Query oriented multi-document summarization with training on wikipedia data

Charles Sutton
Technion, Computer science
Haifa, Israel
sutton@cs.technion.ac.il

Kira Radinsky
Technion, Computer science
Haifa, Israel
kirar@cs.technion.ac.il

ABSTRACT

We introduce the first deep supervised learning framework able to summarize a corpus of documents given a query. The summary is performed with an extractive style, meaning our model identifies the most relevant combination of sentences existing in the corpus of document to create a summary. The key component of the framework is the *completion score*, that measures how much a sentence contains information relevant to the summary and not redundant with other sentences already in the summary. During the training stage, we train the completion model on data extracted from the Wikipedia knowledge base. Finally, we different configuration of the framework against the new TD-QFS dataset, more relevant to this task than standard DUC datasets. Our experiments show promising results for further experiment using deep neural network for this summarization task.

Keywords

Query oriented summarization, wikipedia, framework, deep neural networks, external data.

Acknowledgments

This report presents researches carried out for the course *CS236601* from the Technion (Haifa, IsraÃńl). It deals with Deep learning for natural language processing. It has been made under the great supervision of Kira Radinsky and I am thankful for her guidance and all the inspiring discussions we had along this project.

1. INTRODUCTION

Search engines is the most common tool used nowadays to retrieve the relevant information among a massive amount data. Due to the multiplication of online resources, the amount of information related to a query is most-often distributed in multiple documents.

The rise of the artificial intelligence assistance in searches makes this issue more complex since query are more and more sophisticated and, even if powerful search engines are able to retrieve relevant documents efficiently they are not designed to extract the core information over different documents when they are given a specific query. Summarizing data gathered among a corpus of documents is by design the solution to this issue.

In this paper we focus on query-oriented multi-document summarization. Summarizing information is a difficult process involving multiple human abilities in analyzing and synthesizing redundant information. Major contributions to the field of summarization has come along tasks of the document understanding conferences of the mid-2000s (DUC 2002-2006). Best models at these time were based on heuristic, statistical and graph based methods.

Due to the subjectivity and the high-level of abstractness of this task, there is a natural issue regarding the evaluation of a model and this partly explains why this field has been left behind compared to other tasks. Current metrics are quantitative and are related to statistics of co-occurrences between a candidate summary and a gold standard one. They are nevertheless unable to fully measure how much a model has captured and summarized the information.

Another issue concerns the few quantity of training data relative to this task. It has therefore restricted most of the research around to unsupervised learning and fine-tuned approaches, and this paper contribute to open the way for supervised learning approaches.

Recent developments around the deep neural networks have proved significant advances in many NLP tasks and show that there is also a room of improvement for generic summarization [8, 10]. Regarding query oriented summarization only a single deep learning model [7] has been released. Our paper proposes a new method with a new paradigm of supervised training involving external data from Wikipedia.

2. BACKGROUND AND RELATED WORK

2.1 Tasks relative to summarization

Summarization is a wide field and each case of summarization is defined by three factors : the distribution of the corpus, the orientation of the summary and the style of summarization.

When dealing with a task of summarization one has to consider if the content to summarize is a single text document or distributed over a corpus of documents. In the case of a single document summarizing consists in compressing the size of the document while keeping as much information as possible. When it is distributed over a corpus, the task is more complex since summarizing has also to deal with information redundancy and even chronology.

It is also important to decide whether the summary is oriented or not. Generic summarization is when a user may wants to capture the main information contained in the corpus, by opposition query-oriented summarization consists in summarizing the text and retrieve specific information given a query. Queries may refer to a sequence of keywords or to a question sentence. In this project we considered queries of two or three words.
Finally, the last point to consider is the style of summarization. A model can summarize in the extractive style, meaning the model extracts from the corpus the most relevant combination a sentences, given a length constraint. By opposition, abstractive summarization style creates the summary with new words and a new style, similarly to what a human would do.
Regarding the most common performance metrics, most of the best models adopt the extractive style. This is partly due to the complexity involved in creating an abstractive style model.

