A High Precision Information Retrieval Method for WiQA

Constantin Orăsan and Georgiana Puşcaşu

Research Group in Computational Linguistics
University of Wolverhampton, UK
{C.Orasan, georgie}@wlv.ac.uk

Abstract. This paper presents Wolverhampton University's participation in the WiQA competition. The method chosen for this task combines a high precision, but low recall information retrieval approach with a greedy sentence ranking algorithm. The high precision retrieval is ensured by querying the search engine with the exact topic, in this way obtaining only sentences which contain the topic. In one of the runs, the set of retrieved sentences is expanded using coreferential relations between sentences. The greedy algorithm used for ranking selects one sentence at a time, always the one which adds most information to the set of sentences without repeating the existing information too much. The evaluation revealed that it achieves a performance similar to other systems participating in the competition and that the run which uses coreference obtains the highest MRR score among all the participants.

1 Introduction

The Research Group in Computational Linguistics at the University of Wolverhampton has participated in the English task of the WiQA pilot task [2] with a system that employs high precision retrieval and shallow rules to rank candidate passages. Given that this was the first time such a task was proposed, our emphasis was less on fine tuning the system, and more on exploring the issues posed by the task and developing a complete system that can participate in the competition. The main goal of a system participating in the WiQA competition is the extraction of sentences from Wikipedia which contain the most relevant and novel information with respect to the content of the Wikipedia page describing the topic. Therefore, many important issues must be taken into consideration for a WiQA system development. First, it is necessary to identify the topic relevant sentences. Then, in the case of our system, the process of selecting and ranking the most important and novel sentences adopts a greedy strategy that adds to the topic document sentences from the retrieved set that bring the highest information gain. The relevant sentences are then presented as the system's output in the order they were selected.

This paper presents the outline of our system and the results of its evaluation. The structure of the paper is as follows: Section 2 describes the architecture of the system. The three runs submitted in the competition are described in Section 3. The evaluation results are presented and discussed in Section 4. In the last section, conclusions are drawn and future system enhancements are considered.

C. Peters et al. (Eds.): CLEF 2006, LNCS 4730, pp. 561–568, 2007.

2 System Description

WiQA, as a newly introduced pilot task, cannot be characterised by typical approaches for system architecture. Still, three perspectives were foreseen for addressing it: based on Information Retrieval (IR), based on Question Answering (QA) or based on multi-document summarisation capabilities. In the IR perspective, a retrieval engine is the main source of candidate snippets which are then filtered and scored by a set of linguistic processors. The advantage of this method is that it does not rely on too much information about the topics to be processed. In contrast, in the QA-based approach the system knows what kind of information needs to be presented about a topic and employs a QA system to retrieve this information. For the WiQA task, whenever the answer is not contained in the topic page, the sentence containing the answer is returned as candidate snippet. Still, this template-based question answering process is limited to certain factoid questions, therefore losing access to possibly important information that does not fit any template. This approach can prove effective because with a small fraction of question types one could account for relevant information a user is interested in knowing, and the retrieved passages would thus have higher chances of being considered relevant, non-repetitive, and innovative in the case of questions that are not supported by the topic document. The approach based on multi-document summarisation treats Wikipedia as a collection of documents which needs to be summarised, and produces a multi-document summary focused on the topic. In some aspects this approach is similar to the one based on information retrieval because it also requires a retrieval engine, but the techniques employed to extract snippets can differ.

After considering the advantages and disadvantages, as well as the resources we had available to approach the task, we decided to develop an IR-based system for WiQA. Its architecture is depicted in Figure 1 and consists of four core components:

1. high accuracy retrieval engine
2. snippet pre-processor
3. snippet ranking module
4. snippet post-processor

In the remainder of this section each of these components is described in detail.

