Answering Questions from Multiple Documents – the Role of Multi-Document Summarization

Pinaki Bhaskar
Department of Computer Science & Engineering,
Jadavpur University, Kolkata – 700032, India
pinaki.bhaskar@gmail.com

Abstract

Ongoing research work on Question Answering using multi-document summarization has been described. It has two main sub modules, document retrieval and Multi-document Summarization. We first preprocess the documents and then index them using Nutch with NE field. Stop words are removed and NEs are tagged from each question and all remaining question words are stemmed and then retrieve the most relevant 10 documents. Now, document graph-based query focused multi-document summarizer is used where question words are used as query. A document graph is constructed, where the nodes are sentences of the documents and edge scores reflect the correlation measure between the nodes. The system clusters similar texts from the graph using this edge score. Each cluster gets a weight and has a cluster center. Next, question dependent weights are added to the corresponding cluster score. Top two-ranked sentences of each cluster is identified in order and compressed and then fused to a single sentence. The compressed and fused sentences are included into the output summary with a limit of 500 words, which is presented as answer. The system is tested on data set of INEX QA track from 2011 to 2013 and best readability score was achieved.

1 Introduction

With the explosion of information in Internet, Natural language Question Answering (QA) is recognized as a capability with great potential. Traditionally, QA has attracted many AI researchers, but most QA systems developed are toy systems or games confined to laboratories and to a very restricted domain. Several recent conferences and workshops have focused on aspects of the QA research. Starting in 1999, the Text Retrieval Conference (TREC)[1] has sponsored a question-answering track, which evaluates systems that answer factual questions by consulting the documents of the TREC corpus. A number of systems in this evaluation have successfully combined information retrieval and natural language processing techniques. More recently, Conference and Labs of Evaluation Forums (CLEF)[2] are organizing QA lab from 2010.

INEX[3] has also started Question Answering track. INEX 2011 designed a QA track (SanJuan et al., 2011) to stimulate the research for real world application. The Question Answering (QA) task is contextualizing tweets, i.e., answering questions of the form "what is this tweet about?" INEX 2012 Tweet Contextualization (TC) track gives QA research a new direction by fusing IR and summarization with QA. The first task is to identify the most relevant document, for this a focused IR is needed. And the second task is to extract most relevant passages from the most relevant retrieved documents. So an automatic summarizer is needed. The general purpose of the task involves tweet analysis, passage and/or XML elements retrieval and construction of the answer, more specifically, the summarization of the tweet topic.

Automatic text summarization (Jezek and Steinberger, 2008) has become an important and timely tool for assisting and interpreting text information in today’s fast-growing information age. An Abstractive Summarization ((Hahn and Romacker, 2001) and (Erkan and Radev, 2004)) attempts to develop an understanding of the main concepts in a document and then expresses those concepts in clear natural language. Extractive Summaries (Kyoomarsi et al., 2008) are formu-

[1] http://trec.nist.gov/
[2] http://www.clef-initiative.eu//
[3] https://inex.mmci.uni-saarland.de/

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 14–21,
Hissar, Bulgaria, 9-11 September 2013.

lated by extracting key text segments (sentences or passages) from the text, based on statistical analysis of individual or mixed surface level features such as word/phrase frequency, location or cue words to locate the sentences to be extracted. Our approach is based on Extractive Summarization.

In this paper, we describe a hybrid Question Answering system of document retrieval and multi-document summarization. The document retrieval is based on Nutch[4] architecture and the multi-document summarization system is based graph, cluster, sentence compression & fusion and sentence ordering. The same sentence scoring and ranking approach of Bhaskar and Bandyopadhyay (2010a and 2010b) has been followed. The proposed system was run on the data set of three years of INEX QA track from 2011 to 2013.

2 Related Work

Recent trend shows hybrid approach of question answering (QA) using Information Retrieval (IR) can improve the performance of the QA system. Schiffman et al. (2007) successfully used methods of IR into QA system. Rodrigo et al. (2010) removed incorrect answers of QA system using an IR engine. Pakray et al. (2010) used the IR system into QA and Pakray et al. (2011) proposed an efficient hybrid QA system using IR.

Tombros and Sanderson (1998) presents an investigation into the utility of document summarization in the context of IR, more specifically in the application of so-called query-biased summaries: summaries customized to reflect the information need expressed in a query. Employed in the retrieved document list displayed after retrieval took place, the summaries' utility was evaluated in a task-based environment by measuring users' speed and accuracy in identifying relevant documents.

A lot of research work has been done in the domain of both query dependent and independent summarization. MEAD (Radev et al., 2004) is a centroid based multi document summarizer, which generates summaries using cluster centroids produced by topic detection and tracking system. NeATS (Lin and Hovy, 2002) selects important content using sentence position, term frequency, topic signature and term clustering. XDoX (Hardy et al., 2002) identifies the most salient themes within the document set by passage clustering and then composes an extraction summary, which reflects these main themes. Graph-based methods have been also proposed for generating summaries. A document graph-based query focused multi-document summarization system has been described by Paladhi et al. (2008) and Bhaskar and Bandyopadhyay (2010a and 2010b).

