How the Text Summarization affects the Information Retrieval based Question Answering

Gönenç Ercan
Hacettepe University Computer Engineering, Ankara, Turkey
gonenc@gmail.com

Bahadır Adak
Hacettepe University Computer Engineering, Ankara, Turkey
bahadiradak@gmail.com

Eyüpcan Bodur
Hacettepe University Computer Engineering, Ankara, Turkey
bodureyupcanb@gmail.com

Abstract

This paper presents a novel experiment which a text summarization technique is how will affect the question answering system. We investigate a new training paradigm for Information Retrieval based question answering. Normally after the document retrieval part, directly extract passages of documents and narrows possible answer set in the classic information retrieval approach. Instead of collecting large-sized documents first and then obtaining small-sized passages, we want to show how the use of text summaries of these documents will have an effect on the question answering system. Text summarization is reducing a text document into a short set of words or paragraph that contains the key meaning of the text. In that reason we want to test that if we train model with summarized documents how it contributes the evaluation result of that model.

Keywords: Question Answering, Text Summarization, Information Retrieval

I. INTRODUCTION

Question Answering (QA) is the process of retrieving correct information or exact answer from a large collection of documents against a natural language question. Also, it is a fast-growing research area that combines research from different, but related, fields which are Information Retrieval (IR), Information Extraction (IE) and Natural Language Processing (NLP). When we look at first examples of information retrieval systems we can see that they focused on the retrieval of the most relevant documents from a collection with given a set of keywords. In the next generation approaches, it was based on receiving the most relevant passages in the most important documents. However, despite reducing the search space, users cannot deal with all the available information founded by an information retrieval system because of the complexity of the query.

In general, question answering systems have three components such as question processing, document processing (information retrieval) and answer processing. The first step begins with extracting information from the question with question processing. This process has two important sub-process. First one; query formulation for prepare query that is going to send to an IR Engine and second one; answer type detection that tells us what kind of name entity we are looking for. After passing these sub-stages, indexing of already saved documents is performed. Thus, documents are available to apply a query and from those documents we get relevant documents with Information Retrieval. Passage retrieval helps extract passages of documents and narrows possible answer set. In the last step, those passages are processed in answer processing. Looking for output of answer type detection helps that step and then returning a possible answer.

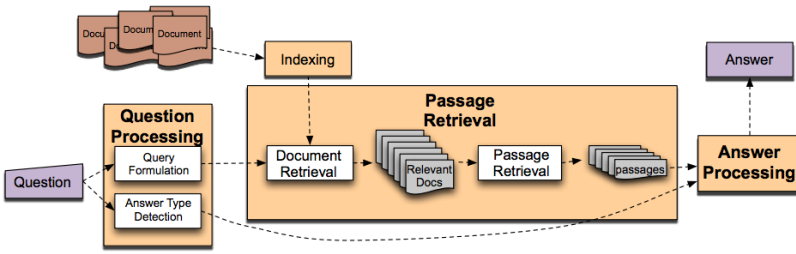

Fig. 1. General Architecture of a QA System

Question processing component is very important because if the module does not start correctly, it will make problems for other part of QA system. Also answer processing as important as question processing because systems are often required to rank and validate candidate answers according to answer processing.

Text summarization be useful in retrieval of important information from a large textual data and also reduces the size of the text. In this way, important points can be easily extracted from unnecessarily long documents. With this aspect, it is separated from the question answering system. Question answering system is looking for main points and exact answers according to the question.

In our project we thought that how can these two topics affect each other? Information Retrieval based question answering system naturally needs a lot of documents and the number of documents might be a problem when we are looking for correct answer which made by user query. Also, all retrieved documents might not be related to answer of that question so we may have reached the wrong answer due to information pollution. All these thoughts lead us to the idea of how to get a result if text summarization is used in the document retrieval phase of the question answering system. In this project we will test the relationship and between Information Retrieval and text summarization and effect on each other.

II. BACKGROUND

A. Information Retrieval (IR)

QA systems have applications used in a wide variety of tasks, and one of them is information retrieval. Its task that automatically answer the questions asked by humans in natural language using either a structured database or a collection of documents. Information retrieval (IR) is basically finding relevant documents from a database in response to query which made by user. It is one of the most challenging tasks of NLP because of a lot of various count of unstructured data exist in that process. Improvement of deep learning in computer vision and neural networks (CNNs) have relapsed as a popular machine learning paradigm in many other directions of research, including IR.

Traditional learning to rank models employ machine learning techniques over hand-crafted IR features. By contrast, neural models learn representations of language from raw text that can bridge the gap between query and document vocabulary. Unlike classical IR models, these new machine learning based approaches are required large scale training data before they can be deployed [1]. In our project we used Simple Transformers library by HuggingFace to train and evaluate model. Details will be explained in Experiments section.

