Real-Time Open-Domain QA on the Portuguese Web

António Branco, Lino Rodrigues, João Silva, and Sara Silveira

University of Lisbon, Portugal
{antonio.branco,lino.rodrigues,jsilva,sara.silveira}@di.fc.ul.pt

Abstract. This paper presents a system for real-time, open-domain question answering on the Web of documents written in Portuguese, prepared to handle factual questions and available as a freely accessible online service. In order to deliver candidate answers to input questions phrased in Portuguese, this system resorts to a number of shallow processing tools and question answering techniques that are specifically geared to cope with the Portuguese language.

Key words: natural language processing, question answering, QA, real-time QA, open-domain QA, web-based QA, factoids, QA online service

1 Introduction

The Web allows the access to unparalleled amounts of publicly available online information. A credible estimate from the year 2005 places the indexable Web at 11.5 billion pages [1] and, since then, it has certainly kept growing.[1] Though expectedly smaller, the size of the Portuguese Web, consisting of documents written in Portuguese, should not be underestimated since Portuguese is likely to be one of the most used languages on the Internet—viz. the fourth most used according to an estimate published in the year 2002 [2].

Such a volume of data implies a considerable degree of redundancy as a given piece of information often happens to be phrased in many different ways and published in different sites. All this makes the Web a very attractive source of information when looking for answers to a large range of questions. Current Information Retrieval technology and popular search engines, however, do not deliver actual answers, even when simple factual questions are entered. Instead, for a list of input keywords, they return a list of documents and it is up to the user to seek a possible answer for his question within those documents.

Question Answering (QA) technology aims to go beyond the mere retrieval of relevant documents by returning possible answers to questions worded in plain natural language. This requires additional and more complex procedures which—this being the crucial point—are dependent on the specific idiom at stake.

In this paper we present XisQuê, a system for real-time, open-domain QA over the Portuguese Web. In order to deliver candidate answers to input questions phrased in Portuguese, this system resorts to a number of QA techniques

[1] `http://www.worldwidewebsize.com`, for instance, estimates 47 billion pages.

and shallow processing tools, e.g. for part-of-speech annotation, morphological analysis, lemmatization or named entity recognition, that are specifically designed to cope with the Portuguese language.

This system supports a freely accessible online QA service, which can be found at `http://xisque.di.fc.ul.pt`.

Paper structure. Section 2 presents the architecture adopted for the QA system, and describes its major modules. Section 3 addresses other QA systems for Portuguese. In Section 4, the performance of XisQuê is evaluated and compared with that of other systems. Finally, in Section 5 we discuss ongoing work and paths for future development, and present concluding remarks.

2 The QA System

XisQuê is a QA system that has been developed to comply with the following major design features:

Portuguese input: the admissible input are reasonably well formed, fully-fledged questions from Portuguese (e.g. *Quem assassinou John Kennedy?*).

Real-time: upon receiving a question, the service provides output to human users in real-time, i.e. without any QA-specific pre-processing indexing procedure of documents that are the source of information for getting answers.

Web-based: the possible answers to the input question are searched in and extracted from documents retrieved on the fly from the Web.

Extraction-based: the returned answers are excerpts extracted verbatim from the retrieved documents, and displayed with no further processing.

Portuguese Web: the possible answers are those that can be found and extracted from the Portuguese Web, that is the collection of documents written in Portuguese and publicly available online.

Open-domain: the input questions may address issues from any subject domain (e.g. Sports, History, etc.)

At the heart of the system heart lies the QA infrastructure described in [3], which is responsible for handling the basic non-linguistic functionality. This includes the managing of requests to the QA service, the submitting of the queries to search engines, and the downloading of selected documents.

On top of this infrastructure, the natural language driven modules were implemented by using existing, state-of-the-art shallow processing tools, developed by our group [4–6].

The underlying architecture for this set of modules follows what has become a quite standard configuration, that has been explored and perfected in similar QA systems for other natural languages: The first phase deals with question processing; This is followed by a stage of document retrieval; Finally, the answers are extracted from the retrieved documents. The modules responsible for these functionalities are addressed in the following Sections.