In the present work, we have used the IR system as described by Pakray et al. (2010 and 2011) and Bhaskar et al. (2011) and the automatic summarization system as discussed by Bhaskar and Bandyopadhyay (2010a and 2010b) and Bhaskar et al. (2011).

3 System Architecture

In this section the overview of the system framework of the current INEX system has been shown. The current INEX system has two major sub-systems; one is the Focused IR system and the other one is the Automatic Summarization system. The Focused IR system has been developed on the basic architecture of Nutch, which use the architecture of Lucene[5]. Nutch is an open source search engine, which supports only the monolingual Information Retrieval in English, etc. The Higher-level system architecture of the combined Tweet Contextualization system of Focused IR and Automatic Summarization is shown in the Figure 1.

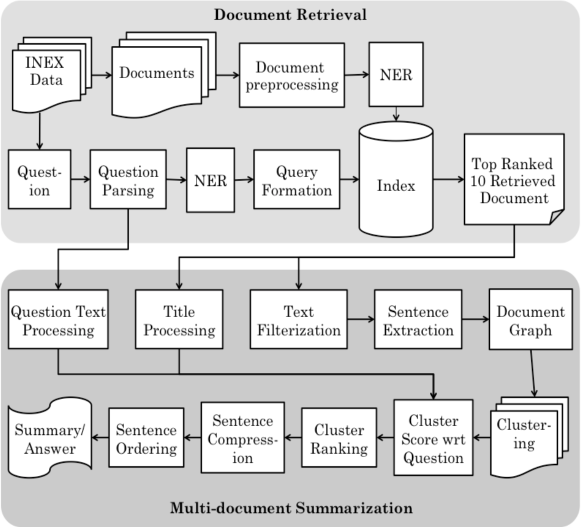

Figure 1. Higher-level system architecture

4 Document Retrieval

4.1 Document Parsing and Indexing

The web documents are full of noises mixed with the original content. In that case it is very diffi-

[4] http://nutch.apache.org/

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

cult to identify and separate the noises from the actual content. INEX 2012 corpus had some noise in the documents and the documents are in XML tagged format. So, first of all, the documents had to be preprocessed. The document structure is checked and reformatted according to the system requirements.

XML Parser: The corpus was in XML format. All the XML test data has been parsed before indexing using our XML Parser. The XML Parser extracts the Title of the document along with the paragraphs.

Noise Removal: The corpus has some noise as well as some special symbols that are not necessary for our system. The list of noise symbols and the special symbols like “&quot;”, “&amp;”, “"”, multiple spaces etc. is initially developed manually by looking at a number of documents and then the list is used to automatically remove such symbols from the documents.

Named Entity Recognizer (NER): After cleaning the corpus, the named entity recognizer identifies all the named entities (NE) in the documents and tags them according to their types, which are indexed during the document indexing.

Document Indexing: After parsing the documents, they are indexed using Lucene, an open source indexer.

4.2 Question Parsing

After indexing has been done, the questions had to be processed to retrieve relevant documents. Each question / topic was processed to identify the question words for submission to Lucene. The questions processing steps are described below:

Stop Word Removal: In this step the question words are identified from the questions. The stop words[6] and question words (what, when, where, which etc.) are removed from each question and the words remaining in the questions after the removal of such words are identified as the question tokens.

Named Entity Recognizer (NER): After removing the stop words, the named entity recognizer identifies all the named entities (NE) in the question and tags them according to their types, which are used during the scoring of the sentences of the retrieved document.

Stemming: Question tokens may appear in inflected forms in the questions. For English, standard Porter Stemming algorithm[7] has been used to stem the question tokens. After stemming all the question tokens, queries are formed with the stemmed question tokens.

4.3 Document Retrieval

After searching each query into the Lucene index, a set of retrieved documents in ranked order for each question is received.

First of all, all queries were fired with AND operator. If at least ten documents are retrieved using the query with AND operator then the query is removed from the query list and need not be searched again. If not then the query is fired again with OR operator. OR searching retrieves at least ten documents for each query. We always ranked the retrieved document using AND operator higher than the same using OR operator. Now, the top ranked ten relevant documents for each question is considered for milti-document summarization. Document retrieval is the most crucial part of this system. We take only the top ranked ten relevant documents assuming that these are the most relevant documents for the question from which the query had been generated.

5 Multi-Document Summarization

5.1 Graph-Based Clustered Model

The proposed graph-based multi-document summarization method consists of following steps:

(1) The document set $D = \{d_1, d_2, \dots d_{10}\}$ is processed to extract text fragments, which are sentences in this system as it has been discussed earlier. Let for a document d_i, the sentences are $\{s_{i1}, s_{i2}, \dots s_{im}\}$. Each text fragment becomes a node of the graph.

(2) Next, edges are created between nodes across the documents where edge score represents the degree of correlation between inter-documents nodes.

(3) Seed nodes are extracted which identify the relevant sentences within D and a search graph is built to reflect the semantic relationship between the nodes.

(4) Now, each node is assigned a question dependent score and the search graph is expanded.

(5) A question dependent multi-document summary is generated from the search graph.

Each sentence is represented as a node in the graph. The text in each document is split into

[6] http://members.unine.ch/jacques.savoy/clef/

[7] http://tartarus.org/~martin/PorterStemmer/java.txt