# Document Retrieval for Question Answering: A Quantitative Evaluation of Text Preprocessing

Gracinda Carvalho
L$^2$F/INESC-ID Lisboa
CITI – FCT/UNL
Universidade Aberta
Rua da Escola Politécnica, 147
1269-001 Lisboa, Portugal
+351 21 3916 465
gracindac@univ-ab.pt

David Martins de Matos
L$^2$F/INESC-ID Lisboa
Instituto Superior Técnico/UTL
Rua Alves Redol 9
1000-029 Lisboa, Portugal
+351 21 3100 305
david.matos@inesc-id.pt

Vitor Rocio
CITI – FCT/UNL
Universidade Aberta
Rua da Escola Politécnica, 147
1269-001 Lisboa, Portugal
+351 21 3916 465
vjr@univ-ab.pt

## ABSTRACT

Question Answering (QA) has been an area of interest for researchers, in part motivated by the international QA evaluation forums, namely the Text REtrieval Conference (TREC), and more recently, the Cross Language Evaluation Forum (CLEF) through QA@CLEF, that since 2004 includes the Portuguese language. In these forums, a collection of written documents is provided, as well as a set of questions, which are to be answered by the participating systems. Each system is evaluated by its capacity to answer the questions, as a whole, and there are relatively few results published that focus on the performance of its different components and their influence on the overall system performance. That is the case of the Information Retrieval (IR) component, which is broadly used in QA systems.

Our work concentrates on the different options of preprocessing Portuguese text before feeding it to the IR component, evaluating their impact on the IR performance in the specific context of QA, so that we can make a sustained choice of which options to choose. From this work we conclude the clear advantage of the basic preprocessing techniques: case folding and removal of punctuation marks. For the other techniques considered, stop word removal enhanced the performance of the IR system but that was not the case as far as Stemming and Lemmatization are concerned.

## Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

## General Terms

Experimentation, Measurement, Performance

## Keywords

Information Retrieval, Question Answering.

## 1. INTRODUCTION

This paper describes a set of experiments conducted in the context of the development of QA system for the Portuguese language with an IR component. The experiments focus on an analysis of the different preprocessing techniques that can be performed before the input of the information into the IR system. Several different options are compared and their impact on the performance of the IR is quantified having in mind the final QA goal of the system.

### 1.1 Information Retrieval in the Question Answering Context

Open domain QA systems seek to give a concise answer to a question, addressed in natural language that is not restricted to any specific field. The knowledge base of a QA system is usually a large collection of documents, also in natural language.

Considering the size of the information involved, many QA systems use IR modules in their architecture, because of their techniques to process and store the information in a way that enables a query over a large amount of data to be retrieved in a reasonably short time.

IR systems process and store large quantities of unstructured information, that does not need to obey a rigid format (usually text) in an efficient manner, so that it is able to quickly return the information that is relevant to a given request.

Information is input into the IR system through the document concept. A document is a block of text that will be returned as a whole, by the IR system, as a match to a query to the system. The returned documents of the IR, called hits, are usually ordered by a scoring function that tries to determine the relevance of the document to the query. The decision about the granularity of the documents is up to the user of the system. For instance, if one wants to feed the novel “War and Peace” to an IR system to find out details about the action, one can either consider each chapter a document, each paragraph a document, or each sentence a document, depending on the level of detail of the analysis to be made.

The main difference between an IR system and a QA system is that while the former returns to the user the documents that are more likely to be of interest to the query, the latter aims at producing a succinct answer extracted from the document(s), not the list of documents.

In a QA system, the IR component is generally used to filter out documents that have nothing to do with the question, retaining only the documents that are related, for further processing. It is therefore of fundamental importance that among the documents retrieved by the IR is the one (or several ones) that contains the answer.

## 1.2 Recent Research in QA

QA has been an area of interest for researchers, particularly over the last few years. This interest is in part motivated by the international QA evaluation forums, namely the Text REtrieval Conference (TREC), which has been conducting a track for QA since 1999 dedicated to the English language [7], and more recently the Cross Language Evaluation Forum (CLEF) through QA@CLEF, that since 2004 includes the Portuguese language [5]. In these forums, a collection of written documents is provided, as well as a set of questions, which are to be answered by the participating systems. Each answer is assessed manually, and, based on that assessment, an overall score is attributed to each participating system.