B. Text Summarization (TS)

Text summarization is a technique for creating a concise and accurate summary of large texts with an emphasis on parts that convey useful information and do not lose its general meaning. Automatic text summarization is designed to convert long documents into short versions, which can be difficult and expensive if they are done manually. Machine learning algorithms use to be trained to identify documents and sections that convey important facts and information before creating the required generalized texts.

The advantage of using a text summarization is the output reduces the reading time. Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. An Abstractive summarization is an understanding of the main concepts in a document and then express those concepts in clear natural language [2].

Summary can be generated through extractive as well as abstractive methods but abstractive methods are highly complex as they need extensive natural language processing. Therefore, research community is focusing more on extractive summaries, trying to achieve more coherent and meaningful summaries. During a decade, several extractive approaches have been developed for automatic summary generation that implements a number of machine learning and optimization techniques [3]. In our project we used Python - Gensim library to make text summarization part. Details will be explained in Experiments section.

III. RELATED WORK

Before starting our project, we first searched the literature and searched for the previous studies in this field. We could not observe a study that matches the same way we thought, but we came across several studies that would guide us. In the articles we encountered in general, we observed that the opposite of the method we want to apply was tried. In other words, we observed that studies on the use of question answering techniques to implement the text summarization process more efficiently. For example, the study (Arumae, Kristjan, and Fei Liu. "Reinforced extractive summarization with question-focused rewards.", 2018), they used reinforcement learning to explore the space of possible extractive summaries and introduce a question-focused reward function to promote concise, fluent, and informative summaries [4]. The same research group approached this problem from a different perspective and presented a new study the following year. In this study (Arumae, Kristjan, and Fei Liu. "Guiding extractive summarization with question-answering rewards.", 2019), they thought that quality summaries should serve as a document surrogate to answer important questions, and such question-answer pairs can be conveniently obtained from human abstracts. The system learns to promote summaries that are informative, fluent, and perform competitively on question-answering [5].

We made several observations on this subject, as it consists of the text summarization part of our research and half of our project. One of them is this study that explains how to improve the summarization technique (Rahman, Nazreena, and Bhogeswar Borah. "Improvement of query-based text summarization using word sense disambiguation." Complex & Intelligent Systems (2019)), where the system finds semantic relatedness score between query and input text document for extracting sentences. The drawback with current methods is that while finding semantic relatedness between input text and query, in general they do not consider the sense of the words present in the input text sentences and the query. However, particular method can enhance the summary quality as it finds the correct sense of each word of a sentence with respect to the context of the sentence in their research [6]. This allowed us to have an idea that the text and questions in the dataset that we will use while training our model can be used more efficiently. Because our main starting point was to try the contribution of the text summarization method to information-retrieval based question answering, but we learned thanks to the article that we can also improve our text summarization method in the training phase of the model we train to perform the information retrieval in the stages we will use. In other words, since the dataset we will use for information retrieval contains question-answer and texts, we can use the word sense disambiguation technique described in this article while doing text summarization.

This work (Balage Filho, Pedro Paulo, et al. "Using a Text Summarization System for Monolingual Question Answering." CLEF, 2006.) is the most similar work to the subject of work in our project. They differ in their work using monolingual question answering at CLEF 2006 and topic-oriented summaries, also used GistSumm as the summarizer method. They aimed at assessing its accuracy in finding answers to the posted questions, which were used as the topics for producing the corresponding summaries [7].

IV. EXPERIMENTS

In this section, we will talk about the design of the system, methods and algorithms applied and evaluation of our project.

A. System

Gensim is a module that automatically summarizes the given text, by extracting one or more important sentences from the text. In a similar way, it can also extract keywords. This summarizer is based on the , from an "TextRank" algorithm by Mihalcea et al [10]. However, summarization only works for English for now, because the text is pre-processed so that stopwords are removed and the words are stemmed, and these processes are language dependent. In our project we use Gensim to summarize text of dataset [11].

Simple Transformers library is based on the Transformers library by HuggingFace and lets you quickly train and evaluate Transformer models. It is basically consisting of three stages : initialize a model, train the model, and evaluate a model. Also, it is currently providing and supporting Sequence Classification, Token Classification (NER) and Question Answering. Transformers (formerly known as pytorch-transformers and pytorch-pretrained-bert) provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, CTRL...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between TensorFlow 2.0 and PyTorch [12]. In our project we use Simple Transformers to train and evaluate a model.

