HIGH-COMPRESSION TEXT SUMMARIZATION

A PREPRINT

Yizhao He
Dalhousie University
Nova Scotia, Canada
james.he@dal.ca

Evangelos Milios
Dalhousie University
Nova Scotia, Canada
eem@cs.dal.ca

July 8, 2024

ABSTRACT

We proposed a new high-compression text summarization model for very short and fixed-length text summarization. Unlike most text summarization models that require training or fine-tuning to generate reasonable domain-specific summaries, our proposed model does not require any training and fine-tuning. The high-compression text summarization model includes two stages. In the first stage, the "Sentence Extractor" selects salient sentences from long documents, and in the second stage, the "Summarizer" summarizes the selected sentences into a very short summary.

We also investigated and compared various commonly used text summarization data-set, and proposed a new data-set for very short text summarization. The experimental results shows that the model achieved high scores in terms of ROUGE-1 and ROUGE-L metrics. In addition, we have established a website to directly interact with the model to further explore its potential.

Based on our knowledge, this is the first work to generate high compression text summarization, which can be further developed for the application of file name generation, email subject generation and paper title suggestion.

***K*eywords** High-compression · Text Summarization · Extractive Summarization · Deep Learning · Bert · Clustering · Information Retrieval

1 Introduction

In the past few decades, the number of text on the Internet is growing exponentially. The need to be able to understand the overall meaning without reading the entire document is becoming ever greater. Text summarization is the task to generate short summaries from long documents.

Text summarization can be used in various applications, such as search engine, and title generation, etc. The short summary can be used as a preview of a long website to improve search efficiency[1]. News websites can use text summarization models to generate brief descriptions of news topics as headlines[2].

Recently, researchers have proposed various model and achieved great results in the task of text summarization. However, many approaches requires several hours or even several weeks of manual tuning to produce meaningful results in specific domains. Therefore, we proposed a new high-compression text summarization model for very short and fixed-length text summarization(1-4 words). Unlike most text summarization models that require training or fine-tuning to generate reasonable domain-specific summaries, our proposed model does not require any training or fine-tuning, and it can still generate reasonable general summaries.

Skipping the process of fine-tinning would make the model more difficult to generate domain-specific result. However, the users have the option to upload domain-specific corpus to let the model automatically adapt to it.

Besides, most text summarization models generate a complete sentence as the summary, but the generated summary is usually more than 10 word tokens, and they lack the control of the number of words in the sentence. However, our

proposed model focus on generating very short length summaries(1-4 words), and it have fully control over the number of words in the generated summary.

The high-compression text summarization model includes two stages. In the first stage, the "Sentence Extractor" selects salient sentences from long documents, and in the second stage, the "Summarizer" summarizes the selected sentences. We also investigated and compared various commonly used text summarization data-set, and proposed a new data-set for very short text summarization. We will describe the summarization model in more detail in the Mythology section.

The experiments runs on Wikipedia pages, and the model takes the Wikipedia pages in and generate given number of words as the output summary. We compared n-word summaries respectively with target Wikipedia titles.

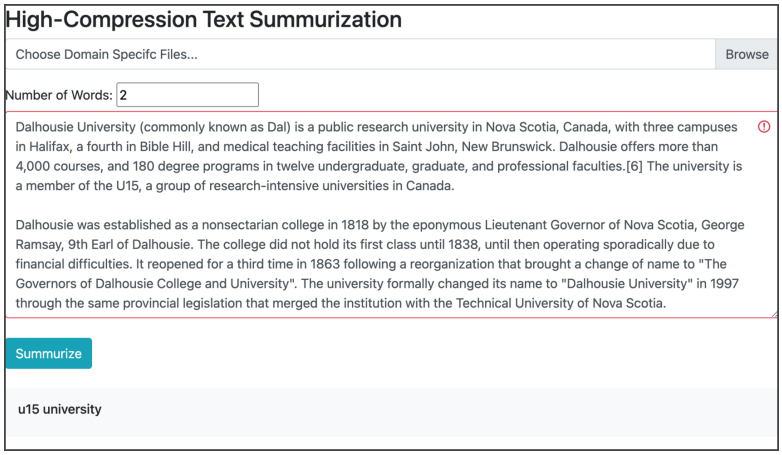

Figure 1: High-Compression Text Summarization User Interface

In addition, we have established a website to directly interact with the model to further explore its potential. As the Figure 1 shows, users have the option to upload domain-specific files to fit the model, making it more likely to generate domain-specific summaries. Users can simply put the desired number of words in the summary and the original document into associated text box, then press the "Summarize" button, the generated summary will appear in the bottom.

