Answering List Questions using Web as a corpus

Patrícia Nunes Gonçalves, António Branco
University of Lisbon
Edifício C6, Departamento de Informática
Faculdade de Ciências, Universidade de Lisboa
Campo Grande, 1749-016 Lisboa, Portugal
{patricia.nunes, antonio.branco}@di.fc.ul.pt

Abstract

This paper supports the demo of LX-ListQuestion, a Web Question Answering System that exploit the redundancy of information available in the Web to answer List Questions in the form of Word Cloud.

1 Introduction

The combination of web growth and improvements in Information Technology has reignited the interest in Question Answering (QA) systems. QA is a type of information retrieval combined with natural language processing techniques that aims at finding exact answers to natural language questions. In a search engine, the user inserts a few keywords and gets as a result links and snippets. The task of finding the desired answer among the results that were returned then falls on the user. From the point of view of QA, in turn, the users use a question in natural language and the system searches within the documents for the answers.

Questions have various levels of complexity. When thinking about questions immediately come to mind factual questions (eg: When did Nelson Mandela die?), however, the QA area has expanded beyond factual questions towards more complex questions. One of the most common types of complex factual is list questions. The list questions are questions for which there is a list of answers, e.g., *In which countries Portuguese is an official language?* List answers: *Angola, Brazil, Cape Verde, Guine Bissau, Mozambique, Portugal, Macau, Sao Tome and Principe and East Timor.*

When the information that is needed is non-trivial and it is found spread over several texts, a lot of human effort is required to gather the various separate pieces of data into the desired result, which is not an easy task. Ideally, they would prefer to quickly get a precise answer and go on to make use of it instead of spending time searching and compiling the answer from pieces spread over several documents.

Our purpose is to provide better QA solutions to users, who desire direct answers to their queries, using approaches that deal with the complex problem of extracting answers found spread over several documents and use them to compile a list of answers that are the most accurate possible. The LX-ListQuestion development is guided by the circumstance that answers may appear redundantly in many places and in many forms. Our approach to address the problem of answering list questions is to explore this redundancy. To build on this redundancy, we use techniques that will be explained in section 4.

2 Related Work

List QA is an emerging topic and few approaches have been developed. The most common approach is to take a QA system for factoid questions and extend it to answer List questions. Some pioneering systems using this approach are (Gaizauskas et al., 2005) and (Wu and Strzalkowski, 2006), which show a low performance. Other systems explore NLP tools and linguistic resources (Hickl et al., 2006) (Yang et al., 2003). This approach seems to have achieved competitive results. However the time required for processing is very high and the performance of these systems depend on the performance of the supporting NLP tools.

Other approaches resort to statistical and machine learning approaches. The system developed in (Whittaker et al., 2006) is based on a statistical model. The system developed by (Yang and Chua, 2004) employ classification techniques to find complete and distinct answers. The system proposed by (Razmara and Kosseim, 2008) answers List questions using a clustering method to group candidate answers that co-occur more often in the collection.

Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 81–84,
Gothenburg, Sweden, April 26-30 2014.

Systems that take advantage from semantic content to answer List questions (Cardoso et al., 2009), (Hartrumpf and Leveling, 2010), (Dornescu, 2009) achieved good results although all information should be stored in the database. This approach seems suitable to QA system that focus on a specific domain where the information source can be limited and more easily stored.

3 List Question

In the context of QA research, list questions may appear in three basic forms: (1) a question starting with an interrogative pronoun, (2) a request using an imperative verb and (3) other forms: without interrogative pronoun or imperative verb (usually a complex noun phrase). Table 1 shows some examples.

Type of List Question	Example
Interrogative Pronoun	What European Union countries have national parks in the Alps?
Imperative Form	Name rare diseases with dedicated research centers in Europe.
Other	Chefs born in Austria who have received a Michelin Star.

Table 1: Examples of List questions

Another important topic in QA is to identify the so called question target. Our study shows that target can be expressed by a named entity, a common noun or be multiple targets. Most List Questions have named entities as target question, e.g. *Which cities in Germany have more than one university?*. Common noun is not so frequent as target question but it can show up, e.g. *Typical food of the Cape Verde cuisine.* Multiples target can be of named entities or common nouns, e.g. *Newspapers, magazines and other periodicals of Macau.*

The list answers (for a list question) may appear in many places and in many forms. They can be in the same document; when the answer is already a list, e.g., *list of cities in Portugal: Lisbon, Coimbra, Porto e Faro*; or the answers can be spread over multiple documents; e.g., (document A): *Lisbon is the capital of Portugal.* (document B) *Porto is a very important city in Portugal.* In the latter case, a QA system able to answer List questions has to deal with this diversity and find all answers of several texts and compose the final list of answers.

Our system focuses on answering List questions where the answers are extracted from several documents from the Web.