As a consequence, QA systems tend to be evaluated as a whole, and there are relatively few results published that focus on the performance of its different components, or alternative ways to perform a given task. That is the case in particular as far as IR component is concerned. However, there are some exceptions covering Passage Retrieval algorithms [4] [6] and the difference between Stemming and Query Expansion for English [2] .

Our work concentrates on the different options of preprocessing Portuguese text before feeding it to the IR system, evaluating their impact on the IR performance in the specific context of QA, so that we can make a sustained choice of which options to choose.

## 1.3 Paper Organization

This paper proceeds with a discussion of the preprocessing procedures to which text is generally subjected before the information retrieval techniques are used. Then a brief explanation on the IR performance metrics used is made, followed by the description of the experiments that were conducted, and the results obtained. We conclude with a final analysis and the direction we intend to follow in the development of our QA system.

# 2. TEXT PREPROCESSING

First of all, it is important to clarify what we call preprocessing. When using this concept we mean the thin layer that precedes the use of a component of a system that is prepared to treat different types of data for different purposes. The aim of that layer is to get the best results out of the module, regarding the specific type of data one wants to process.

If the architecture of a system is such that it uses a text classifier module prior to using an IR module, we do not consider the text classifier as preprocessing but another module of the system. The text classifier will probably need its own layer of text preprocessing.

The architecture for such system is shown in Figure 1.

It would depend on system design if the two preprocessing layers would be the same or slightly different, or even if the preprocessing of the text classifier (PP1) would make it unnecessary for the IR to have its own layer (PP2).

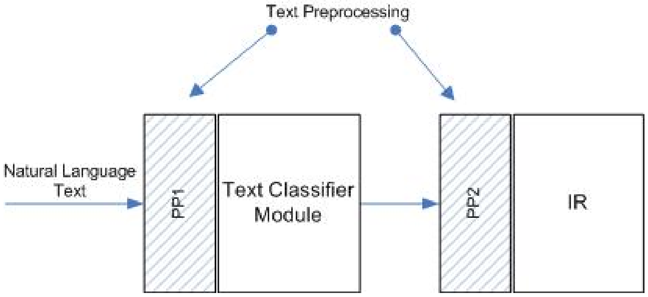


**Figure 1 - Architecture of a hypothetical system using text preprocessing**

In the case of the present work, the data type is natural language text and the component we consider is the IR.

## 2.1 Preprocessing and IR

An IR system is usually prepared to treat any kind of information, regardless of its nature. Its internal organization requires the information to be organized into units that will probably occur many times in the data. These units are commonly called tokens.

If the nature of the data is known in advance, preprocessing is a way to introduce meaning to the data in the IR, so that retrieval will be easier.

Preprocessing the input data can also have the goal of saving space and processing time, so the full data after preprocessing becomes a logical representation of that data with the most representative tokens of data chosen. However, the advance in the speed of computer components and the compression techniques, and despite the very large amounts of data available for processing, we concentrate mainly in efficacy rather than efficiency.

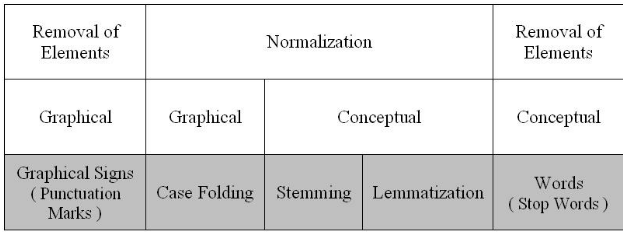
## 2.2 Text Preprocessing Techniques

There are mainly two approaches to text preprocessing: one is normalization and the other is the removal (or “cleaning”) of elements from the original text, that we believe contain “little” information and/or introduce “noise” in the retrieval.

