Portuguese-English Experiments using Latent Semantic Indexing

Viviane Moreira Orengo
Christian Huyck

School of Computing Science
Middlesex University
The Burroughs, London NW4 4BT
[v.orengo, c.huyck]@mdx.ac.uk

This paper reports the work of Middlesex University for the CLEF bilingual task. We have carried out experiments using Portuguese queries to retrieve documents in English. The approach used was Latent Semantic Indexing, which is an automatic method not requiring dictionaries or thesauri. We describe the methods used along with an analysis of the results obtained.

1 Introduction

Middlesex University is participating in CLEF for the first time. We have submitted runs for the bilingual task, using Portuguese queries to retrieve English documents. The approach adopted was Latent Semantic Indexing (LSI), which has achieved some promising results in previous experiments using other languages [5,10] and has the great advantage of not requiring expensive resources such as thesauri.

This paper is organised as follows: Section 2 presents some reasons for choosing Portuguese; Section 3 describes LSI and how it can be applied to CLIR; Section 4 reports our experiments and Section 5 analyses our results.

2 Why Portuguese?

Portuguese is the fifth biggest language in number of native speakers (see Figure 1). It is spoken on 4 continents: Europe (Portugal, Madeira, Azores), South America (Brazil), Africa (Angola, Mozambique, Cape Verde Islands, Guinea-Bissau) and Asia (Goa, Macau, East Timor). There are over 176 million native speakers and another 15 million people use it as a second language.

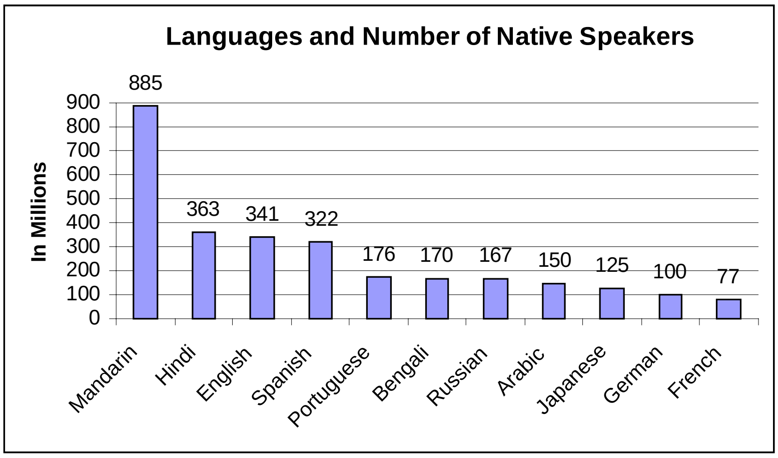


Figure 1– languages in number of native speakers [3]

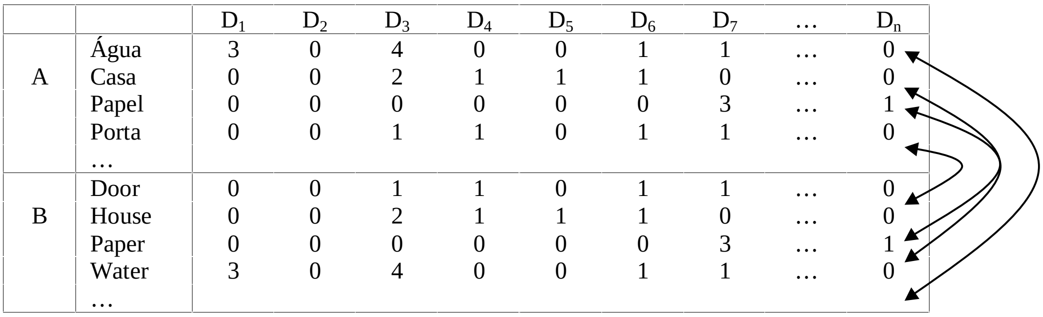
On the other hand, according to the Internet Software Consortium [4], less than 1% of all web hosts are in Portuguese. In addition, only a small percentage of this population is competent in English, the vast majority (including university students) are not able to present good queries (in English) to a search engine like Google, Altavista, Lycos, etc. As a consequence, sources of information for the 8 million Portuguese speakers accessing the Internet are extremely limited compared to the immense amount of information available to the English speaking population. Moreover, no CLIR research has been done using Portuguese. For all those reasons we decided to put Portuguese in the picture, by using it in our experiments.

3 CLIR using Latent Semantic Indexing

Latent Semantic Indexing [1] is a technique developed in 1990 by Dumais, Derweester, Landauer, Furnas and Harshman. The main goal is to retrieve on the basis of conceptual content rather than the actual terms used in the query. There are several ways of expressing a concept and the terms used in the query may not match the terms used in the documents.

