Experiments in Term Expansion Using Thesauri in Spanish

Ángel F. Zazo, Carlos G. Fígueroa, José L.A. Berrocal, Emilio Rodríguez, and Raquel Gómez

Grupo de Recuperación Automatizada de la Información (REINA)
Dpto. Informática y Automática - Universidad de Salamanca
37008 Salamanca - SPAIN
http://reina.usal.es

Abstract This paper presents some experiments carried out this year in the Spanish monolingual task at CLEF2002. The objective is to continue our research on term expansion. Last year we presented results regarding stemming. Now, our effort is centred on term expansion using thesauri. Many words that derive from the same stem have a close semantic content. However other words with very different stems also have semantically close senses. In this case, the analysis of the relationships between words in a document collection can be used to construct a thesaurus of related terms. The thesaurus can then be used to expand a term with the best related terms. This paper describes some experiments carried out to study term expansion using association and similarity thesauri.

1 Introduction

A major problem in word based information retrieval (IR) systems is the word-mismatch or vocabulary problem [1]. Lexical phenomena such as synonymy and polysemy mean that the same concept can be expressed with different words and the same word can appear in documents that deal with different topics. The performance of IR systems depends on the number of query terms. The problem is less severe for long queries because more index terms are included, and thus there is more possibility to find query terms in the relevant documents. In addition, short queries are poor for recall and precision: they do not take into account the variety of words used to describe a topic, and they are too broad to retrieve relevant documents on specific topics. Our interest is centred on queries with very few terms. This kind of query has special importance in Web search engines, where queries are typically of one to three terms in length [2].

Many techniques have been used to try to reduce this problem, inter alia automatic query expansion. Query expansion methods have been investigated for almost as long as the history of information retrieval. This technique involves two basic steps: expanding the original query with new terms, and reweighting the terms in the expanded query. With query expansion, the retrieval performance can improve but the computational cost or the response time may increase.
order to expand the query, words or phrases with similar meaning to those of
the initial query must be added. There are several possible approaches to this
task, and the use of a thesaurus is the most important. A thesaurus is a system
composed of words or phrases and of a set of related words for each of them. In
information retrieval, thesauri are used to help with with the query formation
process. Likewise, stemming can be thought of as a mechanism for query expansion,
and can be seen as similar to using a thesaurus. Some stemmers can be
created or modified using the same techniques as for thesaurus construction [3].

This paper explores the association and similarity thesauri approach to term
expansion. Additionally, an experiment in term expansion using stemming has
been carried out. This is useful for comparison purposes with our last year’s
paper. We assume the well-known vector space model, but queries are first ex-

tended to help improve retrieval performance.

2 Stemming

The impact of our stemmers for the Spanish monolingual track at CLEF2001
was presented in [4]. For all query fields (title + narrative + descriptive), the
improvement is only about 3% for inflectional stemming over non-stemming.
Derivational stemming is even a little worse than no stemming. At CLEF2002
we have corrected small bugs in our stemmers, and only inflectional stemming
was applied. The objective is to measure the improvement, taking into account
only the ES-title field of the queries. We then compare the results with those
derived from applying a thesauri.

3 Thesaurus

One of the most important methods for query expansion is the application of a
thesaurus. A general thesaurus could be used, but this usually does not give good
results (e.g. [3]). The relations among entries in a general thesaurus are usually
not valid in the scope of the document collection being used. Better results are
obtained if thesauri, or other expansion techniques, are constructed from the
document collection. When the thesaurus is constructed automatically, without
additional user feedback information, several approaches can be used [6]: auto-
matic term classification (term co-occurrences statistics) [7], use of document
classification [8], concept based query expansion [9,10], phrase-finder expansion
[11] or expansion based on syntactic information [12]. We have tested approaches
that use association (term co-occurrence statistics) and similarity (conceptually-
based query expansion) thesauri, because they are relatively simple and effective.