4 LX-ListQuestion Architecture

LX-ListQuestion System seeks to answer List questions through the use of the techniques of Question Answering running over the Web of Portuguese pages, while ensuring that the Final Answer List is as correct and complete as possible. This system has three main modules: Question Processing, Passage Retrieval and Answer Extraction. Figure 1 shows its architecture.

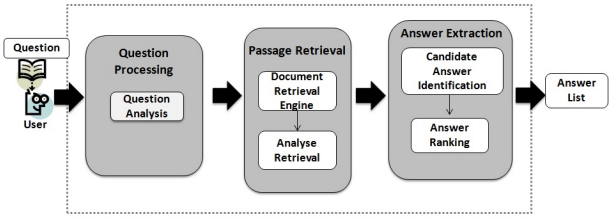

Figure 1: Question Answering System Architecture

The Question Processing module is responsible for converting a natural language question into a form that a computer is capable of handling and extracting the information that will be passed and used by the subsequent modules. Question Analysis task is responsible to clean the questions, i.e. removing question marks, interrogative pronoun and imperative verbs. Besides identifing each meaningful element of the question, the system annotate each word with their part-of-speech tag. The following information are set apart: main verb, question target, named entity.

The Passage Retrieval module is responsible for searching web pages using the keywords into the question and save them into local files for post-processing. This version of the system is working with 10 downloaded files.This module is also responsible for cleaning the HTML files and saving into local files only the content information. We use a relevance score based on the average of the number of words in the question to preserves the original idea of the question. After the content is saved into a file, the system will select the relevant sentences based on matching and counting the keywords in the sentences.

The Answer Extraction Module aims at identifying and extracting relevant answers and presents them in the form the a list. This module has two main tasks: Candidate Answer Identification and Building the List Answer. Candidate Answer Identification task extracts all words tagged with

the proper name tag (in this version of the system version we are assuming that all answers are proper names).

The process of Building the List Answers based on frequency and word occur rules. The process based on frequency uses two lists: Premium List and Work List. The Premium List is composed by candidates extracted from sentences previously classified as highly relevant and will serve to guide the rest of the processing. If we were to consider only these elements, the list of answers would probably contain correct items. However, the list may be incomplete and lack elements. Then, continuing in the same vein of our strategy, the Work List is build with candidates extracted from sentences classified as medium and low. This list will be used to confirm and expand the elements in the list. The word occur rules take advantage of the title of web page, of sentences that perfectly matches with the question and of candidate verb that matches with the question verb.

4.1 Results

For the experiments we used a set of 10 questions[1] that require List Answers:
Q1: Instrumentos musicais de origem africana comuns no Brasil. *African musical instruments common in Brazil.*
Q2: Parques do Rio de Janeiro que têm cachoeiras. *Parks of Rio de Janeiro that have waterfalls.*
Q3: Igrejas em Macau. *Churches in Macau.*
Q4: Cidades que fizeram parte do domínio português na Índia. *Cities in India that were under Portuguese rule.*
Q5: Parques nacionais de Moçambique. *National parks in Mozambique*
Q6: Ilhas de Moçambique. *Islands of Mozambique*
Q7: Movimentos culturais surgidos no nordeste do Brasil. *Cultural movements that emerged n the northeast of Brazil.*
Q8: Dioceses católicas de Moçambique. *Catholic dioceses in Mozambique.*
Q9: Candidatos a alguma das eleições presidenciais na Guiné-Bissau. *Candidates for any of the presidential elections in Guinea-Bissau.*
Q10: Capitais das províncias de Angola. *Capitals of the provinces in Angola.*

[1]These questions were based on Pagico: www.linguateca.pt/Pagico

Table 2 shows the metric evaluation of the LX-ListQuestion. The metrics used are: recall, precision and F-measure. These metrics take into consideration two lists: a reference list (correct answers expected) and the system list (answers returned by the QA system). Precision: C is the number of common elements between reference and system lists and S is the number of elements given by the system.
$Precision = \frac{C}{S}$
Recall: C is the number of common elements between reference and system lists and L is the number of elements in reference list.
$Recall = \frac{C}{L}$
F-measure: its the combination between Recall and Precision.
$F - measure = \frac{2*Recall*Precision}{Recall+Precision}$

Table 2: Metric Evaluation.

Question	Precision	Recall	F-Measure
Q1	0.15	0.36	0.21
Q2	0.08	0.50	0.14
Q3	0.13	0.35	0.18
Q4	0.14	0.23	0.17
Q5	0.10	0.75	0.18
Q6	0.13	0.42	0.20
Q7	0.05	0.20	0.08
Q8	0.15	0.71	0.24
Q9	0.05	0.25	0.09
Q10	0.38	0.42	0.40
AVERAGE	0.14	0.38	0.20

We observe from Table 2 that the system answered all questions. It achieved better recall for the questions Q5 and Q8. The Question Q10 obtained better precision and also f-measure. Overall, the system scores 0.38 of recall, which is a very competitive result for the current state-of-art. Exploring the redundancy of information seems to be a good approach to this task, but it alone cannot handle all problems. The word occur rules implemented was important step to enrich the system.