2.1 High Accuracy Retrieval Engine

The role of this component is to retrieve snippets on the selected topic from Wikipedia and to filter out snippets according to some predefined rules. The information retrieval engine used in our system is Lucene [3]. Given that the topics referred to named entities such as persons, locations or organisations, it was decided not to process the topic in any way and to query the retrieval engine for the exact string identifying the topic, in this way making sure that we have a high precision retrieval (i.e. all the retrieved snippets contain the string identified by the topic). Given that the majority of topics were unambiguous, most of the retrieved snippets were related to the topic.

The drawbacks of this method are that for several topics we were unable to obtain any candidates due to the way the topic was formulated, and that sentences which

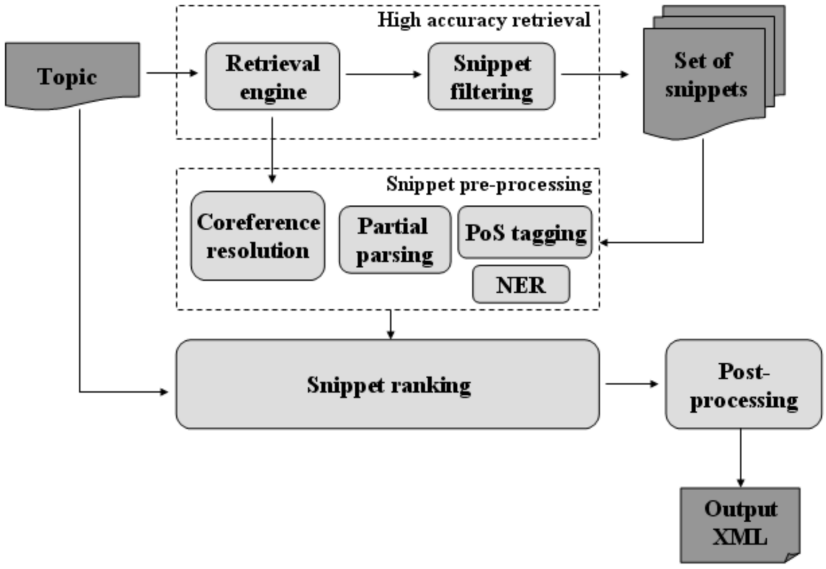

Fig. 1. System architecture

refer to the topic but do not contain the exact string identifying the topic cannot be extracted in this manner. The former problem appeared for topics such as *Ministry of Defence (Singapore)*, where the string *Singapore* was provided to disambiguate between different *ministries of defence*. In this case, when Lucene was queried with the string *Ministry of Defence (Singapore)*, no result was returned.

Not all the snippets retrieved by the engine can and should be used. The **Snippet filtering module** filters out snippets which are not appropriate for processing. First of all, the snippets extracted from the topic document have to be filtered out because they should not be considered for ranking, but are used to determine the ranking of the extracted snippets. Depending on the approach taken, snippets from documents directly connected to the topic document can be ignored because it can be argued that they contain information which is easily accessible to the reader of a document. Some of the runs we submitted used this filter.

In addition to the sentences extracted by our system, it was noticed that there are sentences which refer to the topic, but do not contain the actual string which identifies the topic. As a result, they cannot be directly extracted by the retrieval engine. Because in the majority of cases, these sentences refer to the topic using a referential expression, a coreference resolution module was added at the **Snippet pre-processing** stage and used in one of our runs.

The formatting of Wikipedia also determined us to implement some further filters. In some cases the snippets to be extracted are parts of a list and therefore it needs to be decided whether to ignored them, or whether to extract only the item in the list which contains the topic. Our runs used both types of filtering.

2.2 Snippet Pre-processing

At this stage, both the snippets extracted by the retrieval component and the sentences contained in the Wikipedia page describing the topic are annotated with linguistic information. Conexor's FDG Parser [4] is used to produce morpho-syntactic information and partial parsing trees for snippets. On the basis of this information, we identify noun phrases (NPs) and verb phrases (VPs), and to convert the snippets into logical forms. We distinguish between several predicates connecting nouns / verbs and corresponding to different syntactic relations, such as the relation between a verb and the head of its subject NP, between a verb and its object NP head, between a verb and its other complements and adjuncts, between two verbs and between two nouns. The derived logical forms have been only superficially employed at the snippet ranking stage, as more investigation is needed into ways of taking better advantage of the information they provide.

