Figure 8 reports an example of phones confusion matrix (courtesy of Kenny Ng, MIT).

With and ERCM built at phones level, and assuming that phones comprising each term are independent, we can evaluate $PSim(t_i, t_j)$ using a dynamic programming procedure:

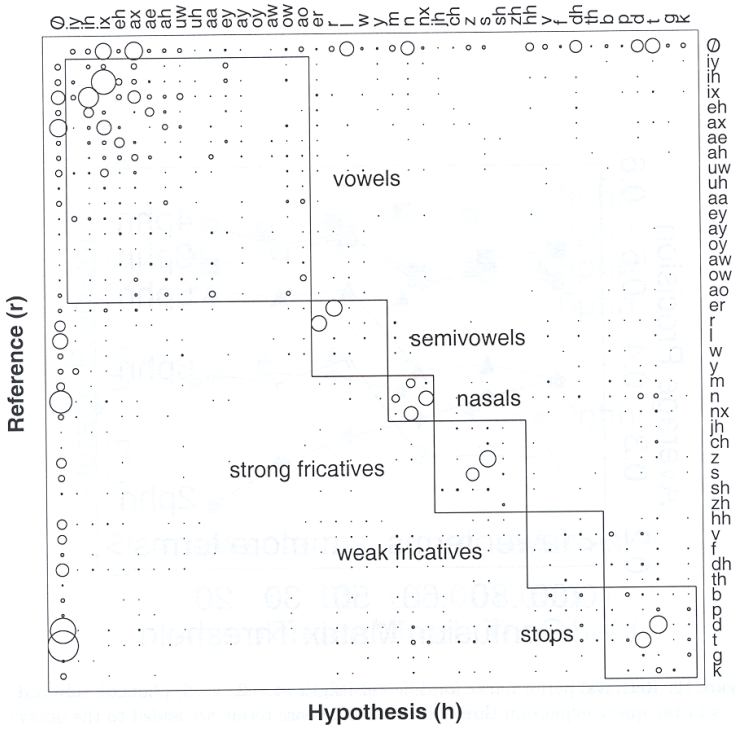

Figure 8: Example of confusion matrix (courtesy of Kenny Ng, MIT), with $r = 0$ and $h = 0$ corresponding respectively to insertion and deletion errors

$$PSim(t_i, t_j) = A(l_i, l_j)$$

where l_i and l_j are respectively the length of t_i and t_j, and A is the $l_i \times l_j$ dynamic programming matrix evaluated recursively as in the following formula:

$$A(m,n) = \begin{cases} 1 & m = 0, n = 0 \\ A(0, n-1) \cdot C_n(0, j[n-1]) & m = 0, n > 0 \\ A(m-1, 0) \cdot C_n(i[m-1], 0) & m > 0, n = 0 \\ max \begin{cases} A(m-1, n) \cdot C_n(i[m-1], 0) \\ A(m-1, n-1) \cdot C_n(i[m-1], j[n-1]) \\ A(m, n-1) \cdot C_n(0, j[n-1]) \end{cases} & m > 0, n > 0 \end{cases} \quad (8)$$

where $C_n(r, h)$ is the probability (actually the normalized frequency) of observing h given that it really was r. This is obtained as:

$$C_n(r, h) = \frac{ERCM(r, h)}{\sum_{k \in \{h\}} ERCM(r, k)}$$

where $ERCM(r, h)$ is a value of the ERCM matrix with row 0 for insertion and column 0 for deletion errors. Notice that with this estimation procedure we can interpret $PSim(t_i, t_j) \approx P(t_i \mid t_j)$.

Other techniques based for example of Hidden Markov Models can also be used to estimate $PSim$ [12].

8 Combination of Semantic and Phonetic Similarity for Spoken Document and Spoken Query Retrieval

We can easily adapt the models presented in section 6 to dealing with the misrecognition problem. This can be done by using phonetic similarity, instead of, or in conjunction with, semantic similarity. So for example if we are only considering phonetic similarity, formula 3 can be rewritten as:

$$RSV_{max_p(q \triangleright d)}(d, q) = \sum_{t \in q} PSim(t, t^{p*}) \cdot w_d(t^{p*}) \cdot w_q(t) \quad (9)$$

where t^{p*} is a document term for which the value of $PSim(t, t^{p*})$ is maximum given the query term t, $w_d(t^{p*})$ is the indexing weight assigned to term t^{p*} in the context of document d, $w_q(t)$ is the indexing weight assigned to term t in the context of query q, and $PSim(t, t^{p*})$ is the phonetic similarity value between t and t^{p*}.

While if we consider only the semantic similarity, the formula for $RSV_{max_s(q \triangleright d)}$ can be written using $SSim$ instead of $PSim$. On the other hand, if both measures of

similarity were normalized, we could consider the combination of both the semantic and phonetic similarity. So, for example, we could write:

$$\begin{aligned} & RSV_{max_{ps}(q \triangleright d)}(d,q) = \\ & \quad = \tfrac{1}{2}\ (RSV_{max_{p}(q \triangleright d)}(d,q) + RSV_{max_{s}(q \triangleright d)}(d,q)) \\ & \quad = \tfrac{1}{2} \sum_{t \in q}\ (PSim(t,t^{p*}) \cdot w_d(t^{p*}) + SSim(t,t^{s*}) \cdot w_d(t^{s*}))\ w_q(t) \end{aligned} \tag{10}$$

where the properties described in section 6.4 are preserved.