2.2 Datasets

Not much data are available for the query oriented summarization task and up to our knowledge, there isn't any framework used to train summarizers in a supervised fashion.
To measure the performance of a summarizers, models are evaluated against two standard corpora from the document understanding conference (DUC). DUC datasets have been introduced in 2005 and 2006, when query-oriented summarization[1] were the object of multiple tasks. DUC corpora contain 50 clusters of English newspaper documents, and for each cluster a single query is provided along with a topic description and corresponding newspaper articles [3]. For each cluster of document, there are gold standard summaries written by four human experts among a pool of ten. These summaries are written in the abstractive style and are limited to 250 words.
In 2016, [1] claim the DUC 2005/2006 corpora are not enough topic diversified, and they proposed a new topic diversified query-focused summarization corpora (TD-QFS). They prove their assumption with a measure of the topic concentration among each cluster and show that a single topic is covered within each cluster of DUC corpora, whereas the more topics are found in clusters of the TD-QFS. They also show that generic summarizers are able to compete with query-oriented ones on DUC corpora, whereas the query orientation has an edge on (TD-QFS).
The TD-QFS corpus aggregate documents related to medical diseases and come from various sources (Wikipedia, WebMD ...). Each cluster of document deals with multiple topics at the same time. There are four large clusters of documents, each focusing on a disease among alzheimer, asthma, cancer, obesity. The gold standard summaries are produced by medical experts.
In this paper, we will only measure the performances of our model on the TD-QFS dataset,

2.3 Related models

[1]for the sake of brevity query-oriented summarization refers query-oriented multi-document summarization in the following.

All along our research we compare our model with Biased Lexrank[9] the best existing model up to our knowledge. Biased Lexrank (BL) is the query-oriented extension of the multi-document summarizer Lexrank [4] :
BL represents the corpus of documents as a graph, where each node represents a sentence and each vertex denotes similarity between two nodes. Once the graph is built, BL attribute to each node a score coming from the computation of the stationary distribution and the query relatedness. The summary is then written in the extractive way according to the score.
We have to mention the Query-oriented deep extraction model (QODE) [7], which is the single deep approach tackling query-oriented summarization we have found in the literature. QODE is a biomimetic neural framework unifying three steps : concepts extraction, summary generation, and reconstruction validation.
There isn't any official implementation available and since the paper is hard to reproduce, this model won't be taken into account when comparing summarization performances.
Interestingly, above models learn in an unsupervised fashion, meaning they don't train on examples of summarization but only use the text data contained in the corpus to produce summaries. By opposition, our model learns on training data and it is the first deep approach tackling query-oriented summarization in a supervised fashion.

3. MODEL DESIGN

This section fully describes our new framework for the query-oriented summarization task. We first introduce the completion score which is the key part in our supervised model. Next, we explain how to train it on data extracted from Wikipedia.

3.1 The completion score

When performing an extractive query oriented summary, one has to choose the most relevant set of sentences according to a query, and a cluster of document. Ideally the output is a summary that first gathers all the query-related information contained in the original text. The ideal summary also avoids redundant information by minimizing the "information overlap" among the selected sentences. Finding the best combination has a combinatorial complexity, that's why we have considered a solution, iterative by design, to construct our summary.
To do so, we introduce the completion score. This score applied to a sentence, and measures how much the sentence brings information to an existing partial summary, while avoiding redundancy with sentences already in the summary. Formally, the completion score comes from the combination of four inputs :

$$CS : (s, q, p, t) \rightarrow y \in [0, 1] \quad (1)$$

where s denotes the sentence to score, q denotes the query, p denotes the partial summary composed of sentences from the original cluster of document, and t denotes all the text contained in the cluster of document.
In practice, we use a pre-trained doc2vec model as the text embedder of all the components mentioned above. Our deep

model learns the completion score of any sentence, given a known triplet : query, partial summary and text.

3.2 Wikipedia external data

In order to train our model, we use the Wikipedia knowledge base which presents natively a structure corresponding the completion score presented above.
The Wikipedia presents two main advantages in our framework :

- There is a huge amount of data distributed on a very wide range of topics and therefore our framework is very flexible to the topics.
- It is easy to get and structure wikipedia data using an API [2]. And therefore, our model can be totally autonomous.

In practice, we have selected the article corresponding to the four clusters of the TD-QFS dataset : Alzheimer, Asthma, Cancer and Obesity and then we have selected all related articles [3] up to a given degree, in our case we limited our research to the order 2.