2.1 Question Processing

In our system, the stage of question processing is concerned with three tasks: (i) Extraction of the main verb and major supporting noun phrase (NP) of the question; (ii) Detection of the expected semantic type of the answer; and (iii) Extraction of relevant keywords. Tasks (i) and (ii) deliver elements that will play a key role in a subsequent phase of processing, viz. the answer extraction stage. Task (iii), in turn, helps to create the query to be used in the document retrieval step.

Phrase extraction. The main verb and the major supporting NP of the question (e.g. *John Kennedy* in *Quem assassinou John Kennedy?*) are extremely useful for helping to pinpoint the candidate answer in the answer extraction stage. For example, in many cases, the simple inversion between that NP and the main verb will help match declarative sentences that may contain an answer:

`Q:` `Quando` [`nasceu`]$_{\texttt{V}}$ [`Nelson Mandela`]$_{\texttt{NP}}$`?`
When was Nelson Mandela born?
`A:` [`Nelson Mandela`]$_{\texttt{NP}}$ [`nasceu`]$_{\texttt{V}}$ `em...`
Nelson Mandela was born in...

In order to detect its phrases, and in particular its supporting NP, the question is firstly annotated with part-of-speech (POS) by a state-of-the-art tagger for Portuguese [4]. The phrases are then extracted with the help of patterns defined by regular expressions.

With respect to the patterns for capturing the main verb, they permit to grab simple verbs (e.g. `assassinou`), verbs with proclisis (e.g. `se casaram`), and verbs with other auxiliary verbs (e.g. `foi assassinado`).

As for the NPs, we adapted the pattern coming out of the detailed study of their structure in [7, p. 69]. This pattern builds upon a number of syntactic slots reflecting precedence constraints in terms of word order:

1. predeterminers (e.g. `todos`);
2. determiners (e.g. `aquela`);
3. prenominal possessives (e.g. `suas`);
4. cardinals, ordinals, vague quantifiers (e.g. `três`, `terceira`, `muitos`);
5. prenominal adjective phrases (e.g. `grande`);
6. *head* (e.g. `bicicleta`);
7. adjectival arguments (e.g. `americana`);
8. adjective phrase adjuncts, prepositional phrase arguments, prepositional phrase adjuncts, adverbial phrase adjuncts, postnominal possessives, postnominal demonstratives (e.g. `intensa`, `com pedais`, `do Iraque`, `ali`, `teus`, `esse`).

These slots can be mapped into the POS categories assigned by the tagger.[2] Naturally, this kind of "shallow" chunking is limited by the expressive power of regular expressions, but tends to produce satisfactory results given the purpose and the application at stake, and has the upside of being quick and efficient.

[2] The correspondence is specific to the tagset used. Readers interested in these details are referred to [4, 7].

Answer type detection. Detecting the expected semantic type of the answer allows the system to narrow down the search space in the answer extraction stage. In that subsequent stage, we resort to a named entity recognizer (NER), which is able to identify expressions that behave like proper names for entities, and semantically classify them with one of a small set of pre-defined types, viz. person, organization, location, number, measure or time [6].

Accordingly, the answer that can be expected on the basis of the analysis of the input question is thus classified as being of one of these semantic types.

In many cases, the interrogative pronoun found in the question is a major clue and may be enough to anticipate the semantic type of the answer. For instance, the answer to a "Quando. . . ?" *(When. . . ?)* question will likely take the form of a time expression.[3]

In questions with the format "Que X. . . ?" *(Which X. . . ?)*, the interrogative pronoun does not provide enough information about the answer type. In these cases, the system uses a WordNet-like ontology which, though small, is yet large enough for the application at stake as it includes all the relevant sub-ontologies under the concepts that can be identified by the NER (person, organization, etc.). For a given "Que X. . . ?" question, this ontology is instrumental in finding the hypernym of "X" that is the semantic type of the expected answer. For instance, in the following question

Q: `Que pintor criou a Mona Lisa?`
Which painter created the Mona Lisa?

the hypernyms of `pintor` will be recursively retrieved until one that corresponds to an answer type (viz. person in this case) is found, as illustrated below.