The process of normalization can be interpreted in terms of defining equivalence classes between different representations, and the use of one of the representations for all the occurrences of that class.

Both of these techniques can be done at a graphical level or at a conceptual level. Generally, the graphical tasks are performed first because they are useful for the conceptual tasks.

The classification we propose for preprocessing methodologies is depicted in Figure 2. The different lines of action we just described are on the top part of the figure, while the bottom shaded area corresponds to the particular implementation of these approaches that we consider in our work, and that we will now describe in more detail.



| Removal of Elements | Normalization | | | Removal of Elements |
|---|---|---|---|---|
| Graphical | Graphical | Conceptual | | Conceptual |
| Graphical Signs ( Punctuation Marks ) | Case Folding | Stemming | Lemmatization | Words ( Stop Words ) |

**Figure 2 - Classification of preprocessing techniques**

Starting by the left-hand side, we find the graphical tasks. These are the first preprocessing tasks and they consist of preprocessing the full text, doing case folding (we chose lower case) and removing punctuation marks, as there is little information we can derive from them isolatedly. The result is a collection of words written in lower case. They will be used as tokens to IR because our knowledge base is natural language text, and words are the core concept of a language. We have thus performed tokenization.

We now have one representation for each word, even if it occurs in different formats in the text. For instance, the words “Lisboa” (“Lisbon”), “lisboa” and “Lisboa?” are all converted to “lisboa”.

The conceptual normalization is implemented by means of two different techniques: Stemming and Lemmatization. Both techniques aim at aggregating words that are related morphologically.

Stemming is a technique that tries to find the base (or stem) of the word by removing its affixes. It then replaces the word by its stem, thus combining the words that come from the same base. That is the case for instance of the words “amável”, “amigo” and “amor” (“kind”, “friend” and “love”) that all have for basis “am” which comes from Latin and means union and friendliness.

The stem of a word, as shown in the above example, does not need to be a word itself; generally it is just one syllable long.

There are algorithmic approaches to perform stemming, based on rules related to the affixes of the language. These rules do not always produce correct results. For instance in the case of the words “proteger” and “tecto” (“to protect” and “roof”) the common origin is “teg” (Latin for “cover”), but it is a case in which, through time, the letter “g“ turned into a “c”.

Lemmatization is a technique in which a valid word of the language (the lemma) is used as a representative for all the lexical variations that may apply. It is the headword that appears in a dictionary definition, as in the case in which “andar” (“to walk”) subsumes words as “andando” or “andei” (“walking” or “walked”).

These techniques are expected to produce good results for highly inflectional languages since they use the same representation for words with similar meaning. For instance, the sentence “A Maria vai ao Algarve” (“Mary goes to the Algarve”), might be converted to “a maria ir ao Algarve” since the word “vai” (third person, singular, future of the verb “to go”) is converted into its lemma “ir” (infinitive of the verb “to go”).

The removal of elements form the text at a conceptual level consists of removing a set of words, called stop words, that have little information per se (like conjunctions and articles). For instance, the sentence “O João comeu a sopa” (“John ate soup”), might be converted to “joão comeu sopa” if the stop list include the words “o” and “a”.

The definitions that we have made so far include some concepts that are ambiguous in the area of text processing. Although these techniques are part of almost all QA systems that use IR, questions like “What is the relevance of including a specific stop word in the stop list?” or “Should I use lemmatization?” are rarely addressed and even less quantified as far as its impact on IR performance for QA usage is concerned. With this work we hope to give a contribution to help clarify this sort of questions.

As a final note, we are aware that preprocessing takes out some information that was present in the full text, since we believe that, apart from involuntary mistakes like spelling mistakes or mistyped words, everything in the text has its function, that we are losing should we remove or change it. However, we are looking for a balance as far as IR is concerned, which means we may have to make some removals or changes to enhance IR performance. In any case, these changes are temporary, since we keep the identification of the texts from the retrieval phase, and use the full text again to do further processing after the retrieval phase . In this way the loss of information that the preprocessing might have introduced, will not affect the end goal of the QA system.