Analogously, $RSV_{tot_{ps}(q \triangleright d)}(d,q)$ can be easily evaluated as:

$$RSV_{tot_{ps}(q \triangleright d)}(d,q) = \sum_{t_k}(\sum_{t_j} \frac{PSim(t_k,t_j) + SSim(t_k,t_j)}{2} \cdot w_d(t_j)) \cdot w_q(t_k) \tag{11}$$

Formulas 10 and 11 are very simplistic combinations of semantic and phonetic similarities, and other more complex combinations can be devised. We also leave to the reader the task of deriving $RSV_{max_{ps}(d \triangleright q)}$ and $RSV_{tot_{ps}(q \triangleright d)}$ where semantic and phonetic similarity are combined.

9 Evaluation Issues

The use of semantic similarity to approach the term mismatch problem has already been experimented with "relative" success. Some results of this experimentation can be found in [2]. It has to be recognized that the experimental results obtained so far do not provide definite evidence of the real effectiveness of the proposed models. In fact, there are a number of limitations in the experiments carried out, in particular:

- the size of the test collections employed in the experimentation is considerably smaller than the size of the collections used in most current IR experimentation (e.g. TREC);
- the benchmark used in the experimentation is the classical vector space model, without the most advanced weighting and normalization techniques;
- a number of approximations where employed in the evaluation of the semantic similarity which limit its full potentials.

Further experimentation is currently under way to overcome the above limitations. The major problem encountered in the experimentation is the computational burden of the models which needs to be further addressed before starting experimenting with the combination of semantic and phonetic similarity. It is easy to envisage that the use of complex combinations of semantic and phonetic similarity will make the computational burden heavier and the evaluation of the proposed models more difficult.

10 Conclusions and Future Work

In this report we present a model for dealing with the term mismatch and the term misrecognition problems in spoken document retrieval and and spoken query processing. An initial experimental investigation with a small test collection is currently being carried out. The experimental results will provide useful feedback on the effectiveness of the proposed models and on how to effectively combine semantic and phonetic similarity.

References

[1] F. Crestani. Sonification of an Information Retrieval environment: design issues. In *International Forum on Multimedia and Image Processing*, pages 789–794, Anchorage, Alaska, USA, May 1998.

[2] F. Crestani. Exploiting the similarity of non-matching terms at retrieval time. *Journal of Information Retrieval*, 1999. In press.

[3] 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.

[4] F. Crestani and C.J. van Rijsbergen. Information Retrieval by Logical Imaging. *Journal of Documentation*, 51(1):1–15, 1995.

[5] 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.

[6] C.J. Crouch and B. Yang. Experiments in automatic statistical thesaurus construction. *Information Processing and Management*, pages 77–88, June 1992.

[7] 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.

[8] E. Efthimiadis. Query expansion. *Annual Review of Information Science and Technology*, 31:121–187, 1996.

[9] 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.

[10] P. Gärdenfors. *Knowledge in flux: modelling the dynamics of epistemic states.* The MIT Press, Cambridge, Massachusetts, USA, 1988.

[11] J. Heid. Getting started with optical character recognition. *MacWorld*, pages 77–83, October 1990.

[12] J.A. Markowitz. *Using speech recognition.* Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[13] K. Ng. Towards robust methods for spoken document retrieval. In *Proceedings of Int. Conf. on Spoken Language Processing*, volume 3, pages 939–942, Sydney, Australia, November 1998.

[14] J.Y. Nie. An outline of a general model for Information Retrieval. In *Proceedings of ACM SIGIR*, pages 495–506, Grenoble, France, June 1988.

[15] Y.C. Park and K-S. Choi. Automatic thesaurus construction using bayesian networks. *Information Processing and Management*, 32(5):543–553, 1996.

[16] Y. Qiu and H.P. Frei. Concept based query expansion. In *Proceedings of ACM SIGIR*, pages 160–171, Pittsburgh, PA, USA, June 1993.

[17] V.V. Raghavan and G.S. Jung. A machine learning approach to automatic pseudo-thesaurus construction. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, Poster Session Program, pages 111–121, Charlotte, NC, USA, October 1989.

[18] 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.

[19] D.R. Raymond and F.Wm. Tomps. Hypertext and the Oxford English dictionary. *Communications of the ACM*, 31(7):871–879, 1988.

[20] 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.

[21] G. Salton. *Automatic information organization and retrieval.* McGraw Hill, New York, 1968.

[22] 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.

[23] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, London, second edition, 1979.

[24] C.J. van Rijsbergen. A non-classical logic for Information Retrieval. *The Computer Journal*, 29(6):481–485, 1986.

[25] S.K.M. Wong and Y.Y. Yao. A probabilistic inference model for Information Retrieval. *Information Systems*, 16(3):301–321, 1991.

[26] S.K.M. Wong and Y.Y. Yao. On modelling Information Retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, 1995.

[27] 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.