4.2 User interface

LX-ListQuestion is available on the web:
http://nlxserv.di.fc.ul.pt/lxlistquestion/
In our tests, the response time ranged between 16 and 28 seconds of processing from submitting the question and getting the list of answers. We chose to use word cloud instead of a traditional list as presentation of results because the final list an-

swers evidence not only the correct answers but of the possibly relevant words related to the question. The confidence that the system has a given answer is based on the frequency of words found in the texts collected from the web. The most frequent words are represented in the word cloud using a greater font size. The word cloud also helps the user to understand the context in which the answers may be embedded. Figure 2 shows LX-ListQuestion online GUI.

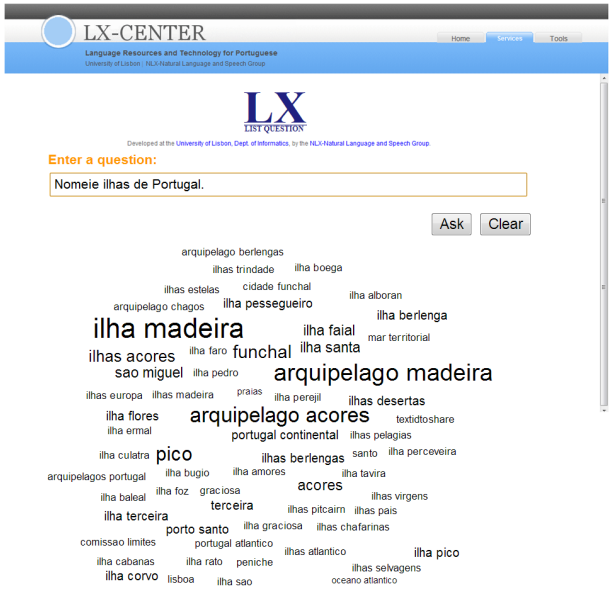

Figure 2: LX-ListQuestion online GUI

5 Concluding remarks

LX-ListQuestion is a fully-fledged Web based QA system that generates answers to list questions and presents them in a word cloud. The system exploits the redundancy of information available in the Web and combine with word occur rules to improve QA accuracy. This version handles Portuguese Language. The next version will be extended to provide answers to other languages as well. This work has being developed as the subject of a progressing doctoral thesis and new improvements will be implemented.

References

Nuno Cardoso, David Batista, Francisco J. López-Pellicer, and Mário J. Silva. 2009. Where in the wikipedia is that answer? the xldb at the gikiclef 2009 task. In Carol Peters, Giorgio Maria Di Nunzio, Mikko Kurimo, Djamel Mostefa, Anselmo Peñas, and Giovanna Roda, editors, *CLEF*, volume 6241 of *Lecture Notes in Computer Science*, pages 305–309. Springer.

Iustin Dornescu. 2009. Semantic qa for encyclopaedic questions: Equal in gikiclef. In Carol Peters, Giorgio Maria Di Nunzio, Mikko Kurimo, Djamel Mostefa, Anselmo Peñas, and Giovanna Roda, editors, *CLEF*, volume 6241 of *Lecture Notes in Computer Science*, pages 326–333. Springer.

Robert J. Gaizauskas, Mark A. Greenwood, Henk Harkema, Mark Hepple, Horacio Saggion, and Atheesh Sanka. 2005. The university of sheffield s trec 2005 q&a experiments. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, November 15-18, 2005*, volume Special Publication 500-266. National Institute of Standards and Technology (NIST).

Sven Hartrumpf and Johannes Leveling. 2010. GIRSA-WP at GikiCLEF: Integration of structured information and decomposition of questions. In *10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30-October 2, Revised Selected Papers*, Lecture Notes in Computer Science (LNCS). Springer. (to appear).

Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Ying Shi, and Bryan Rink. 2006. Question answering with lcc's chaucer at trec 2006. In *TREC*.

Majid Razmara and Leila Kosseim. 2008. Answering list questions using co-occurrence and clustering. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Edward W. D. Whittaker, Josef R. Novak, Pierre Chatain, and Sadaoki Furui. 2006. Trec 2006 question answering experiments at tokyo institute of technology. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, November 14-17, 2006*, volume Special Publication 500-272. National Institute of Standards and Technology (NIST).

Min Wu and Tomek Strzalkowski. 2006. Utilizing co-occurrence of answers in question answering. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics.

Hui Yang and Tat-Seng Chua. 2004. Web-based list question answering. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1277, Morristown, NJ, USA. Association for Computational Linguistics.

Hui Yang, Hang Cui, Mstislav Maslennikov, Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2003. Qualifier in trec-12 qa main task. In *TREC*, pages 480–488.