# 3. PERFORMANCE MEASURES

IR system performance is generally evaluated in terms of two standard measures, namely, precision and recall. Precision is the ratio of retrieved documents that are relevant to the query, in relation to the retrieved documents, whereas recall is the proportion of retrieved documents that are relevant to the query but in relation to all documents relevant to the query.

We will use specific measures defined for evaluating IR performance for QA usage: coverage and redundancy. If we consider the first *n* documents of the hits list, coverage indicates the probability of having a relevant document among those *n* documents, whether redundancy indicates the number of relevant documents in those *n* documents [4].

These measures are preferable in QA because in QA the IR system is used to find a number (*n*) of documents that may contain the answer. Those documents must be processed to check if they contain the answer and in the positive case, build the answer. This methodology fails if the IR system does not return any relevant document in the first *n* documents. So we are interested in knowing if a relevant document is among those *n* first hits, and that is what coverage represents, the probability that a relevant document is processed.

# 4. TEST DESIGN

## 4.1 Working Environment

In our experiments, we focus on domain-independent QA for Portuguese.

The text collection used is made available by Linguateca[1], and the texts belong to the knowledge base of the Question Answering task of the Cross-Language Evaluation Forum (QA@CLEF) for Portuguese. This collection consists of news articles from the

[1] http://www.linguateca.pt/

Portuguese daily newspaper Público, from Lisbon, for the years of 1994 and 1995. The edition of a given day is divided into news articles, to which a unique identification is assigned. In our case, a document for the IR system corresponds to a news article. The total number of documents is 106,821. The questions used are from the year 2004 evaluation campaign and they total 180. We use questions from this year because they are the only ones that have the information about the relevant documents, which allows automatic calculation of the coverage measure.

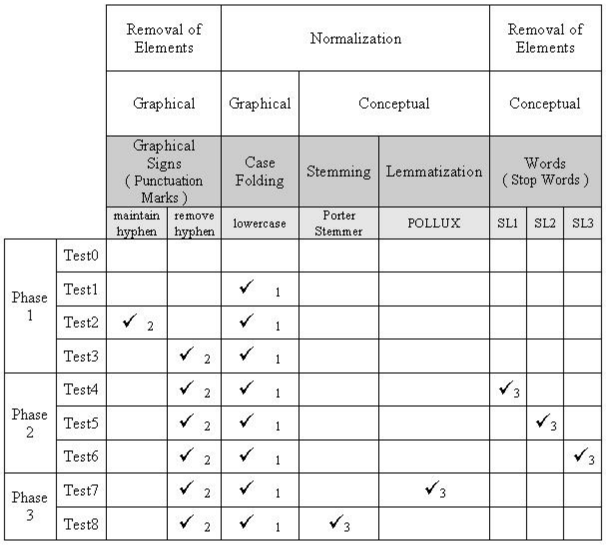
In our experiments we use CLucene[2], the C++ version of the open source IR API of Apache Lucene[3]. This IR system is commonly used in QA systems, with satisfactory performances [6].

## 4.2 Tests

We conducted a series of experiments to test the different text preprocessing techniques described in section 2.2.

In all the experiments, the same preprocessing used for the text collection is applied to the question, and the result is used to query the IR. We then search the hit list returned by the IR for the reference of one of the documents that contains the answer.

We conducted nine tests covering different preprocessing options. Figure 3 presents the techniques used in each test. It is an extension of Figure 2, where a line in light grey was added with specific implementations of the concepts of the dark grey line. A line was also added for each test, marking he technique(s) used and the number that appears on the bottom left-hand side of the cell indicates the order in which the different techniques were applied (1 - first  to 3 - third).