3.3 Training stage

To build our training data set, we use the natural structure of wikipedia, formally :
The query is chosen at the subsection level of an article, and is represented by the concatenation of the title, the section and the subsection of an article. It is important to keep the title and the corresponding section to stick to "search style" queries for instance : "alzheimer diagnosis technique".
The partial summary corresponds to the selection of randomly chosen sentences among the subsection sentences. The size of the partial summary is also randomly chosen.
The text is chosen to be or the the text contained in the wikipedia article or also the text contained in related articles up to a given order.
To finish, we create positive examples by selecting randomly a sentence in the subsection that is not already in the partial summary, and we create negative examples by selecting a random sentence outside the subsection (see table 1).
Considering the huge quantity of wikipedia articles and the combinatorial aspect of the training set constitution, there is a quasi-infinite number of possible triplets query - partial summary - article.
For the sake of generalization and to take advantage of the (almost) infinite quantity of training samples, we always feed our model with new simulated triplets during the training stage. Nevertheless this new paradigm of training presents the drawback that we can't monitor the evolution of the validation loss along the epochs, since in our case training samples are simulated and always renewed.
Finalyl, to train our model, we use the binary cross-entropy function :

$$e = \frac{1}{n}\sum_{i=1}^{n} y_i \cdot log(\hat{y_i}) \quad (2)$$

[2]We acquired data through the Python wikipedia package : https://pypi.python.org/pypi/wikipedia/
[3]Related articles are found thanks for hyper-references inside the article

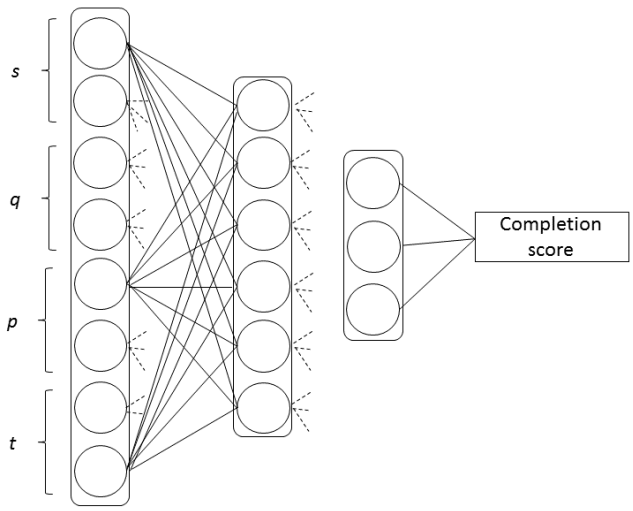

Figure 1: Schema of the fully connected model to estimate the completion score. In practice, each input has size 400, and we use a decreasing factor of 10 between layers. Therefore, layers have respectively a size of 1600, 160 , 16 and 1.

Where e denotes the average loss.

The deep neural aspect of our approch is the estimation of the completion score. We have chosen to use a dense architecture by stacking three fully connected layers with decreasing size.
The input layer of the model is the concatenation of the embedding of the sentence, the query, the partial summary and the text.
To represent short and long text data, We have chosen to use Paragraph vector, with pre-trained representation described in [5]. We also attempted to build more specific embedder regarding the size and the topic of data to represent, but it required to much time to be used efficiently.

4. EXPERIMENTS AND RESULTS

4.1 Summarization procedure

The completion score represents itself the trade-off between quantity of information and absence of redundancy. It is therefore iterative by design to build a summary 1.
We have decided to adopt this greedy approach since finding the best combination of sentence is a combinatorial complex problem.

Other approaches using sentence scoring uses rerankers such as MMR [2]. This is mainly due to the fact their scores identify which sentences contains a lot of information, but they don't avoid redundancy. Since the completing score avoids redundancy by design, we choose not to use reranking methods.

4.2 ROUGE metrics

ROUGE metrics [6] are the standard in evaluating the performances of summarizers. To stick to the standard, we mea-

Query	Partial Summary	Sentence	label
Asthma causes genetics	Family history is a risk factor for asthma, with many different genes being implicated. If one identical twin is affected, the probability of the other having the disease is approximately 25%. Many of these genes are related to the immune system or modulating inflammation. Even among this list of genes supported by highly replicated studies, results have not been consistent among all populations tested. Some genetic variants may only cause asthma when they are combined with specific environmental exposures.	Risk for asthma, then, is determined by both a person's genetics and the level of endotoxin exposure.	positive
		Asthma is the result of chronic inflammation of the conducting zone of the airways (most especially the bronchi and bronchioles), which subsequently results in increased contractability of the surrounding smooth muscles.	negative

Table 1: This table illustates how we create training data. These examples are taken from the *Asthma* Wikipedia article, with the query path Asthma causes genetics. Here, the partial summary is formed from four sentences, chosen randomly inside the subsection genetics. The first candidate sentence is also from the subsection genetics that's why it is labeled 1 (positive example), the second sentence come from another subsection and is therefore labeled 0 (negative example).