Based on our knowledge, this is the first work to generate high compression text summarization, which can be further developed for the application of file name generation, email subject generation and paper title suggestion.

2 Background/Related Work

This section organized related works in Text Summarization, including state-of-arts text summarizing models, text summarization automatic metrics, and text summarization data-set. In addition, we will emphasis the uniqueness and motivation behind the high-compression text summarization model.

2.1 Evaluation Metric: ROUGE-i

There are two commonly used automatic metrics for evaluating text summarization, Automatic metrics Recall-Oriented Understudy for Gisting Evaluation(ROUGE)[3] and ROUGE-L[4], even if they both have limitations on evaluating quality aspect(grammaticality, fluency, coherence, etc). ROUGE-i[3] measures the overlap of i-grams[5] between the reference summaries and the result, while ROUGE-L[4], evaluate the sentence-level structure similarity based on longest common sub-sequence. In the case of ROUGE-1, it refers to the overlap of unigram between the output and target. In the case of ROUGE-1, it refers to the overlap of bigrams between between the output and target. In this paper, we will use ROUGE-1 and ROUGE-L as the evaluation metrics.

2.2 Text Summarization Data-Set

There are various of commonly used data-sets for training and evaluating text summarization, such as CNN/Daily Mail[6], Gigaword[7], and X-Sum[8].

CNN/Daily Mail was firstly collected and introduced by Nallapati in 2016 [6]. The CNN/Daily Mail data-set contains in total 312,084 pairs of online news articles and paired multi-sentence summaries(287,226 training pairs, 13,368 validation pairs, and 11,490 test pairs). In average, each article contains 3.74 sentences with 781 tokens, and each summary contains 3.74 sentences wit 56 tokens[9].

The Extreme Summarization data-set (X-Sum) was collected and introduced by Narayan et al., in 2018[8]. The X-Sum data-set was collected from online article and summary from the BBC. The X-Sum data-set is mainly used to create short and one-sentence news summary, and it contains 204,045 training pairs, 11,332 for the validation pairs, and 11,334 for the test. In average, each article contains 20 sentences with 431 tokens, and each summary contains 23 token[8].

The Gigaword data-set was collected and introduced by Rush et al in 2015[7]. The Gigaword data-set contains 3.8 million training pairs, 189 thousand development pairs, and 1951 test pairs. In average, each article contains 31.4 tokens, and each summary contains 8.3 tokens[7].

2.3 Text Summarization Models

Researchers have proposed various model and achieved great results in the task of text summarization. However, many approaches requires several hours or even several weeks of manual tuning to produce meaningful results.

Most of leading performance models are descendants of BERT(Bidirectional Encoder Representations from Transformers), but they all comes with various different implementation and optimization algorithms. Different from other language representation models, BERT is designed to pre-train deep bidirectional language representations from unlabeled text by jointly conditioning on the entire context in all layers[10]. Therefore, the pre-trained BERT model is usually fine-tuned with one additional output layer to create state-of-the-art models for other tasks, such as text summarization.

Daily Mail Data Set[9]			
Model Name	ROUGE-1	ROUGE-2	ROUGE-L
BertSumExt[11]	43.85	20.34	39.90
BERT-ext+RL[12]	42.76	19.87	39.11
PNBERT[13]	42.69	19.60	38.85
HIBERT[14]	42.37	19.95	38.83
Latent[15]	41.05	18.77	37.54

Table 1: Models comparison on Daily Mail[9] data-set.

BertSumTxt[11], PNBERT[13], BERT-ext+RL[12], and HIBERT[14] are all descendants of BERT[10] with various detail implementation and optimization algorithms. They all achieved great results in the task of text summarization, where BertSumTxt [11] introduced a document-level encoder and achieved 43.85 in ROUGE-1, 20.34 in ROUGE-2, and 39.90 in ROUGE-L in the CNN/Daily Mail[9] test data-set as shown in Table 1. Zhang[15] proposed a latent variable extractive model in 2018, and and achieved 41.05 in ROUGE-1, 18.77 in ROUGE-2, and 37.54 in ROUGE-L in the CNN/Daily Mail[9] test data-set.

X-SUM Data Set[8]			
Model Name	ROUGE-1	ROUGE-2	ROUGE-L
PEGASUS[16]	47.21	24.56	39.25
BART[17]	45.14	22.27	37.25
BertSumExtAbs[11]	38.81	16.50	31.27

Table 2: Models comparison on X-SUM[8] data-set.