| | | Removal of Elements | | Normalization | | | Removal of Elements | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Graphical | | Graphical | Conceptual | | Conceptual | | |
| | | Graphical Signs ( Punctuation Marks ) | | Case Folding | Stemming | Lemmatization | Words ( Stop Words ) | | |
| | | maintain hyphen | remove hyphen | lowercase | Porter Stemmer | POLLUX | SL1 | SL2 | SL3 |
| Phase 1 | Test0 | | | | | | | | |
| | Test1 | | | ✓ 1 | | | | | |
| | Test2 | ✓ 2 | | ✓ 1 | | | | | |
| | Test3 | | ✓ 2 | ✓ 1 | | | | | |
| Phase 2 | Test4 | | ✓ 2 | ✓ 1 | | | ✓3 | | |
| | Test5 | | ✓ 2 | ✓ 1 | | | | ✓3 | |
| | Test6 | | ✓ 2 | ✓ 1 | | | | | ✓3 |
| Phase 3 | Test7 | | ✓ 2 | ✓ 1 | | ✓3 | | | |
| | Test8 | | ✓ 2 | ✓ 1 | ✓3 | | | | |

**Figure 3 – Summary of Tests**

The tests were divided in three phases that we describe in the following subsections.

### *4.2.1 Phase 1 – Basic Preprocessing*

In this phase, the techniques that work at graphical level are tested. Test0 corresponds to the full text, without any kind of processing, to establish a baseline to compare when introducing preprocessing. We proceed to Test1 in which only case folding was done (turning all letters to lower case). The tests related to the removal of punctuation marks were divided into two different situations: Test2, where the hyphen was the only punctuation mark that was kept, and Test3, where the hyphen was removed along with the rest of the punctuation marks.

We gave special attention to the treatment of the hyphen for two reasons:

1) The use of the hyphen in composite words like "co-orientador" ("co-advisor").
2) The use of the hyphen in Portuguese in the enclitic pronouns like in  "Ela disse-**me** …" ("She told **me** …") and in the mesoclitic pronouns like "Ela dir-**me**-ia …" ("She would tell **me** …").

We also treat unknown characters as word delimiters. For instance information like e-mail addresses or URLs are split up.

The parameterization of this phase that conducts to best results will be used in subsequent phases. As will be shown in the next section, it corresponds to that of Test3.

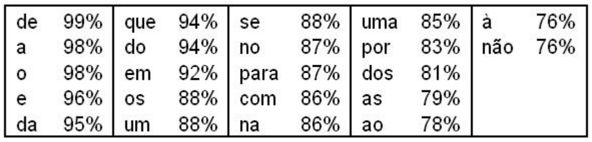
### *4.2.2 Phase 2 – Stop Lists*

We have several instances of stop lists for Portuguese. One, SL1, is composed by the 100 most frequent words in the corpus, and is published by Linguateca.

Another one, SL2, is published by the University of Neuchâtel[4] and is the Portuguese version of the procedure described in [3]. This list is composed of 356 words.

Stop list SL3 was built automatically and consists of the words that are in at least 75% of the documents of the collection. This list contains 22 words, and is shown if Figure 4, where the word is followed by the percentage of documents it which it occurs. The idea behind this list is that a word that belongs to practically all documents, does not contribute to make a distinction between them, so they belong to the class of "little" information.

Stop Lists SL1 ad SL2 are shown in Appendix A and B, respectively.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| de | 99% | que | 94% | se | 88% | uma | 85% | à | 76% |
| a | 98% | do | 94% | no | 87% | por | 83% | não | 76% |
| o | 98% | em | 92% | para | 87% | dos | 81% | | |
| e | 96% | os | 88% | com | 86% | as | 79% | | |
| da | 95% | um | 88% | na | 86% | ao | 78% | | |

**Figure 4 – Stop List SL3**

SL1 has many words specific to the corpus, i.e. commonly found in the newspaper context. Examples of this kind of words are:

- Lisboa – Lisbon
- nacional – national

[2] http://clucene.sourceforge.net

[3] http://lucene.apache.org

[4] http://members.unine.ch/jacques.savoy/clef/portugueseST.txt

- país – country
- Portugal
- presidente – president
- Público – ( the name of the newspaper ).

List SL2 contains almost all the word from SL1 (apart from some of the examples above), and SL3 is a subset of both lists SL1 and SL2.

### *4.2.3 Phase 3 – Stemming and Lemmatization*

As seen in section 2.2, the first technique consists of automatically shortening the word down to its stem, based on a set of rules, while the second replaces a word by its linguistic lemma (also a word), and therefore requires linguistic knowledge.