A thesaurus is a matrix that measures term relations [13]. This matrix is
used to expand the query terms with related terms. The matrix can be seen
as a semantic description of terms, which reflects the impact of terms in the
conceptual descriptions of other terms [14]. We note two fundamental aspects to
apply the matrix in our tests. First, the expansion is made only with best related
terms. No threshold values are taken into consideration: terms with highly related
values are selected for each original query term. Secondly, the entire query, i.e.,
the query concept [10], is taken into account. The top ranked terms for the entire
query are considered.

We must state at this point that results for term expansion using thesauri may
be differ considerably. Several of the papers cited previously show acceptable results.
On the other hand, for example, [15] offers perhaps the most critical study
of term co-occurrence based models. Indeed, earlier studies [16] showed even better
results with randomly selected terms than when using term co-occurrence
statistics. However, we have obtained satisfactory results, which perhaps helps
to confirm the two observations just made.

3.1 Association Matrix

Term co-occurrence has been frequently used in IR to identify some of the semantic
relationships that exist among terms. In fact, this idea is based on the
Association Hypothesis [17, p.104]. If query terms are useful to identify relevant
and non relevant documents, then their associated terms will also be useful, and can
be added to the original query.

Several coefficients have been used to calculate the degree of relationship
between two terms. All of them measure the number of documents in which they
occur separately, in comparison with the number of documents in which they co-occur. In our tests three well-known coefficients have been used [9]:

\[
\text{Tanimoto}(t_i, t_j) = \frac{c_{ij}}{c_i + c_j - c_{ij}}
\]

\[
\text{Cosine}(t_i, t_j) = \frac{c_{ij}}{\sqrt{c_i \cdot c_j}}
\]

\[
\text{Dice}(t_i, t_j) = \frac{2 \cdot c_{ij}}{c_i + c_j}
\]

where \(c_i\) and \(c_j\) are the number of documents in which terms \(t_i\) and \(t_j\) occur,
respectively, and \(c_{ij}\) is the number of documents in which \(t_i\) and \(t_j\) co-occur. The
coefficients have values between 0 and 1: if two terms occur only in the same
documents, the associated coefficient is 1. If there is no document in which they
co-occur, the value is 0.

3.2 Similarity Matrix

The similarity matrix measures term-term similarities, instead of term-term co-
ocurrences. To compute the values of the elements, each term is indexed by
the documents in which it occurs, i.e., the roles of terms and documents are
switched. This theory is fully explained in [10]. The broad outlines are: first, each
term in the document vector space is represented by a vector, whose elements are
computed adapting the normalized \(tfidf\) weighting scheme to this new situation.
We use the same calculus as in [10]. Second, to compute the similarity between
two terms the simple scalar product of vectors is used. We have also used it. Computation for every term produces the similarity matrix.

Both the association and the similarity matrices produce comparable results. Table 1 shows the 20 best related terms with the Spanish word *terremoto*.

### 3.3 Expansion of the Query

The aim of using the association or the similarity matrices is to expand the entire query, not only separate terms. A term can be included in the list of expanded terms only if it has a high relationship value with all query terms. To obtain the expanded terms with the highest potential, each term of the original query should be expanded with all related terms. A new value is computed multiplying the weight of the original query term by the corresponding association/similarity value of related terms. For all original query terms, all values are added for each term that could be included in the list of expanded terms. The sum represents the value of the relationship of that term with the entire query. The list of expanded terms is then sorted in decreasing order. Only top ranked terms are used to expand the original query.

Finally, it is necessary to calculate the weight of the term added to the query. Obviously this depends on the relationship value with the entire query. In [10]