`pintor` → · · · → `artista` → · · · → `pessoa`
painter *artist* *person*

Keyword extraction. For a given input question, the retrieval of documents where to look for its possible answers is done on the basis of a query. These queries are just a set of keywords. In order to obtain these keywords the system simply removes, from the input question, words belonging to the so-called closed, functional morphosyntactic categories. This implies that the keywords to be used for document retrieval are the words in the question pertaining to open classes.

For instance, for the question in the previous example, `Que pintor criou a Mona Lisa?`, the keywords would be `pintor`, `criou`, `Mona` and `Lisa`.

[3] This generalization has to be used with care. Exceptions to it can be found in cases like: Q: *When was Sting in Portugal?* A: *In Rock in Rio–Lisboa.* Though the question is introduced by *when*, the expected answer is not an expression denoting time but denoting an event. Exceptions with respect to the typical expected semantic type can also be found for other pronouns like *where* or *who.*

2.2 Document Retrieval

In order to perform the document retrieval task, the QA system acts as a client of several established high-quality search engines (viz. Ask, Google, MSN Live and Yahoo!). The system thus needs only to submit the list of keywords extracted from the question to each search engine, taking care also to select the option that, for that engine, limits the search to online pages that are written in Portuguese.

However, after the search engines return their list of result links, there are still many tasks that are performed to prepare the documents for the next stage:

- The links to the resulting documents are extracted from the HTML page with the results provided by the search engine. The top 10 results of each search engine are retrieved, avoiding repeated URLs, for a total of 40 links;
- The extracted URLs are used to download the documents in case there is an HTML version of them available. To ensure that the system provides a quick answer, the size of the download is capped at 512 Kb, and a timeout mechanism is used to safeguard against site accessibility problems;
- The character encoding of the downloaded documents is made uniform by converting them to Windows-1252 (roughly, a superset of ISO-8859-1);
- For documents that happen to be downloaded more than once (i.e. from more than one site), duplicates are discarded to avoid irrelevant processing;[4]
- Documents are stripped from HTML markup (formatting, tables, frames, menus, etc.), leaving only plain text.

2.3 Answer Extraction

The answer extraction stage is divided into two tasks: (i) candidate selection, where the best sentences, i.e. those most likely to contain an answer, are selected using an heuristic; and (ii) answer extraction itself, where candidate answers are extracted from the selected sentences.

Candidate selection. The documents that were gathered are split into sentences, which are then ranked. The top ranking sentence are selected as possibly containing answers to the input question. These sentence are the ones that proceed to the answer extraction task, thus reducing the number of sentences that must be processed in that (more computationally expensive) step.

The criterion to select the sentences that should be retained is based on the number of keywords that are in the query that came out of the input question and occur in those sentences. The threshold function is $q \geq \lfloor \sqrt{Q-1} \rfloor + 1$ where q is the number of keywords found in the sentence and Q is the number of keywords in the query, following the heuristic adopted in [8].

If they happen to pass this threshold, the retained sentences are ranked, receiving a higher rank if they contain a higher number of keywords and these keywords occur adjacent to each other.

[4] Currently, a simple heuristic is used to determine if two documents are duplicates: If the snippet of a document, that is returned by the search engine, is contained in the other document, one of them is discarded.

Answer extraction. The candidate sentences that remain in the final set are then annotated with a variety of linguistic information obtained with shallow processing tools, including POS, inflection features and lemmas.[5] The NER is then run over this annotated material. This is an hybrid tool, using regular expressions to detect entities with well-defined formats, mostly based on numbers (e.g. dates, measures, etc.) but relying on statistical methods for entities with a larger variation in format, mostly based on names (e.g. persons, organizations, etc). In addition, this tool also classifies entities according to their semantic type: PERSON, ORGANIZATION, LOCATION, NUMBER, MEASURE or TIME [6].

After the candidate sentences have been annotated, a set of patterns—defined by regular expressions—is applied to them to pinpoint and extract the desired answer. These patterns are specific to the semantic type of the question, hence the importance of detecting that type during the question processing stage.