PEAGASUS[16] uses a sequence to sequence model to generate summaries, and proposed pre-training objectives tailored to reduce the lack of systematic evaluation across diverse domains. It achieved 47.21 score in ROUGE-1, 24.56 in ROUGE-2, and 39.25 in ROUGE-L for the X-SUM[8] data set. BART[17] also used the sequence to sequence model to generate summaries, but the model was trained to learns how to map corrupted documents to the original ones. As shown in Table 2, it achieved 38.81 score in ROUGE-1, 16.50 in ROUGE-2, and 31.27 in ROUGE-L for the X-SUM[8] data set.

Song and others[18] proposed a Transformer-Based framework to generate summaries with the possibilities to copy from the inputs, and achieved 39.08 score in ROUGE-1, 20.47 in ROUGE-2, and 36.69 in ROUGE-L for the Gigaword data-set[7] data set. While BiSET[20] presented a bi-directional selective encoding model to apply filters to avoid noisy

Gigaword Data Set[7]			
Model Name	ROUGE-1	ROUGE-2	ROUGE-L
ControlCopying[18]	39.08	20.47	36.69
UniLM[19]	38.90	20.05	36.00
PEGASUS[16]	39.12	19.86	36.24
BiSET[11]	39.11	19.78	36.87

Table 3: Models comparison on Gigaword[7] data-set.

training data, and achieved 39.11 score in ROUGE-1, 19.78 in ROUGE-2, and 36.87 in ROUGE-L for the Gigaword data-set[7] data set as shown in Table 3.

3 Methodology

As Figure 2 shown, the high-compression text summarization models consists of two major components: *Sentence Extractor*, and *Summarizer*. The purpose of the *Sentence Extractor* is to select salient sentences to represent the entire document, and each selected sentence should be semantically different from other selected sentences in order to represent the full text as comprehensively as possible. The purpose of the *Summarizer* is to summarizes the selected sentences into one very short summary, and the generated abstract should cover as much context of the original document as possible.

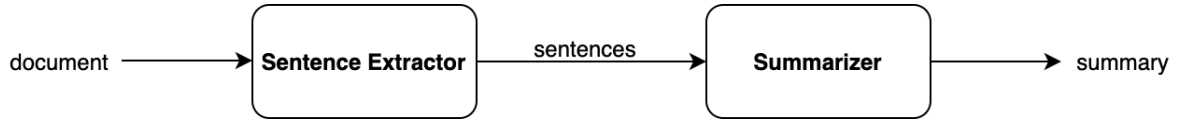

Figure 2: Overview: The Model Architecture

Given D documents and output summary length L_i for each document d_i (where each document d_i consists of S_i sentences, and $1 \leq i \leq D$). For each document d_i, the *Sentence Extractor* groups S_i sentences into L_i groups where each group of sentences have similar semantic meanings, then select the most representable sentence from each group and pass them into the *Summarizer*. After that, the *Summarizer* summarize selected sentences into a summary of L_i length.

3.1 Sentence Extractor

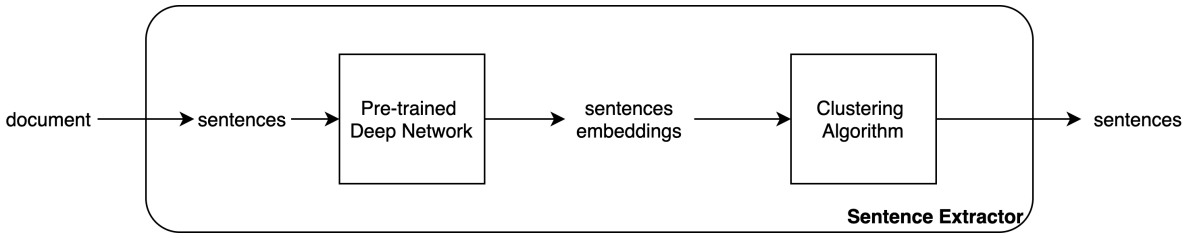

Figure 3: Overview: The Model Architecture

The purpose of the sentence extractor is to select salient sentences to represent the entire document, and each selected sentence should be semantically different from other selected sentences in order to represent the full text as comprehensively as possible. As shown in figure 3, a document was firstly split into sentences, and then passed into a pre-trained deep network to obtain representations for each sentence. These sentences representations can also be called sentence embeddings, and they can best represent the semantics of each sentence. After that, we apply clustering algorithm to cluster sentences embeddings into L_i thematically different groups(where L_i is the number of words in the summary), then pick the sentence which is closest to the centroid for each cluster. These selected sentences will be more likely to represent the entire document, because the clustering algorithm group semantically similar sentences together. Selecting the sentence closest to the centroid allows us to obtain the most representative sentence from each thematically different cluster.

