Language Modeling for Document Selection in Question Answering

Nicolas Foucault, Gilles Adda, Sophie Rosset
LIMSI - CNRS
firstname.lastname@limsi.fr

Abstract

Usually, in the Question Answering domain, for a question in natural language, precise answers to the question are extracted from documents according only to the context of the question. In this work, we complemented this approach by adding a filtering process on top of the document retrieval. This way, the system re-evaluates the documents it has originally selected during the information retrieval step before the answer extraction and scoring. Such re-evaluation aims at filtering out documents considered unusable for the search. Based on statistical language modeling, the filtering process firstly determines the intrinsic relevancy of a document and then decides whether this document is *a priori* relevant for finding answers. Evaluation on factoid questions and a collection of 500k web documents has shown our approach properly supports the Question Answering task.

1 Introduction

Question-Answering (QA) systems can be seen as an extension of the Information Retrieval (IR) engines. In IR systems a user is able to search for information using a set of keywords. The search result is a set of documents or links to documents the user needs to peruse to find the precise information he asked for. In contrast, the QA task consists of providing short, relevant answers to natural language questions which can be textual or spoken. For instance, looking for the main actors playing in the "Titanic" movie directed by James Cameron, a possible question to a QA system would be: *Who did play the main roles in the Titanic movie directed by James Cameron?*. In return, the system might reply: *Leonardo DiCaprio and Kate Winslet*.

Question-Answering systems usually follow a standard strategy. They start by preprocessing the documents before their indexation.

The indexation for subsequent retrieval is done by a classical (e.g. Lucene[1]) or specific search engine (Rosset et al., 2008) developped on purpose to best fit the system needs.

Following these steps which predates any retrieval, the work turns towards the questions. The question analysis aims at providing information from the question that has to be found in the documents. The second part of the analysis aims to predict what type of answers the question expects (Pardino et al., 2008), usually a named entity category (such as person, location, etc.) and also to predict what the question class is, so as to constrain the system to search for specific answer types.

The results of these analysis are given to the search engine which retrieves whole documents or snippets, based on the indexation, in order the system finally rank candidates answers it extracted from them.

In this paper, we describe a method which first determines the intrinsic relevancy of a document using a language model and then decides whether this document is relevant for searching answers to any question. In the following section we present related work. Section 3 presents the proposed method. Section 4 shows experiments conducted for its evaluation. Finally, in section 5 we conclude and gives future perspectives about our work.

2 Related Work

In QA the document selection is done given a specific question. As far as we know, no work addressed the problem of selecting documents independently to the question, using only a document

[1] http://www.lucene.apache.org

Proceedings of Recent Advances in Natural Language Processing, pages 716–720,
Hissar, Bulgaria, 12-14 September 2011.

quality evaluation. Such a method involves assessing whether a document is intrinsically relevant or not, and is totally compliant with previous and further analysis in the standard QA strategy.

Statistical language modeling (SLM) seems suitable for such a task. SLM (Jelinek et al., 1990; Rosenfeld, 2000) provides an easy way to cope with the complexity of natural language by expressing various language phenomena in terms of simple parameters in a statistical model. If SLMs have not been used extensively in pure QA, although they have shown promizing results e.g. to evaluate the intrinsic relevancy of documents estimated for ranking passages (Ganesh and Varma, 2009), they are classically used to help solving tasks closely related to the QA one, especially when topic modeling is worth e.g. entity linking and guided summarization [2] (Varma et al., 2010).

3 Document evaluation method

3.1 Overview

The document evaluation method applied to a given *d* document is 2-twofold: firstly, *d* is scored using a language model (LM) in order to estimate its intrinsic relevancy. Then, a Gaussian Mixture Model (GMM) predicts whether *d* is relevant, given a model of *a priori* relevant documents (which are the documents included in the development set, DEV) and the LM. In other words, *d* is considered as relevant only if *d* is close enough both to the documents used to build the LM and to the DEV documents.