Given that many of the snippets contain named entities which can prove important at the ranking stage, GATE [1] was used to extract this information. It should be pointed out that the named entity recogniser provided by GATE has been used without any customisation and as a result it failed to recognise a large number of entities. In the future we plan to adapt the named entity recogniser in order to improve its recognition accuracy.

As already mentioned, not all the snippets relevant to the topic can be extracted by the search engine because they do not contain the topic string. In light of this, a rule based coreference resolver was used to identify sentences related to the topic. The rules implemented referred mainly to presence of a substring from the topic in a sentence following an extracted snippet. For example, in the following sentences:

(a) *After returning home, Liang went on to study with* ***Kang Youwei****, a famous Chinese scholar and reformist who was teaching at Wanmu Caotang in Guangzhou.*
(b) ***Kang****'s teachings about foreign affairs fueled Liang's interest in reforming China.*

using coreference resolution, it is possible to determine that *Kang* refers to *Kang Youwei*, and in addition to sentence (a), sentence (b) should be also extracted as candidate for the *Kang Youwei* topic. By using coreference resolution it is possible to expand the list of candidate snippets without introducing too much noise. In order to achieve this, documents which contain useful snippets are processed by the coreference resolver which identifies additional snippets.

2.3 Snippet Ranking

The **Snippet ranking** module uses information obtained at the pre-processing stage by the partial parser and NE recogniser in order to rank the retrieved snippets according to importance and novelty with respect to the topic document. The ranking algorithm is based on a greedy strategy which scores snippets by measuring how much extra information a snippet brings to a topic, and extracts those with the highest score. The algorithm is presented below:

1. Let **T** be the set of sentences extracted from the Wikipedia page corresponding to the topic, **R** the set of sentences retrieved from other documents, and **C** the current set of relevant sentences. **C** is initially empty (**C** = $\emptyset$).
2. For each retrieved sentence **s** $\in$ **R**, calculate the information gain obtained by adding it to the set of topic sentences **T** unified with the current set of relevant sentences **C** (how much information do we gain having **T**$\cup$**C**$\cup\{$**s**$\}$ when compared to **T**$\cup$**C**).
3. Choose the sentence $\mathbf{s}_{max} \in$ **R** with the highest information gain score, add it to the set of relevant sentences **C**, and, at the same time, eliminate it from the set of retrieved sentences (**C** = **C**$\cup\{\mathbf{s}_{max}\}$, **R** = **R**$\setminus\{\mathbf{s}_{max}\}$).
4. Continue selecting sentences that maximise the information gain score by repeating steps **2** and **3** until the maximum information gain value falls below a certain threshold, or until there is no candidate sentence left in **R**.

The relevant/novel sentences resulted by applying this algorithm are presented, in the order they were added to set **C**, as the output of the **Snippet ranking** module.

2.4 Post-processing

The initial purpose of the **Snippet post-processor** was to take the results from the **Snippet ranking** module and produce the XML file to be submitted for evaluation. After a script was written to produce this output, it was noticed that it did not produce the necessary output due to errors introduced by the FDG Parser. Because the input to the FDG tagger contained all kind of unusual characters, in a large number of cases snippets were wrongly split into several sentences. In other cases snippets were merged together even though they were on different lines due to the fact that they did not contain the final full stop. These errors were identified too late to be able to make any changes to the pre-processing module. In light of this, in order to produce the XML output required for the evaluation, a simple overlap measure between the ranked set of relevant sentences, as they were segmented by FDG, and the original snippets has been implemented in order to determine the source document and the snippet IDs, as required in the XML document. Due to the simplicity of this measure, a number of errors have been introduced in the output.