3.1.1 Pre-trained Deep Network

In this paper, we chosen a triplet network architecture[21] as the pre-trained deep network to get sentence embeddings, because the triplet network can learn useful thematic metrics and it was trained to embed sentences from the same section closer. It also shows that it outperforms state-of-the art multipurpose embeddings and semantic similarity methods on the task of thematic clustering of sentences [21]. In addition, Triplet network is designed to learn thematic similarity between sentences, and we can get high quality sentence embeddings, tailored to reveal thematic relations between sentences [21].

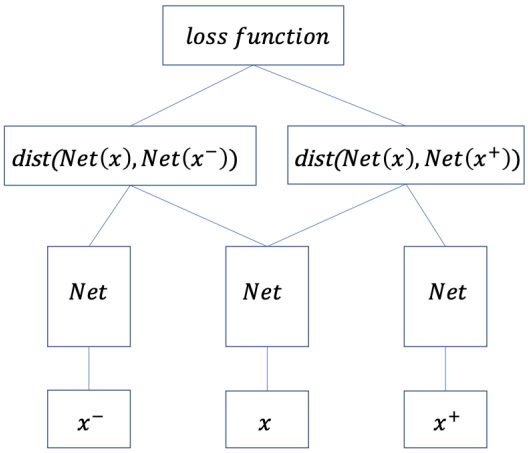

Figure 4: Triplet Network

As Figure 4 shows, the triplet network was trained using triplets (x, x^+, x^-) out of the 5.37M Wikipedia articles, where x and x^+ are sentences from the same section of an Wikipedia article, and x and x^- are sentences from different section of an Wikipedia article[21]. Each of x, x^+, x^- of each triplet was fed into the same network Net as a sequence of word embeddings (pre-trained glove word embeddings of 300d trained on 840B tokens[22]), where Net composed of a Bi-directional LSTM with hidden size 300 and 0.8 dropout followed by an attention [23] layer of size 200. After that, $L1$ distance loss function was used to denote the distance between those results, where d^+ is the $L1$ distance of $Net(x)$ and $Net(x^+)$, d^- is the $L1$ distance of $Net(x)$ and $Net(x^-)$[21].

$$d^+ = dist(Net(x), Net(x^+))$$

$$d^- = dist(Net(x), Net(x^-))$$

Then, it applied soft-max on $d+$ and $d-$ which is denoted as $p(d^+)$ and $p(d^-)$ respectively [21]. Finally, the total triplet-loss is equal to[21]:

$$loss = |p(d^+)| + |1 - p(d^-)|$$

Since the triplet network architecture was trained to embed sentences from the same section closer, the output embedding is more likely to capture each sentence's thematic meaning. Also experiments shows that it outperforms state-of-the art multipurpose embeddings and semantic similarity methods on the task of thematic clustering of sentences [21]. Therefore, we choose the triplet network as the pre-trained deep network model. Once a document was split into sentences, each sentence is feed into this triplet network and output sentence embeddings.

3.1.2 Clustering Algorithm

We chosen K-Mean as the clustering algorithm, since there is no major improvement found from other clustering algorithms, such as Spectral Co-Clustering algorithm[24] and Spectral bi-clustering[25]. Besides, K-Means is pretty fast and easy to implement.

After getting sentence embeddings, we apply K-Mean to cluster sentences embeddings into L_i clusters, then for each cluster we select the sentence closest to the centroid of the cluster, since the sentence which is closest to the centroid can best represent the thematic meaning of its belonging cluster. K-Mean clusters semantically similar sentences together, and each cluster is thematically different from other clusters, which makes the selected sentence more likely to represent the entire document.

3.2 Summarizer

The purpose of the *Summarizer* is to summarizes the selected sentences into one very short summary, and the generated abstract should cover as much context of the original document as possible. The Summarizer consisits of two altnative approaches, the *Extractive Summarizer* and the *Abstractive Summarizer*. The *Extractive Summarizer* extracts several words from the input sentences and combined to make a summary, it normally takes rank or given weights to the important words or section of the document, then select the words based on their level of importance. While, the *Abstractive Summarizer* take advantage of deep learning model to paraphrase the sentences. The involved neural network is built to understand the document semantically, then generate the summary based on the its understandings, as a result, the *Abstractive Summarizer* is more likely to generate human-like sentences. However, they usually needs to be fine-tuned in specific domains to perform properly, but in this paper we focus on the performance and ease of use, therefore, we will focus on the *Extractive Summarizer*.