Extraction patterns. As a way of illustration, here we will discuss only those patterns that are being used for "Quem. . . ?" *(Who. . . ?)* questions since space restrictions prevent us from presenting an exhaustive coverage. The description of the full set of patterns—16 in total—will be postponed for future publications.

Take, for instance, the following question, shown here after the question processing stage, with the main verb and supporting NP already identified:

$\texttt{Quem [escreveu]}_{\texttt{V}}\ \texttt{[Hamlet]}_{\texttt{NP}}\texttt{?}$
Who wrote Hamlet?

For such questions, the system applies the patterns described below to detect and extract the answer, viz. a named entity of type PERSON (marked by angle brackets in this example):

1. A pattern that matches the phrases in the question, but with the answer in the place of the interrogative pronoun, i.e. $\langle\cdots\rangle\ [\cdots]_{\texttt{V}}\ [\cdots]_{\texttt{NP}}$

   $\langle\texttt{William Shakespeare}\rangle\ \texttt{[escreveu]}_{\texttt{V}}\ \texttt{[Hamlet]}_{\texttt{NP}}$
   William Shakespeare wrote Hamlet

   Relative clauses are accounted for by extending this pattern to allow an optional `que` pronoun after the named entity

   $\langle\texttt{Shakespeare}\rangle\ \texttt{, que [escreveu]}_{\texttt{V}}\ \texttt{[Hamlet]}_{\texttt{NP}}$
   Shakespeare, who wrote Hamlet

2. A pattern that matches the passive form of the first case: $[\cdots]_{\texttt{NP}}\ [\cdots]_{\texttt{V}} \rightsquigarrow \langle\cdots\rangle$

   $\texttt{[Hamlet]}_{\texttt{NP}}\ \texttt{[foi escrito]}_{\texttt{V}} \rightsquigarrow \langle\texttt{Shakespeare}\rangle$
   Hamlet was written $\rightsquigarrow$ *Shakespeare*

   To match the passive voice, the pattern uses the lemma and inflection information that has been added to the sentence by the shallow processing

[5] For a detailed account of these tools, see [4, 5].

annotation tools. The $\rightsquigarrow$ is used here to indicate that the pattern allows anything after the verb up to the first named entity with the PERSON type, for instance: `pelo dramaturgo inglês` *(by the English playwright)*

3. A pattern that matches the common writing convention where the name of a work (book, painting, play, sculpture, etc.) is followed by the name of its author: $[\cdots]_{\mathtt{NP}}$ `, de` $\langle\cdots\rangle$

    $[\mathtt{Hamlet}]_{\mathtt{NP}}$ `, de` $\langle$`William Shakespeare`$\rangle$
    Hamlet, by William Shakespeare

2.4 Web Service

The XisQuê system supports a freely available Web service, which can be found at the following address: `http://xisque.di.fc.ul.pt`

The user introduces a question phrased in Portuguese and the system returns a list of 5 answers. Each answer (termed as short-answer from this point onwards) is accompanied by the sentence from where it was extracted (termed as long-answer), as a way of providing some extra context, together with a link to the original, full document. In case no short-answer is extracted from a top-5 scoring sentence, this sentence is still provided as a long-answer to the input question.

3 Related Work

There are some QA systems for Portuguese reported in the literature[6] though, for most of them, their performance is not comparable to that of XisQuê. On the one hand, these systems have no interface for searching the Web that could be used to collect comparable results, by running the test-set used here. On the other hand, running the CLEF test-set of questions [14] with XisQuê leads also to non comparable results since there is no guarantee that the answers in the CLEF corpus for these questions are available in the Web.

Moreover, while the preliminary application of indexing techniques to the closed corpus containing the answers is an option for QA systems evaluated over that corpus, as in CLEF, that is not an option for QA systems evaluated over the Web, as applying such techniques on such an amount of data is out of reach for any academic research team.

Esfinge [9] is the only of these QA systems that currently provides an interface for searching the Web.[7]

[6] Esfinge [9], L2F's [10], Priberam's [11], Raposa [12], Senso [13].