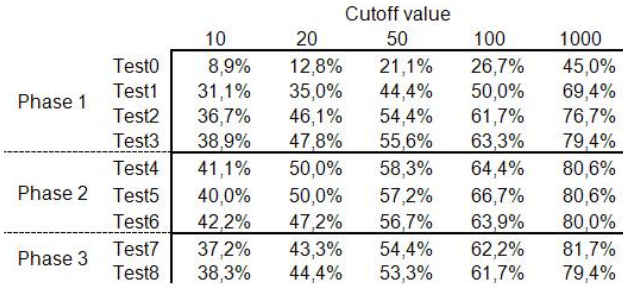
The lexical knowledge came from the POLLUX system (POrtuguese Lexical Largely Usable and eXtensible) [1]. This database has a table with 925,275 Portuguese lexical items, including inflected ones. Based on this information, a text file with the words and their lemma is build. This file is loaded into memory to be consulted in run-time. If a word does not belong to the list, it is maintained; otherwise it is replaced by its lemma.

The stemming algorithm follows Martin Porter's approach. The implementation of the Neuchâtel University was used[5]. This approach consists of successive steps of word reductions like removal of suffixes, normalization of gender and removal of accentuated characters.

## 5. RESULTS

The results for the coverage measure are presented in Figure 5. The different columns indicate several values for the cutoff of the hit list, so the search for documents that answered the question would be limited to the documents until that rank.

The values of 10, 20, 50, 100 and 1,000 were used, and, naturally coverage increases for higher cutoff values.



| | | Cutoff value | | | | |
|---|---|---|---|---|---|---|
| | | 10 | 20 | 50 | 100 | 1000 |
| Phase 1 | Test0 | 8,9% | 12,8% | 21,1% | 26,7% | 45,0% |
| | Test1 | 31,1% | 35,0% | 44,4% | 50,0% | 69,4% |
| | Test2 | 36,7% | 46,1% | 54,4% | 61,7% | 76,7% |
| | Test3 | 38,9% | 47,8% | 55,6% | 63,3% | 79,4% |
| Phase 2 | Test4 | 41,1% | 50,0% | 58,3% | 64,4% | 80,6% |
| | Test5 | 40,0% | 50,0% | 57,2% | 66,7% | 80,6% |
| | Test6 | 42,2% | 47,2% | 56,7% | 63,9% | 80,0% |
| Phase 3 | Test7 | 37,2% | 43,3% | 54,4% | 62,2% | 81,7% |
| | Test8 | 38,3% | 44,4% | 53,3% | 61,7% | 79,4% |

**Figure 5 - Coverage for the different preprocessing tests**

Reading the table we can see that converting all words to lowercase and removing punctuation marks, leads to a clear increment in IR performance. Test3 is consistently better than Test2 through all cutoff values, indicating that treating the hyphen as a special case (leaving it in the preprocessing) decreases IR performance. That can be explained by the fact that composed words are several times written as separate words and sometimes hyphenated; removing the hyphen would help aggregating words in both cases.

As far as stop word removal is concerned, the tests of Phase 2 show a slight increase in IR performance when compared to the results for Test3. The results suggest that list SL2 has better performance. However that is not the case for all cutoff values and the results are only marginally better, especially when compared to list SL1. List SL3 only gives better results for cut off 10.

The results of Phase 3 show a slight decrease in IR performance when using either Stemming or Lemmatization. Given the fact that the Portuguese language is highly inflected, it is a surprising result. Although the idea of aggregating words with similar meaning seem to lead to better results, it seems that if we aggregate too much, we can have unexpected results form the TF/IDF scoring mechanism with other documents scoring higher than the ones we are searching for. An indication of that is the fact that lemmatization has the highest coverage of all test for cutoff 1,000.