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. SQuAD 2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD 2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering. SQuAD 1.1, previous version of the SQuAD dataset, contains 100,000+ question-answer pairs on 500+ articles [13].

B. Method

The approach or method we follow in our project basically consists of two main stages. We had to train a model to develop a question-answering system so we completed this first phase using the SQuAD dataset and the Simple Transformers library in progress of project. Then, in order to examine whether the text summarization process makes any contribution to our model and what effect it has, we have summarized the texts used in the SQuAD dataset through Gensim. Finally, we re-train our model for the version using the summary of texts.

After doing all these, we have two trained models. In order to observe the effect of using text summarization on the question answering system, we evaluate these two models with the test data we have to separate before. But since we use SQuAD in this project, it has its own evaluation script, and we used that script when we evaluated. We used only one dataset which is SQuAD in this project, but the amount of used dataset should be increased for efficient results. Also, if there any dataset that include context – questions and summarized context – questions would be allow you to get better results.

C. Algorithm

First, we are summarizing the Squad dataset. Our code is loop through all of the contexts and its questions. Because after the summarization, questions' answer may not be in the summarized context. So, we need to check and eliminate those kinds of questions. Eliminates means that, change is_impossible field to True if it's not in the summarized context, if it's in the summarized context, change answer_start field with the index of the first character of that answer in the summarized context. Because QuestionAnsweringModel take into account those kinds of data. Then we are giving output to model. As we mentioned before in the System section, all of the training, evaluation done by transformers and features that comes from BERT. For evaluation, we did not split the training data. Because Squad team has a dataset for the evaluation. Therefore, our code evaluates the model and print out the results.

D. Evaluation

After summarization process of dataset texts with Gensim, there was a decrease in the number of total texts we had.

{

Successful Summary Count: 9315

Total Context Count: 19035

}

This is because Gensim passes directly short texts without summarizing them. We left the short texts in this case as they were without making their summary in our project. For this reason, the number of texts we train (with and without summary) remains the same for both models.

While evaluating the models we trained, we used the Evaluation script provided by SQuAD 2.0.

After evaluating our first model which is trained with SQuAD 2.0 dataset, the results are as follows :

{

"exact": 69.36747241640697,

"f1": 72.04296013880722,

"total": 11873,

"HasAns_exact": 64.6255060728745,

"HasAns_f1": 69.98415413766132,

"HasAns_total": 5928,

"NoAns_exact": 74.09587888982338,

"NoAns_f1": 74.09587888982338,

"NoAns_total": 5945

}

After evaluating our first model which is trained with summary of SQuAD 2.0 dataset, the results are as follows :


```
{
  "exact": 70.8582498104944,
  "f1": 73.74096871900701,
  "total": 11873,
  "HasAns_exact": 64.67611336032388,
  "HasAns_f1": 70.44981808380041,
  "HasAns_total": 5928,
  "NoAns_exact": 77.02270815811606,
  "NoAns_f1": 77.02270815811606,
  "NoAns_total": 5945
}
```


exact, means that precision value of results of answers for given questions exactly same as given questions' result. Total, total asked question, HasAns_total: Total number of questions that have an answer in the validation dataset, NoAns_total: Total number of questions that have not an answer in the validation dataset.

V. RESULTS & DISCUSSION

One of the problems we encounter in this project is not to find a dataset in the format suitable for the Simple Transformers model we have used. This model keeps the JSON format used by the SQuAD dataset as a format in its own dictionary. Because of this, even if we found a dataset on the internet that contains question-answer and text to train model, actually we could not train because the JSON format is not compatible. In the final stages of our project, we found a GitHub work that can convert it to the SQuAD JSON format we want. This Dataset Converter for natural language processing tasks such question answering tasks, from one format to other one [8]. Although the dataset number it covered was 15, there were still some problems. All of the datasets it involved did not contain text and question-and-answer related to this text, most of that contains only question and answer so there was no text to summarize. Only 2 of these 15 dataset sets had the features we wanted. These are QAngaroo and NewsQA. In another study we have found, it is a reformer that converts the dataset we provide in excel format to squad format. Each 'title' can have multiple 'context'; each 'context' can have multiple 'question'; each 'question' can have multiple 'text' (answer) [9]. We could import the datasets that we found in other JSON formats to Excel and then use this study, but we had to bypass this study due to lack of efforts and labor.