[7] Vd. `http://www.linguateca.pt/Esfinge`. We should note however that it is unclear to which extent this Web service is supported by the system reported in [9] since the system reported therein pre-processes and indexes the documents where the answer is to be searched, which does not happen in the Web-based version.

4 Evaluation and Comparison

Evaluating a QA system running over the Web raises non negligible issues as the Web's content is constantly changing. The results for a certain question in different runs may thus vary due to external factors, such as document or website availability, etc. [15]. Hence, there is no gold standard against which the system can be compared. However, it is possible and useful to obtain system performance indicators through sampling, by manual evaluation of a test set.

The test-set was built by randomly picking questions from Trivial Pursuit® cards, 15 for each of the interrogative pronouns that XisQuê handles (viz. `Quem`, `Quando`, `Onde` and `Que`), for a total of 60 questions,[8] which were submitted to each Web service. The following metrics are used:

Short answers is the proportion of questions for which the system provided at least a short answer—regardless its rank in the five answer list or its being correct. It serves as an indication of how good a system is at extracting short answers from candidate sentences.

Accuracy is the proportion of questions for which a correct answer was provided. In "all", a long-answer is counted in the lot of the correct ones in case it is correct and no short-answer (correct or not) was extracted from it. This is further divided into accuracy concerning only the top result and accuracy regardless of rank. It indicates the quality of the candidate sentences.

MRR stands for mean reciprocal rank. It is a measure commonly adopted in QA evaluation of how highly, on average, the first correct answer is ranked in the answer list [16].[9] For instance, if all questions have a correct answer and these all appear in position 1, the MRR scores 1; in case they would all appear in position 2, the MRR would score 0.5. It serves as an indication of the quality of the sentence ranking procedure.

XisQuê is compared with Esfinge and with Google, the latter serving as a baseline non-QA system on which QA systems should be able to improve since Google does not return short answers and thus only the result page is considered when checking for answers, i.e. we do not search within the returned documents and take the snippets provided in the results page as long-answers.

The results are summarized in Table 1.

4.1 Result analysis

XisQuê provides short-answers to 56.67% of the test set questions, which is slightly better than Esfinge. Google, naturally, does not provide short answers.

Since XisQuê returns a correct short-answer to 55.00% of the test set questions, a short-answer, when returned, is almost always correct. In this regard, Esfinge is similar, albeit with fewer short answers being returned.

[8] The full test-set may be found at `http://xisque.di.fc.ul.pt`.

[9] If no correct answer is given, a rank of 0 is used.

	XisQuê		Esfinge		Google	
short answers	56.67%		53.33%		n/a	
rank	1st	1–5	1st	1–5	1st	1–5
accuracy (short)	45.00%	55.00%	36.67%	51.67%	n/a	n/a
accuracy (all)	73.33%	98.33%	38.33%	53.33%	40.00%	91.67%
MRR (short)	0.4819		0.4500		n/a	
MRR (all)	0.7372		0.4667		0.5956	

Table 1. System comparison

For XisQuê, the long-answer acts as an effective fallback for those cases where a short answer was not provided, since a correct answer (short- or long-) is returned to 98.33% of the test set questions. In this respect, it is better than the Google baseline and it vastly improves on Esfinge's score since this system, when unable to provide a short-answer, does not return a long-answer. If we only look at the top ranked answer, these values are naturally lower, with XisQuê scoring 45.00% accuracy for short answers and 73.33% for short- or long- answers.

The overall MRR value obtained for XisQuê is 0.7372 when short- and long-answers are considered, and is 0.4819 when only short-answers are taken into account (i.e. when a rank of 0 is used for questions without any short-answer).

The difference between the "short" and "all" accuracies for XisQuê is a strong indication that the system still has room for improvement in the answer extraction stage: It is good at choosing candidate sentences (long-answers), but it is not always able to extract short-answers from them. However, those that it does extract tend to be correct.