LSI seeks to tackle synonymy and polysemy as they are the main problems with keyword matching. Synonymy is the fact that there are many ways to refer to the same object. And polysemy refers to the fact that many words have more than one distinct meaning. Attempts to solve the synonymy problem have been addressed by query expansion, which works by looking up a thesaurus and augmenting the query with related terms. The polysemy problem is considerably more difficult. Attempts to solve it include research done on word sense disambiguation. However, we are not aware of any adequate automatic method for dealing with it.

The main goal of using LSI for CLIR is to provide a method for matching text segments in one language with text segments of similar meaning in another language without needing to translate either, by creating a language-independent representation of the words. This means that words are given an abstract description that does not depend on the original language.



		D_1	D_2	D_3	D_4	D_5	D_6	D_7	…	D_n
A	Água	3	0	4	0	0	1	1	…	0
	Casa	0	0	2	1	1	1	0	…	0
	Papel	0	0	0	0	0	0	3	…	1
	Porta	0	0	1	1	0	1	1	…	0
	…									
B	Door	0	0	1	1	0	1	1	…	0
	House	0	0	2	1	1	1	0	…	0
	Paper	0	0	0	0	0	0	3	…	1
	Water	3	0	4	0	0	1	1	…	0
	…									

Figure 2 – Term by document matrix

LSI is initially applied to a matrix of terms by documents (see Figure 2). Therefore, the first step is to build such a matrix based on a set of dual-language documents[1]. The matrix contains the number of occurrences (or weights) of each term in each document. In a ideal situation the pattern of occurrence of a term in language A should be identical to the pattern of occurrence of its match in language B. The resulting matrix tends to be very sparse, since most terms do not occur in every document.

This matrix is then factorised by singular value decomposition[2] (SVD). SVD reduces the number of dimensions, throwing away the small sources of variability in term usage. The *k* most important dimensions are kept. Roughly speaking, these dimensions (or factors) may be thought as artificial concepts; they represent extracted common meaning components of many different words and documents. Each term or document is then

[1] Dual-language documents are composed by the document in the original language together with its translation in another language.

[2] Mathematics are presented in detail in [1].

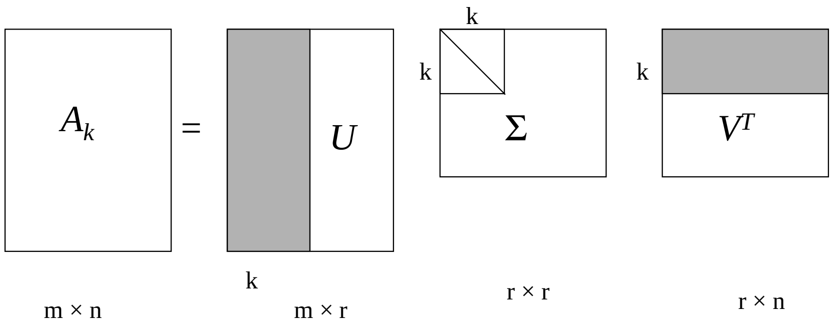
characterised by a vector of weights indicating its strength of association with each of these underlying concepts. Since the number of factors or dimensions is much smaller than the number of unique terms, words will not be independent. For example, if two terms are used in similar documents, they will have similar vectors in the reduced-dimension representation.

It is possible to reconstruct the original term by document matrix from its factor weights with reasonable accuracy. However, it t not advisable to reconstruct it with perfect accuracy, as the original matrix contains noise, which can be eliminated through dimension reduction. LSI implements the vector-space model, in which terms, documents and queries are represented as vectors in a *k*-dimensional semantic space. The SVD of a sparse matrix A is given by:

$$A = U\Sigma V^T$$

Where *U* and *V* are orthogonal matrices and Σ is the diagonal matrix of singular values. The columns of *U* and *V* contain the left and right singular vectors, respectively. The m × n matrix A_k, which is constructed from the *k*-largest single triplets of A, is the closest rank-k matrix to A.

Figure 3 – Singular value decomposition



A_k = Best rank-k approximation to A
U = Left singular vectors (term vectors)
Σ = Singular values
V = Right singular vectors (document vectors)

m = number of terms
n = number of documents
k = rank (number of dimensions)
r = number of dimensions of *A*

After deriving the semantic space with an initial sample of documents, new documents can be added (folded in), those are placed at the average of its corresponding terms. Queries are treated as pseudo-documents and placed at the weighted sum of its component term vectors. The similarity between query and documents is measured using the cosine between their vectors.