Another problem we encounter is the process of train the model which planned to train with the text summarization. The confusing point here is that the dataset, because in that original dataset most of the question's answer could be find in text. However, when we summarize that text, answer of that question might be lost in that summary of the text because it is short and compressed version of original text. In that way, new trained model with summarized data knows the answer of question but actually in that summary of the text does not contain any match about answer due to the shrinking the size of original text. But situations like this are the problem that can occur in any system with text summarization, so a summary of a text can never contain as much content as the original text. In addition, as we mentioned in the related works section, some of these situations can be overcome with the semantic relationship score to be created between the question and the input text. In our project, we tried to overcome this situation by checking whether the answers were contained in the summarized text. If it contains, change the answer start index with looked possible answer index. Thus, without changing the content of the answers to the questions required for the model we will train, we have just changed the index from which the answer starts in given text. So as a result, we use the answers to the questions of the original dataset, which we do not use summarizing, we just change the index to indicate where the answers to these questions start on the text while training the model which using the summarization.

Before starting our project, we thought that one of the positive aspects of using text summarization is the increase in the number of contents without changing the dataset size used. In other words, for example if we consider that we have a 5 GB data area limit and we can fit 1 million documents into this field, we might fit the summary of 2 million documents in the same size when we do this with the text summarization version. We thought that this would be beneficial in terms of recognizing the diversity of the model to be trained. We should also state that we did not find so many datasets in this project, we tried to create a model with limited data in a limited time. In a longer period of time, we can catch the variety we have mentioned by finding a dataset in a suitable format. We think that the effects of the use of text summarization on the result will be more clearly understood and positive when the mentioned conditions are met to achieve this. Although we used limited data according to the results we obtained after running the evaluation script of SQuAD in the Evaluation section, a slight improvement was observed. We think that these results are a kind of inspiration for the future planned studies. Because, we think that the difference between these two models will be more evident when we bring the variety and dataset of the number of summaries produced to a suitable training format. On the other hand, the increase in the dataset amount will also increase the success rate of the model trained without using the summary we have compared, the difference between these two will not always be in a linear form.

At the beginning of the project, we wanted to test the following: Is there a difference between a Question Answering System using text summarization and the Question Answering System that using to traditional rules? How does Text Summarization affect the Information Retrieval process? Obviously, at the beginning of the project, we thought that text summarization would not have much effect and would lower the success rate even more. Because the articles we investigated (Balage Filho, Pedro Paulo, et al. "Using a Text Summarization System for Monolingual Question Answering." CLEF, 2006.) said the results were poor and when a text was summarized, we thought that little details and the integrity of meaning would deteriorate, so it would not be very productive. But today, the studies in the field of text mining and NLP are progressing very fast and showing great improvement. Perhaps due to the good technology we use, we did not get results like the results mentioned in another article. In this study, we got the answers to the questions we wondered before starting our project, and we came to the conclusion that successful results can be obtained if this issue is further concentrated and appropriate conditions are provided.

VI. CONCLUSION

In this work, we introduce relationship between two main topic of NLP which are IR Based Question Answering and Text Summarization. The main purpose of this study is to test what effect it will be when we train the model used when performing Information Retrieval in the Question Answering System using Text Summarization. In this process, both the model that follows with the normal traditional rules and the model using the text summaries we want to test were compared with each other. When we compared the results, it was observed that the model we trained using the text summary was more successful than the other model. It is possible to conclude that using text summarization in question answering system will have a positive effect. But for future studies, it is worth noting that the enrichment of dataset and the increase of content will not increase too much the difference between the results. As a result of our study, we proved that two topics such as text summarization and question answering can work in harmony with each other. We hope this will give new ideas for other future studies.

REFERENCES

[1] Mitra, Bhaskar, and Nick Craswell. "Neural models for information retrieval." arXiv preprint arXiv:1705.01509 (2017).J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Babar, Samrat & Tech-Cse, M & Rit,. (2013). Text Summarization:An Overview.

[3] Gambhir, Mahak, and Vishal Gupta. "Recent automatic text summarization techniques: a survey." Artificial Intelligence Review

[4] Arumae, Kristjan, and Fei Liu. "Reinforced extractive summarization with question-focused rewards." arXiv preprint arXiv:1805.10392 (2018).

[5] Arumae, Kristjan, and Fei Liu. "Guiding extractive summarization with question-answering rewards." arXiv preprint arXiv:1904.02321 (2019).

[6] Rahman, Nazreena, and Bhogeswar Borah. "Improvement of query-based text summarization using word sense disambiguation." Complex & Intelligent Systems (2019): 1-11.

[7] Balage Filho, Pedro Paulo, et al. "Using a Text Summarization System for Monolingual Question Answering." CLEF (Working Notes). 2006.

[8] Dataset Converter for Question-Answering (QA) Tasks

[9] Domain-specific Excel dataset to SQuADv1.1 JSON format using Python.

[10] Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

[11] Gensim - Text Summarization

[12] Hugging Face - Transformers

[13] SQuAD