The LM is built on a very large collection of journalistic articles to define a model with a broad scope. Preliminary experiments have shown that the *perplexity* (PPX) and the *out of vocabulary words* (OOV) ratio were the most suitable parameters to characterize the document relevancy. PPX is defined as:

$$PPX(d) = P_{LM}(d)^{-\frac{1}{|d|}} \quad (1)$$

where $P_{LM}(d)$ is the document estimated probability, given the LM, and $|d|$ is the number of word in d. PPX might be seen as a distance between d and the documents known by the LM. OOV ratio is defined as:

$$OOV(d) = \frac{|\overline{d \cap LM}|}{|d|} \quad (2)$$

[2] for details about such tasks see the KBP/GS tracks at `http://www.nist.gov.tac`

where $d \cap LM$ are the words in d which belong to the LM vocabulary, and conversely $\overline{d \cap LM}$ are the words in d unknown by the LM. OOV is a ratio, corresponding to the number of words unknown by the LM divided by the total number of words in *d*.

3.2 Methods

The first step is to build a 3-gram LM based on a 500k words dictionary obtained from a large corpus of French newspapers articles. Then, OOV and PPX scores are calculated to each DEV documents using the LM and we estimate the distribution (assuming they are Gaussian) related to each parameter by calculation of the mean and standard deviation. Finally, we define a GMM which combines the OOV and PPX distributions. The GMM acts as a binary classifier able to predict whether any new web page is *relevant* or *irrelevant*.

As the DEV set is noisy, and contains some errors or marginal documents i.e. the *outliers* documents, we introduced a variant to estimate the distributions in our method and remove the outliers from the DEV set. In order to find them, we used the OOV and PPX parameter mean values estimated based on the DEV documents. Any of the DEV documents having a PPX and/or an OOV score either too high or too low, given the mean values is considered as an outlier.

The approach using the variant is named the *restricted* method, as oppsed to the *normal* method, which was first described. For each method, we give the mean and deviation values used to build the GMMs in Table 1. As we can see in this table, removing outliers affects largely both PPX and OOV distributions.

We defined 3 ways to combine the OOV and PPX distributions estimated during the GMM creation: OOV+PPX, OOV alone and PPX alone. The F filtering function of our GMM is defined as:

$$F = Mp + c \times SDp \quad (3)$$

where $p \in \{OOV+PPX, OOV, PPX\}$ Mp and SDp corresponds to the mean and standard deviation related to p. Relying on some preliminary experiments, we chose $c \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$ and forces the standard deviation to variate. Bigger is c or larger is SDp and more documents will be conserved by the GMMs. Conversely, smaller is c or tighter is SDp and more documents will be filtered out by the GMMs. Based on the over-

all c and p values, plus the two ways of creating GMMs, we built a total of 42 GMMs.

	normal		restricted	
	M	SD	M	SD
OOV	1.74	1.98	1.46	1.12
PPX	210.2	252.9	187.6	106.1

Table 1: Mean (M) and standard deviation (SD) estimated for the OOV and PPX parameters and the *normal* vs. *restricted* methods.

3.2.1 Data

The data used in our work is split in 3 corpora: the documents collection used in the LM creation, the DEV documents set used to generate the GMMs and the corpus of documents (french5G) used to test our filtering method during the experiments.

The first corpora is about 2G words. It is composed of French news articles in journalistic style. 85% of them come from newspapers e.g. Le Monde, AFP, and web newspapers e.g. Google news, Yahoo!.

The second corpora counts 509 documents. This corpora has been released behind the previous QA evaluation campaign we participated in (Quintard et al., 2010). It gathers documents containing only adjudicated answers to the evaluation questions found by the systems participant. As a control, we verified that the GMMs we build have rejected less than 10% of the DEV documents.

The last corpora count 499734 French web pages, provided by the Quaero project.

4 Evaluation

4.1 Experimental setup

4.1.1 Ritel-qa

The QA system used in our experiments is presented in details in (Rosset et al., 2008).

The same complete and multilevel analysis is carried out on both questions and documents. The analysis identifies about 300 different types of entities.

From the question analysis, the system build a search descriptor that contains the important elements of the question, the question class predicted from them, and the possible answer types with associated weights. This search descriptor is used by our IR engine to retrieve documents and snippets (Rosset et al., 2008). Then answer extractions and validation procedures are applied (Bernard et al., 2009).

4.1.2 List selection

We submitted to each GMM induced in Section 3 the entire french5G corpus and obtained 42 different lists of a-priori relevant documents used during our experiments. Table 2 shows the quantity of documents composing these lists according to each GMM, as a ratio of the total number of documents in the corpus. We also created the *full-list*, which is composed of all french5G documents.