3 Description of Submitted Runs

We have submitted three runs referred to in the following as *one*, *one-old* and *two*. All three use the same retrieval engine, named entity recogniser, part-of-speech tagger, partial parser and post-processor. The runs *one* and *one-old* have in common the same set of extracted snippets, but the method in which they are ranked differs. For run *two* the set of snippets is different, but the same ranking method as that in run *one* is used.

For runs *one* and *one-old* the set of candidate snippets consists of only snippets which contain the topic string, without employing coreference resolution to expand the set. In addition, the sentence candidate set includes items from lists which contain the topic string, but not the whole lists. Snippets from documents directly linked to the topic document were discarded. In contrast, run *two* employs coreference resolution to

extend the set of candidates, does not filter out snippets from documents linked to the topic document, and does not include snippets from lists. As a result of this, the snippets contained less noise, but their number was also reduced.

The ranking method of run *one-old* differs from the ranking method of runs *one* and *two* in the formula employed for computing the information gain obtained by adding a sentence to a set of sentences. The run *one-old* uses formula (F1), while the runs *one* and *two* use a modified version of (F1) by increasing / decreasing the information gain score according to the following criteria:

- the score is penalised if no verbs are present in the snippet, or if the number of NEs or the number of occurring years (any 4-digit number greater than 1000 and smaller than 2020 is considered a possible year) are above 20 (if any of these cases apply, there is a high probability the snippet is a list or a list element);
- a small penalty applies in the case the topic is preceded by a preposition;
- the sentence is given a bonus if the topic is its subject;
- the sentence is penalized proportionally with the number of third person pronouns it contains.

(F1) IG(s, Set) = $\sum$ (P(t|Set) * log P(t|Set) - P(t|Set$\cup${s}) * log P(t|Set$\cup${s}))

In the above formula, **s** and **Set** are the sentence and the sentence set we compute the information gain score for, **t** is a token which ranges over the set of NEs, nouns, verbs and logical forms present in the sentence, and P(t|Set) is the probability of the token **t** to appear in the **Set**.

4 Results and Analysis

As one of the purposes of the WiQA pilot task was to experiment with different measures for evaluating systems' performance, several scores are used at the assessment stage. The evaluation measures used were: *yield*, *average yield per topic*, *MRR* and *precision* at top 10 snippets and were proposed by the organisers of the task [2]. Table 1 presents our results in the evaluation as well as the overall results obtained by all the participants. As our system targeted high precision, at the cost of low recall, by retrieving only those snippets that almost certainly referred to the query topic, we did not expect high *yield* and *average yield* scores. As one can notice in Table 1, our *yield* and *average yield* scores are close to the median value achieved by all WiQA participants. One of our runs (run *two*) established the maximum value for the MRR measure. The low precision can be explained by the fact that our runs contains a large number of repetitions introduced by the **Post-processing module**. If the repeated entries were discarded from the submission, the precision would have been 0.35, 0.34 and 0.35 for runs *one-old*, *one* and *two* respectively.

Comparison between runs reveals some interesting results. The weighting method used for runs *one* and *two* lead to better values for MRR, but the retrieval module used for runs *one-old* and *one* ensure better precision. The inclusion of coreference resolution and of snippets from pages directly linked to the topic page leads to the best value for MRR. In light of this, it would be interesting to develop a method which integrates

Table 1. Evaluation results

Run	Yield	Average yield per topic	MRR	Precision
One-old	142	2.32	0.57	0.30
One	135	2.21	0.58	0.28
Two	135	2.21	0.59	0.25
Min	58	1.52	0.30	0.20
Max	220	3.38	0.59	0.36
Median	158	2.46	0.52	0.32

them into the retrieval method used in runs *one* and *one-old*. Investigation of the results obtained in run *two* reveals that coreference resolution proposes a large number of very good snippets for ranking, but very few of them are present in the output of the system. In light of this, the ranking method will need to be changed to take into account the fact that a snippet was extracted by the coreference resolver and as a result it may not have the topic string or a large number of named entities.