A factor that affects both Stemming and Lemmatization is that, since we are working simply at word level, proper nouns or named entities composed of multiple words or acronyms are not recognized as such. Since the stemming algorithm attempts in an automatic fashion to reduce all words to their stems, while Lemmatization leaves a word unaltered if it is not found in the lexicon, it was to be expected that Lemmatization would produce better results than Stemming, but that only happens in higher cutoff values (50 and above).

That can be explained by the fact that the Lemmatization process can also have shortcomings, and produce no results where Stemming does. As shortcomings to Lemmatization, we can indicate incomplete lexical information and the fact that words obtained by derivations that imply a change in morphological class are not considered: for example: "democracia" ("democracy"), the noun, will not be related to "democrático" ("democratic"), the adjective. Since we do not do any morpho-syntactic analysis that allows us to have a notion on the morphological class, whenever a word form has different lemmas we opt for leaving the original word, because we have no basis for deciding which lemma we should consider and we prefer to leave the original word instead of making a blind guess. One example of this situation is "fez" (noun – "the hat from the north of Africa and Turkey") whose lemma is the word itself, and "fez" (verb, 3rd person singular past – "did"), whose lemma is "fazer" ("to do"). This type of situation, however is not frequent in Portuguese.

The information regarding which documents contain an answer to the question is limited to only one document reference in 98% of the cases. We have manually processed a number of questions, and we have found numerous other documents that contain the correct answer, and they usually score higher than the ones indicated. We believe that this is the main reason why the coverage of our IR system is not better. We intend to improve this information (for instance by searching for the answers instead of the questions) so that the information is more comprehensive in terms of references of documents where answers can be found. It will also allow us to calculate the redundancy of the system, which will be useful to determine at what rank on the hit list the

[5] http://members.unine.ch/jacques.savoy/clef/portugueseStemmer.txt

cut off must be done. We also intend to increase the number of questions used.

## 6. CONCLUSIONS AND FUTURE WORK

We conducted nine tests covering different preprocessing options for the IR component of a QA system for the Portuguese language. The experiments focused on an analysis on the different preprocessing techniques that can be performed before the input of the information into the IR system.

The tests allow us to conclude the clear advantage of converting all words to lowercase and removing punctuation marks, and also that is better to treat the hyphen as any other punctuation mark. As far as stop word removal is concerned, the results improved, but there is not a clear better stop list. In Stemming and Lemmatization, for almost cutoff values, the IR performance slightly decreases, the only exception being the cutoff 1,000 for the Lemmatization, which gave better results.

We gave some explanations about the results, but for a better supporting of the decisions we need to improve the information about which documents contain the answer to the questions. We also intend to increase the number of questions.

Also we need to include in the study other normalization tasks, such as:

- named entities recognition and normalization,
- normalization by means of a thesaurus of verbs.

We plan also to study further options of the IR system, like its scoring capabilities because we have already done some preliminary tests where boosting parts of the question can lead to better results.

As far as future work in different components of a QA system is concerned, there are some areas that deserve our attention, for instance:

- Question analysis and classification,
- Answer extraction,
- Evaluation of answer adequacy.

## 7. REFERENCES

[1] Alves, M. A. Engenharia do Léxico Computacional: princípios, tecnologia e o caso das palavras compostas. *Mestrado emEngenharia Informática. Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa,* (20 Feb. 2002). www.liacc.up.pt/~maa/elc.html

[2] Bilotti, M.W., Katz, B. and Lin, J. What Works Better for Question Answering: Stemming or Morphological Query Expansion? *ACM SIGIR'04 Workshop Information Retrieval for QA,* (Jul. 2004).

[3] Fox, C. A stop list for general text. *ACM SIGIR Forum., Volume 24 ,Issue 1-2,* (1998), 19-21.

[4] Roberts, I., and Gaizauskas, R. Evaluating Passage Retrieval Approaches for Question Answering. *Lecture Notes in Computer Science, Book: Advances in Information Retrieval, Volume 2997,* (Mar. 2004), 72-84.

[5] Santos, D. and Rocha,P. CHAVE: topics and questions on the Portuguese participation in CLEF. *In C. Peters and F. Borri, editors, Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop, Bath, UK,* (15-17 September 2004) Pg. 639–648

[6] Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton, G. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. *ACM SIGIR'03., Toronto, Canada,* (Jul.-Aug. 2003).