All the lists were used to feed a filter we plugged in our QA chain to refine the original documents selection made by the system during the IR step. To reduce the number of eligible documents for searching answers we intersect the list of documents retrieved by the system during the IR step with one of the 43 lists. The objective of this filter is to help the QA system to choose the best documents given an estimation of their quality and the question.

For the tuning of answer selection parameters of our QA system, we use a set of 722 *factoid* questions and answers references (Quintard et al., 2010) as well as the 43 document lists provided by the filtering module. For all the possible configurations of parameters, the system provides results for the complete QA chain. These results after tuning serve as a basis for selecting the best document lists.

We defined two different list selection methodologies. In the first one (methodology-1), each question class is associated to the same list: the list for which the QA system obtains the best global success rate. In the second one (methodology-2), the best per-class list is selected, for each of the most frequent question class found throughout the training set. In this case, based on the different success rates obtained per class after tuning, the filtering module automatically determines how to associate question class and document list.

4.2 Results

We evaluated the performance of the different document lists on a test set of 309 *factoid* questions (Quintard et al., 2010) independant from the training set.

For each document list selection methodology, we give in table 3 the results obtained by the system using the best lists according to the tuning (best-1,2) as well as the results it obtained

method	**normal**							**restricted**						
c	0	0.5	1	1.5	2	2.5	3	0	0.5	1	1.5	2	2.5	3
OOV+PPX	27.0	49.5	65.3	75.4	81.8	86.0	88.7	21.0	33.9	45.3	54.7	62.7	69.1	74.0
OOV	39.1	58.7	72.8	81.8	87.4	91.0	93.3	32.9	45.4	56.3	65.2	72.6	78.2	82.5
PPX	47.3	71.9	82.2	87.2	90.1	91.8	93.0	39.9	55.5	66.2	73.1	78.0	81.6	84.2

Table 2: Quantity of a-priori relevant documents per lists, as a ratio of the French Quaero corpus french5G, for each of the 42 GMMs obtained with different distribution combination of LM parameters (OOV, PPX, OOV+PPX), c value ($c \in [0-3]$) and method (normal vs. restricted).

using the *full-list* (baseline). For instance, for methodology-1, the best document list (o+p2.5n) has been generated based on the GMM merging OOV and PPX information with a c value of 2.5 following the *normal* approach for its creation (see section 3.2). The *baseline* system does not use our approach for document filtering. The lines 2 to 4 and 5 to 7 of table 3 shows the results obtained with methodology-1 and -2, respectively.

Results are measured given the classical QA evaluation metrics: precision (or top-1), mean reciproqual rank and recall (or top-10).

S	**L**	**Qc**	**P**	**MRR**	**R**	**#q**
baseline	full	all	31.7	39.5	53.4	309
best-1	o+p2.5n	all	33.0	40.6	55.0	309
best-qc	p2.5n	loc	57.6	64.5	75.8	66
best-1	o+p2.5n	loc	54.5	62.8	75.8	66
best-2	-	all	31.1	39.4	54.7	309
-	p1.5r	time	29.2	38.5	56.2	48
-	p2n	loc	54.5	63.2	75.8	66

Table 3: Results on the test data following methodology-1 (top) and methodology-2 (bottom). **S**: system; **L**: document list selection mode; **Qc**: question class; **P**: precision; **MRR**: mean reciprocal rank; **R**: recall; **#q**: number of questions.

According to the best-1 line, using a document filtering improves the overall results: all the metrics are improved by $\sim$1% absolute. In methodology-1 one single list is chosen for all question classes, which could be sub-optimal locally, i.e. given a specific question class. This is shown in the first part of the Table 3 with the oracle results (best-qc) associated to the *localization* class. If the system had used this list instead of the general best list, the results could have been improved on this question class by almost 3% of precision. The other methodology (choosing the best list for each question class) seemed then to be more optimal. Although, using methodology-2 we observed a significant gain on tuning data, this gain was not preserved with new data (best-2). This is due to an insufficient amount of training data for each question class.

We see also that normal lists give better results than the restricted ones. This shows that, given the small number of DEV documents used to generate the GMMs, the filtering should aim only at removing unarguably bad documents where the system would not be able to extract any correct answers. If a more decent number of developement documents would have been available, more precise filtering techniques could have been more successful.