In the following subsections, we will describe the *Extractive Summarizer* and in more detail.

3.3 Extractive Summarizer

The *Extractive Summarizer* utilize term frequency–inverse document frequency(tf-idf) to indicate how important a word is to the given documents. The tf-idf value is still one of the most effective term-weighting schemes due to its simplicity and efficiency[26], and it helps adjust the weigh for some words appear more frequently by increasing proportionally to the number of times a word appears in the given documents[26].

The tf–idf is the product of two term frequency(tf) and inverse document frequency(idf). There are five most popular weighting schemes to calculating *tf* as follows: binary, raw count, term frequency, log normalization, and double normalization K. And, there are four most popular weighting schemes to calculating *idf* as follows: unary, inverse document frequency, inverse document frequency smooth, inverse document frequency max. After the experiments, we select *log normalization* as the *tf* weighting scheme, and inverse document frequency smooth as the *idf* value.

$$tf(t,d) = log(1 + f_{t,d})$$

$$\mathrm{idf}(t, D) = \log(\frac{N}{1 + |\{d \in D : t \in d\}|}) + 1$$

where *N* is the total number of documents in the corpus $N = |D|$, and $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears.

Then tf–idf is calculated as

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

The calculation of *idf* is time and space expensive if the number of given documents is extremely large(more than 1 million document). In order to be able to handle millions of given documents, we invented an algorithm to pre-calculate *idf*, which dramatically decreased the time complexity and space complexity.

Algorithm 1 Pre-Calculate IDF

```
idfMap ← Hash Map
for document in documents do
    words ← tokenize, lemmatize, and lower each words
    for word in word do
        if idfMap contains word as key then
            increment the associated value of word by 1
        end if
        if idfMap not contains word as key then
            put word into idfMap with associated value 1
        end if
    end for
end for
for each word in idfMap do
    idfMap[word] ← log(N / (1+idfMap[word])) + 1
end for
Save idfMap with pickle
```

More details about Line 3: we used $nltk.tokenize.word_tokenize()$ method to extract the tokens from the document by using $tokenize.word_tokenize()$ method, and used $nltk.wordnet.WordNetLemmatizer()$ to group together the different inflected forms of a word as a single item to improve the space efficiency.

4 Results

Other text-summarization data-sets, such as CNN/Daily Mail[9], Gigaword[7], and X-Sum[8], do not fit our need to evaluate the high-compression text summarization, since its target summaries are relatively long. However, Wikipedia pages normally have a long document, but short title, which fits the need of high-compression text summarization.

Therefore, we randomly select pages from Wikipedia as the test data set, then compare the generated short summaries with its target Wikipedia title.

We randomly selected 2520 Wikipedia pages with titles of 1-2 words, and 3318 Wikipedia pages with titles of 3-4 words. Then, we used the ROUGE toolkit[27] for evaluation of the generated summaries in comparison to the title of target Wikipedia titles.

We selected three variants of this metric to conduct evaluation: ROUGE-1, ROUGE-2 and ROUGE-L (RL), which are computed by matching uni-grams, bi-grams and longest common sub-sequences respectively between the generated summaries and target titles.

4.1 Experiment: Wikipedia Pages with Titles of 1-2 Words

n word summary	Precision	Recall	F-1 Score
1 word Summary			
ROUGE-1	0.435714	0.293386	0.340806
ROUGE-L	0.435714	0.293386	0.340806
2 word Summary			
ROUGE-1	0.408928	0.502248	0.439947
ROUGE-L	0.372420	0.465740	0.4034
3 word Summary			
ROUGE-1	0.340873	0.619907	0.430701
ROUGE-L	0.300396	0.559193	0.382100
4 word Summary			
ROUGE-1	0.289384	0.694378	0.401220
ROUGE-L	0.249900	0.615410	0.348500

Table 4: Experimental results of 2520 randomly selected Wikipedia pages with titles of 1-2 words

As the table 4 shown, the experiment conducted on 2520 randomly selected Wikipedia pages with titles of 1-2 words, and we compared n-word summaries(where n is 1,2,3, and 4) respectively with target Wikipedia titles. Since the number of words in the target is in the range of one to two, we only evaluate ROUGE-1 and ROUGE-2 scores. As the table 4 shown, Four-word summaries achieved average of 0.69 recall, two-word summaries achieved 0.43 F-1 scores. Four-word summaries have the highest recall in average and lowest precision, but two-words summaries have the highest f-1 score.