4.2 A short note on QA timeliness

To assess the performance of a QA system, specially one with real-time as a design feature, one has to measure how fast the system is in the delivering of answers to end-users. In this regard, what matters to the end-user, and his perception of the usability of the service, is how long it takes for an answer to be returned since a question is asked to the system. From a system development point of view, however, it is instructive to also determine how much of that time is spent searching for and downloading documents, since those tasks are contingent on search engines that lie outside the QA core system proper.

On average, XisQuê takes 22 sec. to display the results, with 14 sec. (ca. 64% of total time) being spent in tasks performed "outside" the QA core system. This compares favorably with Esfinge, which takes 91 sec. on average to answer (from 21, for questions with no answers, to 342, for questions with answers).

5 Future Work and Conclusions

XisQuê has room for improvement and extension with new functionality. One of the first efforts will be towards broadening the system to support other question formats, e.g. requests for lists, definitions, question sets, etc.

Together with this extension in width, the system will also be extended in depth in an attempt to retrieve more and better answers. In this regard, there are several research avenues that are being pursued. Of particular interest are query expansion techniques, such as searching for synonyms, morphological expansion, detecting paraphrases (e.g. changing between active and passive voices), etc. Ultimately, we will move away from the shallow processing methods we currently use and apply deep linguistic processing [7].

In this paper we presented XisQuê, a real-time, open-domain, freely accessible online factoid QA service for the Portuguese Web with encouraging performance scores. Run over the 60 question test set, it returns a correct short answer for 55% of them, raising to 98% if long-answers are also taken into account. As for the salience of the correct answers, it got a 0.48 MRR score, raising to 0.75 if long-answers are also considered.

References

1. Gulli, A., Signorini, A.: The indexable web is more than 11.5 billion pages. In: Proceedings of the 14th International Conference on WWW, ACM (2005) 902–903
2. Aires, R., Santos, D.: Measuring the Web in Portuguese. In: Proceedings of the Euroweb Conference. (2002)
3. Rodrigues, L.: Infra-estrutura de um serviço online de resposta a perguntas com base na web portuguesa. Master's thesis, Universidade de Lisboa, Portugal (2007)
4. Silva, J.R.: Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, Universidade de Lisboa, Portugal (2007)
5. Nunes, F.: Verbal lemmatization and featurization of Portuguese with ambiguity resolution in context. Master's thesis, Universidade de Lisboa, Portugal (2007)
6. Ferreira, E., Balsa, J., Branco, A.: Combining rule-based and statistical methods for named entity recognition in Portuguese. In: Actas da 5ª Workshop em Tecnologias da Informação e da Linguagem Humana. (2007)
7. Costa, F.: Deep linguistic processing of Portuguese noun phrases. Master's thesis, Universidade de Lisboa, Portugal (2007)
8. Zheng, Z.: AnswerBus question answering system. In: Proceedings of the 2nd Human Language Technology (HLT). (2002) 399–404
9. Cabral, L., Costa, L., Santos, D.: Esfinge at CLEF 2007: First steps in a multiple question and multiple answer approach. [14]
10. Mendes, A., Coheur, L., Mamede, N., Romão, L., Loureiro, J., Ribeiro, R., Batista, F., Matos, D.: QA@L2F@QA@CLEF. [14]
11. Amaral, C., Cassan, A., Figueira, H., Martins, A., Mendes, A., Mendes, P., Pinto, C., Vidal, D.: Priberam's question answering system in QA@CLEF 2007. [14]
12. Sarmento, L., Oliveira, E.: Making RAPOSA (FOX) smarter. [14]
13. Saias, J., Quaresma, P.: The Senso question answering approach to Portuguese QA@CLEF-2007. [14]
14. Nardi, A., Peters, C., eds.: Working Notes for the CLEF 2007 Workshop. (2007)
15. Breck, E., Burger, J., Ferro, L., Hirschman, L., House, D., Light, M., Mani, I.: How to evaluate your question answering system every day... and still get real work done. In: Proceedings of the 2ndLREC. (2000) 1495–1500
16. Voorhees, E.: The TREC8 question answering track report. In: Proceedings of the 8thText REtrieval Conference (TREC). (1999)