S	**L**	**Qc**	**P**	**MRR**	**R**	**#q**
best-1	o+p2.5n	all	33.0	40.6	55.0	309
baseline	full	all	31.7	39.5	53.4	309
random	-	all	31.4	39.2	53.4	309
*09*best-1	p3r	all	28.2	34.2	45.6	309
*09*baseline	full	all	24.6	31.6	45.6	309

Table 4: Results on the test data using methodology-1 (best-1, *09*best-1, baseline and *09*baseline) and choosing one among the 42 lists for each question class (random). *09* point out results obtained on our 2009 QA chain. **S**: system; **L**: document list; **Qc**: question class; **P**: precision; **MRR**: mean reciprocal rank; **R**: recall; **#q**: number of questions.

In order to validate our approach we did two controls. First, we compared the results obtained with the QA system using our document filtering with the best system obtained following methodology-1 (best-1) against a system using a random document lists selection (random). Then, we reproduced the experiments following the same methodology on our 2009 QA chain completed with filtering. As we can see in table 4, the random selection is worse than methodology-1 and the older version of our system using the filtering method (*09*best-1) outperforms the corresponding baseline system (*09*baseline). Thereby, we confirmed our documents selection method is usefull for a QA system.

5 Conclusion and perspectives

We have presented a method to evaluate the intrinsic quality of web pages to be used in a question-answering system. The approach is twofold: first the intrinsic relevancy of a document is determined using a n-gram language model and then a GMM-based classifier decides whether this document may be considered as relevant for searching answers to any question. The GMMs are built based on the perplexity, the out-of-vocabulary ratio and a combination of these two informations. For this purpose, we completed the classical QA model with a filtering on top of the document retrieval, before the extraction of answers.

The results show that the a-priori document filtering approach provides a significant improvement of the QA system, for all measures.

We observed the best lists are not the most filtering ones but those which kept 80%-90% of the documents. We also observed the best results obtained on the tuning using a per-class decision about the lists were not confirmed on the test data, showing the amount of training data is insufficient to leverage the question classification at this point.

The parameters used in our experiments are very primitive. They are able to filter out only extremely irrelevant documents. In addition to the intrinsic relevancy, we plan testing extra features to support the filtering process. Given the nature of the QA task, we think semantic features like document topics (extracted from URL) could be very useful.

We also think it would be interesting to investigate in the direction of creating specialized classifiers (based on SLMs or other) to support the documents classification according to outputs of linguistics analyzers.

Size and content of web documents are extremely variable. Reducing this variability should help Web-oriented QA. Thus, we plan to segment the documents prior to filtering.

Acknowledgments

This work has been partially financed by OSEO under the Quaero program.

References

G. Bernard, S. Rosset, O. Galibert, E. Bilinski, and G. Adda. 2009. The limsi participation in the qast 2009 track: experimentating on answer scoring. In *CLEF 2009*, Corfu, Grece, September.

Surya Ganesh and Vasudeva Varma. 2009. Exploiting the use of prior probabilities for passage retrieval in question answering. In *RANLP-2009*, pages 99–102, Borovets, Bulgaria, September. Association for Computational Linguistics.

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss I, 1990. *Readings in Speech Recognition*, chapter Self-organized language modeling for speech recognition, pages 450–506. Morgan Kaufmann.

M. Pardino, J.M. Gómez, H. Llorens, R. Múnoz-Terol, B. Navarro-Colorado a nd E. Saquete, P. Martínez-Barco, P. Moreda, and M. Palomar. 2008. Adapting ibqas to work with text transcriptions in qast task: Ibqast. In *CLEF 2008*, Aarhus, Denmark, September.

Ludovic Quintard, Olivier Galibert, Gilles Adda, Brigitte Grau, Dominique Laurent, Véronique Moriceau, Sophie Rosset, Xavier Tannier, and Anne Vilnat. 2010. Question answering on web data: The qa evaluation in quæro. In *LREC'10*, Valletta, Malta, may.

Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here. *Proceedings of the IEEE*, 88(8):1270–1278.

S. Rosset, O. Galibert, G. Bernard, E. Bilinski, and G. Adda. 2008. The limsi participation to the qast track. In *Working Notes of CLEF 2008 Workshop*, Aarhus, Denmark, September.

V. Varma, P. Bysani, K. Reddy, V.B. Reddy, S. Kovelamudi, S.R. Vaddepally, R. Nanduri, K. Kumar N, S. Gsk, and P. Pingali. 2010. Iiit hyderabad in guided summarization and knowledge based population. In *TAC 2010*, Gaithersburg, Maryland USA, November.