<table>
<thead>
<tr>
<th>Tanimoto</th>
<th>Association Matrix</th>
<th>Cosine</th>
<th>Dice</th>
<th>Similarity Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>terremoto</td>
<td>1.0000 terremoto 1.0000</td>
<td>terremoto 0.6192</td>
<td>richter 0.5773</td>
<td></td>
</tr>
<tr>
<td>richter</td>
<td>0.4058 richter 0.5827</td>
<td>richter 0.5188</td>
<td>seismo 0.4833</td>
<td></td>
</tr>
<tr>
<td>seismo</td>
<td>0.3502 seismo 0.5288</td>
<td>seismo 0.4375</td>
<td>epicentro 0.4375</td>
<td></td>
</tr>
<tr>
<td>epicentro</td>
<td>0.2800 epicentro 0.4569</td>
<td>epicentro 0.3395</td>
<td>temblor 0.3395</td>
<td></td>
</tr>
<tr>
<td>temblor</td>
<td>0.2045 temblor 0.3626</td>
<td>temblor 0.3130</td>
<td>escala 0.3130</td>
<td></td>
</tr>
<tr>
<td>escala</td>
<td>0.1855 escala 0.3488</td>
<td>escala 0.3130</td>
<td>grados 0.3130</td>
<td></td>
</tr>
<tr>
<td>grados</td>
<td>0.1844 grados 0.3289</td>
<td>grados 0.3113</td>
<td>temblor 0.3113</td>
<td></td>
</tr>
<tr>
<td>sacudio</td>
<td>0.1725 sacudio 0.3255</td>
<td>sacudio 0.2943</td>
<td>seismo 0.2943</td>
<td></td>
</tr>
<tr>
<td>magnitud</td>
<td>0.1704 terremotos 0.3018</td>
<td>magnitud 0.2860</td>
<td>temblor 0.2860</td>
<td></td>
</tr>
<tr>
<td>terremotos</td>
<td>0.1407 magnitud 0.2935</td>
<td>terremotos 0.2467</td>
<td>temblor 0.2467</td>
<td></td>
</tr>
<tr>
<td>temblores</td>
<td>0.1205 sismico 0.2792</td>
<td>temblores 0.2151</td>
<td>temblores 0.2151</td>
<td></td>
</tr>
<tr>
<td>intensidad</td>
<td>0.1137 temblores 0.2721</td>
<td>intensidad 0.2041</td>
<td>sismico 0.2041</td>
<td></td>
</tr>
<tr>
<td>sismico</td>
<td>0.1080 seismos 0.2591</td>
<td>sismico 0.1949</td>
<td>seismos 0.1949</td>
<td></td>
</tr>
<tr>
<td>seismos</td>
<td>0.1022 sismica 0.2424</td>
<td>seismos 0.1854</td>
<td>sismica 0.1854</td>
<td></td>
</tr>
<tr>
<td>sismica</td>
<td>0.0929 daños 0.2130</td>
<td>sismica 0.1700</td>
<td>intensidad 0.1700</td>
<td></td>
</tr>
<tr>
<td>daños</td>
<td>0.0913 northridge 0.2126</td>
<td>daños 0.1673</td>
<td>northridge 0.1673</td>
<td></td>
</tr>
<tr>
<td>damnificados</td>
<td>0.0853 intensidad 0.2092</td>
<td>damnificados 0.1537</td>
<td>daños 0.1537</td>
<td></td>
</tr>
<tr>
<td>sacude</td>
<td>0.0737 tsunami 0.2056</td>
<td>sacude 0.1373</td>
<td>tsunami 0.1373</td>
<td></td>
</tr>
<tr>
<td>telurico</td>
<td>0.0729 maremoto 0.2026</td>
<td>telurico 0.1358</td>
<td>sismicos 0.1358</td>
<td></td>
</tr>
<tr>
<td>sinto</td>
<td>0.0706 sismicos 0.2006</td>
<td>sinto 0.1318</td>
<td>maremoto 0.1318</td>
<td></td>
</tr>
<tr>
<td>alas</td>
<td>0.0674 sacude 0.1879</td>
<td>alas 0.1263</td>
<td>sacude 0.1263</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Example for the Spanish entry *terremoto* in the expansion matrices.
the sum of the weight of the original query is used to reduce this value, but other criteria may be applied. The aim in this paper is not only to show whether these expansion techniques are valid to increase retrieval performance, but also to study the expansion technique itself. Therefore, as shown in Table 2, we have experimented with coefficients to compute the weight of expanded terms (n and \( \mod \) are respectively the number of terms and modulus of the original query). The coefficient denoted ‘Magic’ has no special meaning in information retrieval, we use it merely as another coefficient for test purposes.