As the Example1 shown, we take the Wikipedia page of Dalhousie University as the input, and generated one word summary, two word summary, three word summary, and four word summary. One word summary gives us the most correct summary, as "dalhousie". The two word summary, "university u15" , makes sense as well, since Dalhousie University is indeed one of U15 Universities in Canada. However, the order of words in the summary is simply based on the weight of each words. Therefore, the generated two word summary is "university u15" instead of "u15 university". We have considered future work to reorganize words into semantic and grammatically correct order by applying linguistic rules along with other NLP models.

"Dalhousie University (commonly known as Dal) is a public research university in Nova Scotia, Canada, with three campuses in Halifax, a fourth in Bible Hill, and medical teaching facilities in Saint John, New Brunswick. Dalhousie offers more than 4,000 courses, and 180 degree programs in twelve undergraduate, graduate, and professional faculties. ***The university is a member of the U15,***

a group of research-intensive universities in Canada. *Dalhousie was established as a nonsectarian college in 1818 by the eponymous Lieutenant Governor of Nova Scotia, George Ramsay, 9th Earl of Dalhousie. The college did not hold its first class until 1838, until then operating sporadically due to financial difficulties.* ***It reopened for a third time in 1863 following a reorganization that brought a change of name to "The Governors of Dalhousie College and University".*** *The university formally changed its name to "Dalhousie University" in 1997 through the same provincial legislation that merged the institution with the Technical University of Nova Scotia....."*

Example 1: Dalhousie University Wikipedia Summarization

As described in the methodology section, the *Sentence Extractor* is suppose to select salient sentences to represent the entire document, and tries to make sure each selected sentence is semantically different from other selected sentences to capture as much context as possible. The *Summarizer* is suppose to selected sentences into one very short summary.

The most representative sentence from Cluster1:

"It reopened for a third time in 1863 following a reorganization that brought a change of name to "The Governors of ***Dalhousie*** *College and University"."*

The most representative sentence from Cluster2:

"The university is a member of the ***U15****, a group of research-intensive universities in Canada."*

At the end, the word "university" is selected from the first sentence, the word "u15" is selected from the second sentence. Since the weight of "university" is higher than "u15", the two-word summary becomes "university u15".

"A paper bag is a bag made of paper, usually kraft paper. Paper bags are commonly used as shopping bags, packaging, and sacks.In 1852, Francis Wolle, a schoolteacher, invented the first machine to mass-produce paper bags. Wolle and his brother patented the machine and founded the Union Paper Bag Company. In 1871, inventor Margaret E. Knight designed a machine that could create flat-bottomed paper bags, which could carry more than the previous envelope-style design. In 1883, Charles Stilwell patented a machine that made square-bottom paper bags with pleated sides, making them easier to fold and store. This style of bag came to be known as the S.O.S., or "Self-Opening Sack". In 1912, Walter Deubener, a grocer in Saint Paul, Minnesota, used cord to reinforce paper bags and add carrying handles. These "Deubener Shopping Bags" could carry up to 75 pounds at a time, and became quite popular, selling over a million bags a year by 1915. ***Paper bags with handles later became the standard for department stores, and were often printed with the store's logo or brand colors.*** *Plastic bags were introduced in the 1970s, and thanks to their lower cost, eventually replaced paper bags as the bag of choice for grocery stores.[4] With the trend towards phasing out lightweight plastic bags, though, some grocers and shoppers have switched back to paper bags. In 2015, the world's largest paper shopping bag was made in the UK and recorded by Guinness World Records. Standard brown paper bags are made from kraft paper. Tote-style paper bags, such as those often used by department stores or as gift bags, can be made from any kind of paper, and come in any color. Paper bags can be made from recycled paper, with some local laws requiring bags to have a minimum percentage of post-consumer recycled content. Paper shopping bags, brown paper bags, grocery bags, paper bread bags and other light duty bags have a single layer of paper. A variety of constructions and designs are available. Many are printed with the names of stores and brands. Paper bags are not waterproof. Types of paper bag are: laminated, twisted, flat tap. The laminated bag, whilst not totally waterproof, has a laminate that protects the outside to some degree. Multiwall (or multi-wall) paper sacks or shipping sacks are often used as shipping containers for bulk materials such as fertilizer, animal feed, sand, dry chemicals, flour and cement. Many have several layers of sack papers, printed external layer and inner plies. Some paper sacks have a plastic film, foil, or polyethylene coated paper layer in between as a water-repellant, insect resistant, or rodent barrier.There are two basic designs of bags: open mouth bags and valve bags. An open mouth bag is a tube of paper plies with the bottom end sealed. The bag is filled through the open mouth and then closed by stitching, adhesive, or tape. Valve sacks have both ends closed and are filled through a valve. A typical example of a valve bag is the cement sack.* ***Paper bags are readily recyclable.*** *Plastic or water-resistant coatings or layers make recycling more difficult. Paper bag recycling is done through the re-pulping of the paper recycling and pressing into the required shapes....."*