Data: Let q denotes a query, and t the corpus of documents, p the partial summary.
Result: S the summary
initialization : p and S are empty ;
while *word limit isn't reached* **do**
  get the sentence in d but not in p with the highest completion score;
  add this sentence to p;
  if *the word limit is reached* **then**
    return *S*
  else
    add the sentence to S;
  end
end

Algorithm 1: Summarization procedure

sure the performance of our models against the ROUGE-2 and ROUGE-SU4 .
ROUGE-2 is a recall related measure of the co-occurrences between a candidate summary and gold standard summaries. ROUGE-SU4 is similar to ROUGE-2 but introduces the relaxation of allowing as many as four words between the two words of a bi-gram.

4.3 Settings

Our framework allows great flexibility of configurations :

Data Wikipedia is very wide and let you access to a very wide range of topic. You can choose to take random articles or specific topics, eventually extracted from the cluster you want to summarize, to train your text embedder or to build you training set.

Text embedder You can choose different separate embedder for the query, and longer text. You can focus the training of your embedder around specific topics or in a general way.

Query depth Every Wikipedia article is organized as tree with a depth up to three. Since content of subsections and sections is highly related, you can choose to limit the "query depth" to the section level.

Model	ROUGE-2	ROUGE-SU4
Biased Lexrank	0.17	0.2
Pargraph Vector + Fully Connected + Random Wikipedia articles	0.102	0.130
Pargraph Vector + Fully Connected + Wikipedia articles up to order 2	0.103	0.133
LSTM + Fully Connected + Wikipedia articles up to order 2	0.10614	0.13561

Table 2: Summary table. Results of our framework are correct but still far from the best current model in query oriented summarization. This is mainly due to the lack of time to perform the heavy computations necessary by our model and experiments. There is a room of improvement for each step of the framework. We see in these experiments that focusing the training on related Wikipedia improves slightly the results. We also see that a LSTM trained on specific data improves also the results of our supervised framework.

In our experiments, we try multiple configurations, summarized in Table 2

Our experiments don't show better results than BL, but we recall that there is a room of improvement at many steps of the framework :
A huge improvement in future research will be to improve how the text is embedded in the framework. We will try to train recurrent models (RNN, GRU, LSTM ...) as a language model, and at different scale : query, corpus of document to summarize, Wikipedia data, wider text data. During our experiments recurrent models weren't trained enough to produce a relevant text embedding.
There are a lot of configuration to experiment but we lacked

of time to test it all : We performed experiments on a virtual machine from Amazon Web Services. We have chosen the g2.2xlarge GPU machine, and each experiment lasts approximatively 4 hours.

5. CONCLUSION

This paper proposes the first framework with a deep supervised approach to tackle query-oriented summarization. We use external data from Wikipedia to train the completion score, which is the key contribution of our project.
Further research consists in exploring the various improvements aformentionned, starting with a separate text embedder for query and text. There is also work in the creation of the dataset, by choosing a shorter query depth, and a better loss function.
Finally, the ultimate improvement would be to find which configuration suits the most to deploy automate such framework to a general corpus of document to summarize.

6. REFERENCES

[1] T. Baumel, R. Cohen, and M. Elhadad. Topic concentration in query focused summarization datasets. In *AAAI*, pages 2573–2579, 2016.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[3] H. T. Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.

[4] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[5] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[6] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[7] Y. Liu, S.-h. Zhong, and W. Li. Query-oriented multi-document summarization via unsupervised deep learning. In *AAAI*, 2012.

[8] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[9] J. Otterbacher, G. Erkan, and D. R. Radev. Biased lexrank: Passage retrieval using random walks with question-based priors. *Information Processing & Management*, 45(1):42–54, 2009.

[10] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.