At this point, it is necessary to comment on the normalization of document and query vectors. Normalization in document vectors prevents large documents from being considered more relevant than short ones. Normalization in query vectors is only used to obtain similarity values between 0 and 1 (we use standard scalar product for similarity function), but it does not affect the ranking. The coefficients in Table 2 have a different behaviour regarding normalization. ‘Average’ and ‘Unit’ coefficients have the same value with normalized and non-normalized query vectors. The other coefficients have different values.

Previous work [18] shows that normalization in query vectors has an impact when the original query is expanded with a few terms (about 0 to 50 terms). If more terms are added to the original query hardly any difference exists between a normalized query and a non-normalized one, except perhaps for the fact that the latter performs slightly better than the former. Thus, in our experiments, we have used non-normalized query vectors.

4 Experiments

Table 3 shows the collection used for our test. Only the TITLE and TEXT fields of the documents are used. For queries, the table indicates the average number of unique index terms for the ES-title field and for all fields. For our tests, we converted all words to lowercase, suppressed the stress signs, and included numbers as index terms. The number of stop words was 573. We used the well-known tfidf scheme and recommendations in [19], and the simple scalar product to calculate the similarity between query vector and document vectors. Only the document vectors were normalized.

The results were evaluated taking into account three measures (averaged over all queries): average non-interpolated precision, average R-Precision and average precision when 10 documents have been retrieved (precision at 10 docs). We include this last measurement because the user’s interface normally shows documents in groups of 10.

<table>
<thead>
<tr>
<th>Query-Freq</th>
<th>Average</th>
<th>Magic</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{\sum_{t_i \in q} q_i} )</td>
<td>( \frac{1}{n} )</td>
<td>( \frac{1}{\mod \cdot \sqrt{n}} )</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 3. Collection.

<table>
<thead>
<tr>
<th>Collection</th>
<th>Spanish (EFE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>215,738 (513 MB raw)</td>
</tr>
<tr>
<td>Queries</td>
<td>50 (C091 to C140)</td>
</tr>
<tr>
<td>Total index terms (TEXT and TITLE)</td>
<td>352,777</td>
</tr>
<tr>
<td>Averaged doc length (words)</td>
<td>333.68 (max. 2,210, min. 9)</td>
</tr>
<tr>
<td>Averaged doc length (unique index terms)</td>
<td>120.48</td>
</tr>
<tr>
<td>Averaged query length (unique index terms)</td>
<td>2.62 (ES-title) 20.48 (all)</td>
</tr>
</tbody>
</table>

Table 4. Results with 100 terms added to original query (ES-title field).

<table>
<thead>
<tr>
<th>Measurement</th>
<th>No expansion no stemming</th>
<th>Association Tanimoto</th>
<th>Cosine</th>
<th>Dice</th>
<th>Similarity Thesaurus</th>
<th>Inflectional Stemming</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg. precision</td>
<td>0.2618</td>
<td>0.2993</td>
<td>0.3281</td>
<td>0.3163</td>
<td>0.3342</td>
<td>0.2733</td>
</tr>
<tr>
<td>avg. R-Precision</td>
<td>0.2752</td>
<td>0.3095</td>
<td>0.3261</td>
<td>0.3121</td>
<td>0.3274</td>
<td>0.2866</td>
</tr>
<tr>
<td>avg. prec. at 10 docs</td>
<td>0.3320</td>
<td>0.3600</td>
<td>0.4160</td>
<td>0.3900</td>
<td>0.4100</td>
<td>0.3460</td>
</tr>
</tbody>
</table>

Table 5. Improvement using inflectional stemming (all query fields).