n word summary	F-1 Score	Precision	Recall
1 word Summary			
ROUGE-1	0.177596	0.355033	0.118444
ROUGE-2	0.000200	0.000301	0.000150
ROUGE-L	0.177797	0.355033	0.118645
2 word Summary			
ROUGE-1	0.305615	0.382007	0.254721
ROUGE-2	0.027325	0.040988	0.020494
ROUGE-L	0.271508	0.338758	0.226843
3 word Summary			
ROUGE-1	0.352363	0.352320	0.352471
ROUGE-2	0.033423	0.033453	0.033403
ROUGE-L	0.297883	0.297468	0.298774
4 word Summary			
ROUGE-1	0.365352	0.319695	0.426311
ROUGE-2	0.030259	0.025215	0.037823
ROUGE-L	0.300011	0.262256	0.351014
5 word Summary			
ROUGE-1	0.364697	0.291802	0.486286
ROUGE-2	0.027727	0.020795	0.041591
ROUGE-L	0.292237	0.233654	0.390521
6 word Summary			
ROUGE-1	0.355476	0.266676	0.533177
ROUGE-2	0.024455	0.017118	0.042796
ROUGE-L	0.2789864	0.209162	0.419379

Table 5: Experimental results of 3318 randomly selected Wikipedia pages with titles of 3-4 words

---

Example 2: Paper Bag Wikipedia Summarization

In the example of "Paper Bag" Wikipedia page, generated two-words summary is "recyclable bag". The *Sentence Extractor* firstly groups sentences into two thematically different groups, then select the most representative sentence from each group. In this example, two sentences are selected as below:

The most representative sentence from Cluster1:

*"Paper **bags** with handles later became the standard for department stores, and were often printed with the store's logo or brand colors."*

The most representative sentence from Cluster2:

*"Paper bags are readily **recyclable**."*

At the end, the word "bag" is selected from the first sentence, the word "recyclable" is selected from the second sentence. Since the weight of "recyclable" is higher than "bag", the two-word summary becomes "recyclable bag".

4.2 Experiment: Wikipedia Pages with Titles of 1-2 Words

As the table 5 shown, the experiment conducted on 3318 randomly selected Wikipedia pages with titles of 3-4 words, and we compared n-word summaries(where n is 1,2,3,4,5,and 6) respectively with target Wikipedia titles. As the table 5 shown, Four-word summaries achieved average of 0.36 f1-scores on ROUGE-1 and 0.30 on ROUGE-L, but only get 0.03 f1-scores in ROUGE-2; Six-word summaries achieved average of 0.53 recall on ROUGE-1 and 0.41 on ROUGE-L, but only get 0.04 recall in ROUGE-2; Two-word summaries achieved average of 0.38 precision on ROUGE-1 and 0.33 on ROUGE-L, but only get 0.04 precision in ROUGE-2; Even if we achieved great results on ROUGE-1 and ROUGE-L, ROUGE-2 get a pretty low score. The reason is that the order of words in the summary is simply based on the weight of each words, but not takes any approaches to optimize the inter-word order.