SVD causes synonyms to be represented by similar vectors (since they would have many co-occurrences), which allows relevant documents to be retrieved even if they do not share any terms with the query. This is what makes LSI suitable for CLIR, given that a term in one language will be treated as a synonym to its match in the other language. The main advantages of using LSI for CLIR are:

- There is no translation. All terms and documents are transformed to a language-independent representation.
- New languages can be added easily.
- There is no need for expensive resources such as dictionaries, thesauri or machine translation systems.

As with any method, LSI also has its drawbacks:

- SVD is computationally expensive and it may take several hours to be computed, for a reasonably sized collection.
- The only currently known way to determine the number of dimensions is through trial and error as there is no automatic way of establishing the optimal number. LSI research suggests a number which is large enough to incorporate important concepts and small enough to not include noise. In previous experiments, this number was typically between 100 and 300.
- Parallel corpora are not always available

4 Experiments

In order to use LSI for a bilingual experiment we needed an initial sample of parallel documents, i.e. the original documents and their translations, to derive the concept space. However, the collection we used, the Los Angeles Times, was in English only. Therefore we used Systran [8] to translate 20% (approximately 22000 documents) of the collection into Portuguese. Figure 4 shows a sample dual-language document used in the experiment. The translation is far from perfect, and many times the incorrect sense of a word was used, e.g. “branch” was translated to “filial” (shop branch), when the correct sense was “ramo” (tree branch). When the system did not have a translation for a term, it remained in the original language. Nevertheless, we did not perform any corrections or modifications on the resulting translations.

<DOCNO> LA012394-0072 </DOCNO> TIME TO CARE FOR ROSES Give attention to roses at this time. Prune them before spraying to reduce the total area that needs to be sprayed at this time. Remember to drench the **branches** and trunk until the material runs off the **branches**.	<DOCNO> LA012394-0072 </DOCNO> A HORA DE IMPORTAR-SE COM ROSAS dá a atenção às rosas neste tempo.Pode-os antes de pulverizar para reduzir a área total que necessita ser pulverizada neste tempo. Recorde drench as **filiais** e o tronco até o material funciona for a das **filiais**.

Figure 4 -Sample dual-language document

We used the Porter stemmer [7] to stem the English documents, and our own stemmer [6] to stem the Portuguese translations. Stop words were also removed. Next step was to run the SVD on the 22,000 dual-language documents. We used a binary version of LSI provided by Telcordia Technologies [9]. An important aspect is the choice of the number of dimensions that will compose the concept space. We chose 700 dimensions since this is the number which gave best performance, within reasonable indexing time, when using last year’s query topics. It was also the highest number that our system could support.

The entries in the term by document matrix were the local weight (frequency of a term in a document) multiplied by the global weight of the term (number of occurrences of a term across the entire collection). The weighting scheme used was “log-entropy” which is given by the formula below. A term whose appearance tends to be equally likely among the documents is given a low weight and a term whose appearance is concentrated in a few documents is given a higher weight. The elements of our matrix will be of the form: $L(\,i\,,j\,) * G(\,i\,)$

Local Weighting: $L(\,i\,,j\,) = \log(tf_{ij} + 1)$

Global Weighting: $G(\,i\,) = 1 - \sum_{j=1}^{N} \frac{p_{ij}\log(p_{ij})}{\log N}$ where $p_{ij} = \frac{tf_{ij}}{gf_i}$

where:

tf_{ij} = frequency of term *i* in document *j*
gf_{i} = total number of times term *i* occurs in the entire collection
N = number of documents in the collection

The next step was to “fold in” the remaining 91,000 English-only documents into that semantic space, which means that vector representations were calculated for those remaining documents. The resulting index had 70,000 unique terms, covering both languages. We did not index terms which were less than 3 characters long. We did not use phrases or multiword recognition, syntactic or semantic parsing, word sense disambiguation, heuristic association, spelling checking or correction, proper noun identification, a controlled vocabulary, a thesaurus, or any manual indexing.

All terms and documents from both languages are represented in the same concept space. Therefore a query in one language may retrieve documents in any languages. This situation benefits from cross-linguistic homonyms, i.e. words that have the same spelling and meaning in both languages; e.g. *“singular”* is represented by one vector only, accounting for both languages. On the other hand, it suffers with “false friends”, i.e. words that have the same spelling across languages but different meanings; e.g. “data” in Portuguese means “date” instead of “information”. The problem in this case is that false friends are wrongly represented by only one point in space, placed at the average of all meanings. The ideal scenario would be taking advantage from cross-linguistic homonyms and at the same time avoiding false friends. We are still looking for a way to do that automatically.