<table>
<thead>
<tr>
<th>Measurement</th>
<th>No stemming</th>
<th>Inflectional stemming</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg. precision</td>
<td>0.3908</td>
<td>0.4051</td>
<td>3.66%</td>
</tr>
<tr>
<td>avg. R-Precision</td>
<td>0.3844</td>
<td>0.4076</td>
<td>6.04%</td>
</tr>
<tr>
<td>avg. prec. at 10 docs</td>
<td>0.4840</td>
<td>0.4900</td>
<td>1.24%</td>
</tr>
</tbody>
</table>

We compute the association and the similarity thesauri from the document collection. The objective is to improve retrieval performance taking into account only the ES-title field of queries. For the sake of efficiency, only the terms in original queries were selected as entries in the thesauri. We do not use word or text windows, as other studies usually do. The whole document (TEXT and TITLE fields) is treated as a single word window. Neither do we apply stemming. For comparison purposes, we have calculated recall-precision values for only the ES-title query field without applying expansion or stemming.

A first set of tests was carried out to verify whether thesauri expansion techniques have better retrieval results than no expansion techniques. In this case, we use the weighting coefficient 'Average' for expanded terms. Table 4 shows the measurements for 100 terms added to the original query. This table also includes results for the experiment in inflectional stemming with no expansion. In all cases we have obtained positive improvement, but the improvement obtained with Cosine or Similarity thesauri is higher than the others. Very little improvement (about 4%) is obtained with inflectional stemming. Table 5 shows that a similar improvement is obtained using inflectional stemming for all query fields.
A second set of experiments was carried out to compare the efficiency for coefficients in Table 2. Figure 1 shows the way in which the number of additional terms affects the retrieval effectiveness. The improvement increases quickly with very few terms added to the original query. After 50 or 100 additional terms the improvement remains about constant. This figure shows improvement in average non-interpolated precision for association and similarity thesauri expansion. Similar results are obtained for average R-precision and average precision at 10 docs. In all cases, ‘Unit’ coefficient has the highest improvement, and ‘Average’ coefficient also shows a good performance. ‘Qiu-Frei’ and ‘Magic’ coefficients show some improvements, but their performance is the lowest. Figure 1 also shows that improvement with Cosine or Similarity thesauri is better (about 30%) than the others.

Figure 2 shows the evolution of Cosine and Similarity thesauri expansion. The results for average precision at 10 docs are very important since this is the number usually shown in the user’s interface for Web search engines or OPACs. For this measurement the improvement is from 35 to 40 percent.

5 Conclusions

In this paper we have explored association and similarity thesauri approaches to query or term expansion: queries are expanded to help improve retrieval performance. The results show that these techniques are valid. The expansion was
carried out using only the terms in the ES-title field of the queries. No stemming was applied to documents or queries in this expansion. For comparison with the CLEF2001 evaluation campaign, we have carried out an additional test on term expansion using inflectional stemming over documents and queries.

The best results are obtained with thesauri expansion. Improvement with stemming expansion is only about 4% for the ES-title field of the queries. For all query fields the improvement is similar. Stemming is a very expensive process, which includes some morphological and semantical information, hard to implement automatically. Simple measures of co-occurrence data as in association thesauri are cheap processes, and appear to give better results than stemming. The construction of the similarity thesaurus is also computationally expensive, but gives the best results.

Results indicate that query expansion using thesauri is a valid tool to improve retrieval performance. However, it should be noted that there is a disadvantage: as the number of query terms increases the response time of the information retrieval system also increases.

It is important to emphasize that thesauri are constructed automatically from the document collection. No user feedback information is used to select terms for expansion. When using relevance feedback information the number of additional terms is about 20. These terms are selected from retrieved documents.
marked as relevant. On the contrary, when the added terms are selected from an automatically constructed thesaurus, 50 to 100 terms is a reasonable figure.

For query expansion using association thesauri, the best results are obtained with the Cosine coefficient. With the similarity thesaurus a little improvement is obtained compared with to Cosine coefficient, but the construction of the former is computationally more expensive than that of the latter.

Our interest is centered on queries with very few terms. They have special importance in Web search engines, which typically use one to three terms per query. In our query set the average query length is 2.62 for E3-title. Expansion with association or similarity thesauri may be an instrument to improve retrieval effectiveness in Web search engines, but the information in Internet is very dynamic, and this implies that thesauri should be continually updated. Thus, we think that this type of expansion performs best when applied to more static document collections.

References