[7] Voorhees, E. M. and Tice, D. M. Building a question answering test collection. *In SIGIR Forum (ACM Special Interest Group on Information Retrieval),* (2000) Pg. 200

## A. APPENDIX – SL1

a, à, agora, ainda, ano, anos, ao, aos, apenas, as, às, até, bem, cento, com, como, contos, contra, da, das, de, depois, dia, do, dois, dos, durante, e, é, em, entre, era, esta, está, estado, este, fazer, foi, foram, governo, grande, há, hoje, isso, já, lisboa, mais, mas, mesmo, mil, milhões, muito, na, nacional, não, nas, no, nos, num, numa, o, onde, ontem, os, ou, outros, país, para, parte, pela, pelo, pode, por, porque, portugal, presidente, público, quando, que, quem, são, se, segundo, sem, ser, seu, seus, só, sobre, sua, também, tem, ter, todos, três, tudo, um, uma, vai, vez

## B. APPENDIX – SL2

a, à, adeus, agora, aí, ainda, além, algo, algumas, alguns, ali, ano, anos, antes, ao, aos, apenas, apoio, após, aquela, aquelas, aquele, aqueles, aqui, aquilo, área, as, às, assim, até, atrás, através, baixo, bastante, bem, bom, breve, cá, cada, catorze, cedo, cento, certamente, certeza, cima, cinco, coisa, com, como, conselho, contra, custa, da, dá, dão, daquela, daquele, dar, das, de, debaixo, demais, dentro, depois, desde, dessa, desse, desta, deste, deve, deverá, dez, dezanove, dezasseis, dezassete, dezoito, dia, diante, diz, dizem, dizer, do, dois, dos, doze, duas, dúvida, e, é, ela, elas, ele, eles, em, embora, entre, era, és, essa, essas, esse, esses, esta, está, estar, estas, estás, estava, este, estes, esteve, estive, estivemos, estiveram, estiveste, estivestes, estou, eu, exemplo, faço, falta, favor, faz, fazeis, fazem, fazemos, fazer, fazes, fez, fim, final, foi, fomos, for, foram, forma, foste, fostes, fui, geral, grande, grandes, grupo, há, hoje, horas, isso, isto, já, lá, lado, local, logo, longe, lugar, maior, maioria, mais, mal, mas, máximo, me, meio, menor, menos, mês, meses, meu, meus, mil, minha, minhas, momento, muito, muitos, na, nada, não, naquela, naquele, nas, nem, nenhuma, nessa, nesse, nesta, neste, nível, no, noite, nome, nos, nós, nossa, nossas, nosso, nossos, nova, nove, novo, novos, num, numa, número, nunca, o, obra, obrigada, obrigado, oitava, oitavo, oito, onde, ontem, onze, os, ou, outra, outras, outro, outros, para, parece, parte, partir, pela, pelas, pelo, pelos, perto, pode, pôde, podem, poder, põe, põem, ponto, pontos, por, porque, porquê, posição, possível, possivelmente, posso, pouca, pouco, primeira, primeiro, próprio, próximo, puderam, qual, quando, quanto, quarta, quarto, quatro, que, quê, quem, quer, quero, questão, quinta, quinto, quinze, relação, sabe, são, se, segunda, segundo, sei, seis, sem, sempre, ser, seria, sete, sétima, sétimo, seu, seus, sexta, sexto, sim, sistema, sob, sobre, sois, somos, sou, sua, suas, tal, talvez, também, tanto, tão, tarde, te, tem, têm, temos, tendes, tenho, tens, ter, terceira, terceiro, teu, teus, teve, tive, tivemos, tiveram, tiveste, tivestes, toda, todas, todo, todos, trabalho, três, treze, tu, tua, tuas, tudo, um, uma, umas, uns, vai, vais, vão, vários, vem, vêm, vens, ver, vez, vezes, viagem, vindo, vinte, você, vocês, vos, vós, vossa, vossas, vosso, vossos, zero