5 Analysis of Results

In and ideal situation the similarity between a term and its translation should be very high (close to 100%). In order to evaluate how accurately LSI represents the correspondence between terms, we calculated the similarities between some English words and their counterparts in Portuguese. Table 1 shows the results for this experiment. The scores are generally quite high. However, when cross-linguistic polysemy is present, the similarity decreases significantly (see second column). This happens because term co-occurrences decrease in the presence of polysemy and results in a term and its translation being placed further apart in the concept space.

baby	bebê	99.67%	Train	trem	12.30%
England	Inglaterra	99.24%	Train	treinar	84.97%
eat	comer	94.19%	Shade	sombra	31.88%
paper	papel	84.38%	Shadow	sombra	31.88%
book	livro	77.11%	Bank	banco	97.33%
Cyprus	Chipre	97.22%	Bench	banco	32.94%
car	carro	91.28%	Match	fósforo	96.72%
run	correr	51.03%	Match	jogo	36.28%
find	encontrar	80.06%	Game	jogo	91.30%

Table 1 – Term Similarity

We submitted three official runs:

- MDXman – keywords manually chosen from topic, description and narrative. The average number of words per query was 3.82
- MDXtpc – automatically chosen all terms from topic
- MDXtd – automatically chosen all terms from topic and description

Our results are summarised at Table 2. Our best score was MDXman, closely followed by MDXtd. The worst result was MDXtpc. We attribute this difference to the fact that shorter queries provide less (or no) context information to the retrieval system. Our performance is consistent with the scores obtained using LSI for a monolingual task [6]. We are less than happy with this results, but we think they are quite reasonable considering that we used machine translation to create a sample of parallel documents. A better quality translation would almost certainly improve the scores.

Run	Average Precision	R-Precision
MDXman	20.85%	23.10%
MDXtpc	15.51%	16.76%
MDXtd	20.88%	21.68%

Table 2 – Summary of results

Analysing the sets of documents retrieved for each query, we observed that most highly ranked but irrelevant documents were generally on the query topic, but did not specifically meet the requirements. As an example, for query 121 – “Successes of Ayrton Senna”, the documents retrieved were about him, but sometimes did not mention his sporting achievements. We also did not score well in queries 97, 98 and 131 for which there was only one relevant document. We believe that happened because we did not index words that occurred in one document only, and “Kaurismäkis”, for example, was one of them.

In comparison to the other groups, we have achieved the best score in four topics: 126,130, 137 (MDXman) and 106 (MDXtd). Also in four topics we achieved the worst result: 91, 98, 134 (MDXtpc) and 108 (MDXtd). For MDXman, 22 topics were at or above the median and 20 topics were below the median. LSI outperforms keyword based searching in the cases where the words used in the query do not match the terms used in the relevant documents. This was the case of topic 137 – “international beauty contests”, both relevant documents did not contain the keywords present in the topic. It had, however, related terms such as “miss world” and “miss universe”. Our superior performance in those cases confirms that LSI can efficiently tackle the synonymy problem, modelling relationships between related terms and placing them close together in the vector-space.

Future work will include further analysis of those results in order to establish methods for improvement. We are also interested in finding automatic ways to minimise the effects of polysemy.

References

1 DEERWESTER, S.; DUMAIS, S.; FURNAS, G.; LANDAUER T. and HARSHMAN,R. Indexing by Latent Semantic Analysis. **Journal of the American Society for Information Science**, 41(6):1-13, 1990.

2 DUMAIS, S. Latent Semantic Indexing (LSI) : TREC–3 Report

3 ETHNOLOGUE http://www.ethnologue.com

4 Internet Software Consortium http://www.isc.org/ds/WWW-200101/dist-bynum.html

5 LANDAUER, Thomas K.; LITMAN, Michael L. Fully Automatic Cross-Language Document Retrieval Using Latent Semantic Indexing. In: **Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research,** pages 31-38, Waterloo, Ontario, Oct. 1990.

6 ORENGO, V.M.; HUYCK, C.R. A Stemming algorithm for the Portuguese Language. In: **Proceedings of SPIRE'2001 Symposium on String Processing and Information Retrieval**, Laguna de San Raphael, Chile, November 2001.

7 PORTER, M.F. An Algorithm for Suffix Stripping. **Program,** 14(3), 130-137, July 1980.

8 SYSTRAN http://www.systransoft.com/

9 TELCORDIA Technologies - http://lsi.research.telcordia.com/

10 YANG Yiming et al. Translingual Information Retrieval: Learning from Bilingual Corpora. In: **15th International Joint Conference on Artificial Intelligence IJCAI'97,** Nagoya, Japan, August 23-29, 1997.