Coreference resolution should be applied not only to sentences which follow a sentence including the topic. It should be applied also to sentences which contain the topic. For example in the sentence *In 2001* ***he*** *took over the role of Shylock in the Stratford Festival of Canada production of The Merchant of Venice after Al Waxman, who was originally scheduled to play the part, died.* it is not possible to know which is the antecedent of *he* without a coreference resolver, and therefore the snippet will be marked as unimportant.

Comparison between the results corresponding to the three different runs revealed inconsistencies in the evaluation process. In this analysis, we first compared automatically the first occurrence of each snippet to identify whether they have been annotated identically in the two evaluation files. Only the first occurrence has been considered due to the fact that a second occurrence is repetitive and consequently receives a different annotation. An inconsistency is considered to be any difference in the values of the attributes corresponding to the first occurrence of a snippet in the two runs for which the comparison is made. This analysis revealed:

* 14 inconsistencies between runs *one* and *one-old*
* 15 inconsistencies between runs *one* and *two*
* 11 inconsistencies between runs *one-old* and *two*

The next stage of the analysis was performed only for the runs *one* and *one-old* and involved manually checking the automatically extracted inconsistencies. This stage has revealed that, out of 14 automatically extracted inconsistencies, there are 10 true inconsistencies, that is snippets that should have received the same annotation. For example, the snippet

(1) *Police cordoned off several areas in the capital of Mal, around the National Security Services, Shaheed Ali Building (Police Headquarters), Republic Square, People's Majlis and Maldivian Democratic Party Headquarters.*

is marked as supported, important, novel and not repeated in run *one-old*, whereas in run *one* it is annotated as a repetitive snippet, even though it is the first snippet presented for topic 47 (*Maldivian Democratic Party*).

There were also four cases for which the annotation provided by the evaluators is correct and the inconsistency is justified. For example in the case of topic 44 (*Al-Arabiya*) the information contained in a supported, important, novel and not repeated snippet of run *one-old* (2) was presented in another form by a previous snippet in run *one* (3), therefore being correctly tagged as repetitive.

(2) *Through the first three months of 2004, a number of attacks on journalists in the West Bank and Gaza Strip have been blamed on the Brigades as well, including the attack on the Arab television station Al-Arabiya's West Bank offices by masked men self-identifying as members of the Brigades.*

(3) *Through the first three months of 2004, a number of attacks on journalists in the West Bank and Gaza Strip have been blamed on the Al-Aqsa Martyrs' Brigades, most clearly the attack on the Arab television station Al-Arabiya's West Bank offices by masked men self-identifying as members of the Brigades.*

These inconsistencies in the evaluation results are not a surprise given that the measures used to assess the snippets are subjective.

5 Conclusions

This paper presented Wolverhampton University's submission to WiQA task. The method chosen for this task combines a high precision, but low recall information retrieval approach with a greedy algorithm to rank the retrieved sentences. The results of the evaluation revealed that it achieves a performance similar to other systems participating in the competition and that one of the submitted runs scores the highest MRR score among all the participants.

System analysis pinpointed several possible improvements for the future. As noticed in the evaluation, better precision can be obtained by filtering out duplicates at the post-processing stage. More attention will need to be paid to the input of the pre-processing stage so the sentences are no longer wrongly identified. As mentioned previously, in order to compute the relevance of each sentence, the sentences are converted into a logical form. The predicates used for this representation are currently little used, but we believe that the ranking can be improved by using them.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (2002)
2. Jijkounand, V., de Rijke, M.: Overview of WiQA 2006. In: Proceedings of CLEF2006 (2006)
3. Lucene: Lucene. `http://lucene.apache.org/`
4. Tapanainen, P., Jaervinen, T.: A Non–Projective Dependency Parser. In: Proceedings of the 5th Conference of Applied Natural Language Processing, ACL (1997)