"The University of Waterloo (commonly referred to as Waterloo, UW, or UWaterloo) is a public research university with a main campus in Waterloo, Ontario, Canada. The main campus is on 404 hectares (998 acres) of land adjacent to "Uptown" Waterloo and Waterloo Park. ***The university also operates three satellite campuses and four affiliated university colleges.*** *The university offers academic programs administered by six faculties and thirteen faculty-based schools. Waterloo operates the largest post-secondary co-operative education program in the world, with over 20,000 undergraduate students enrolled in the university's co-op program. Waterloo is a member of the U15, a group of research-intensive universities in Canada.****The institution originates from the Waterloo College Associate Faculties, established on 4 April 1956****; a semi-autonomous entity of Waterloo College, which was an affiliate of the University of Western Ontario. This entity formally separated from Waterloo College and was incorporated as a university with the passage of the University of Waterloo Act by the Legislative Assembly of Ontario in 1959. It was established to fill the need to train engineers and technicians for Canada's growing postwar economy. It grew substantially over the next decade, adding a faculty of arts in 1960, and the College of Optometry of Ontario (now the School of Optometry and Vision Science), which moved from Toronto in 1967. The university is a co-educational institution, with approximately 41,000 undergraduate and 6,900 postgraduate students enrolled there in 2019. Alumni and former students of the university can be found across Canada and in over 150 countries; with a number of award winners, government officials, and business leaders having been associated with Waterloo.* ***Waterloo's varsity teams, known as the Waterloo Warriors, compete in the Ontario University Athletics conference of the U Sports****....."*

Example 3: University of Waterloo Wikipedia Summarization

In the example of "University of Waterloo" Wikipedia page, generated three-words summary is "waterloo university college". The *Sentence Extractor* firstly groups sentences into three thematically different groups, then select the most representative sentence from each group. In this example, three sentences are selected as below:

The most representative sentence from Cluster1:

*"****Waterloo****'s varsity teams, known as the* ***Waterloo*** *Warriors, compete in the Ontario University Athletics conference of the U Sports."*

The most representative sentence from Cluster2:

"The ***university*** *also operates three satellite campuses and four affiliated university colleges."*

The most representative sentence from Cluster3:

"The institution originates from the Waterloo ***College*** *Associate Faculties, established on 4 April 1956; a semi-autonomous entity of Waterloo* ***College****, which was an affiliate of the University of Western Ontario."*

At the end, the word "waterloo" is selected from the first sentence, the word "university" is selected from the second sentence, the word "college" is selected from the third sentence. Since the weight of "waterloo" is higher than "university" and the weight of "university" is higher than "college", the three-word summary becomes "waterloo university college".

5 Conclusion

We present a new high-compression text summarization model for very short and fixed-length text summarization that achieved high scores in terms of ROUGE-1 and ROUGE-L metrics. In addition, we have established a website to directly interact with the model to further explore its potential.

The high-compression text summarization model does not require any training and fine-tuning, and the model contains two major components, "Sentence Extractor" and "Summarizer". Ar first, the "Sentence Extractor" selects salient sentences from long documents, then the "Summarizer" summarizes the selected sentences into a very short summary.

Based on our knowledge, this is the first work to generate high compression text summarization, which can be further developed for the application of file name generation, email subject generation and paper title suggestion.

6 Acknowledgment

I would first like to thank my thesis advisor Prof. Evangelos E. Milios of the Faculty of Computer Science at Dalhousie University. The door to Prof. Milios office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge Prof. Sageev Oore of the Faculty of Computer Science at Dalhousie University as the second reader of this thesis.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

References

[1] Andrew Turpin, Yohannes Tsegay, David Hawking, and Hugh E. Williams. Fast generation of result snippets in web search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 127–134, New York, NY, USA, 2007. Association for Computing Machinery.

[2] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques, 2017.

[3] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[4] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 605–es, USA, 2004. Association for Computational Linguistics.

[5] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157, 2003.

[6] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[7] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[8] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[9] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[11] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders, 2019.

[12] Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang goo Lee. Summary level training of sentence rewriting for abstractive summarization, 2019.

[13] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Searching for effective neural extractive summarization: What works and what's next, 2019.

[14] Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization, 2019.

[15] Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[16] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[18] Kaiqiang Song, Bingqing Wang, Zhe Feng, Liu Ren, and Fei Liu. Controlling the amount of verbatim copying in abstractive summarization, 2019.

[19] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation, 2019.

[20] Kai Wang, Xiaojun Quan, and Rui Wang. BiSET: Bi-directional selective encoding with template for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2153–2162, Florence, Italy, July 2019. Association for Computational Linguistics.

[21] Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. Learning thematic similarity metric from article sections using triplet networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[23] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[24] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, 2001.

[25] Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716, 2003.

[26] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems : a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

[27] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

A Website: System Design

As the Figure 5 shows, the user enter the number of words in the summary and paste original document in the document, whenever the user click the "Summarize" button, it triggers a post request to the web server which is running on a python flask framework. Then it passes request information to High Compression Text Summarization Model to get the request, then send the result back to user.

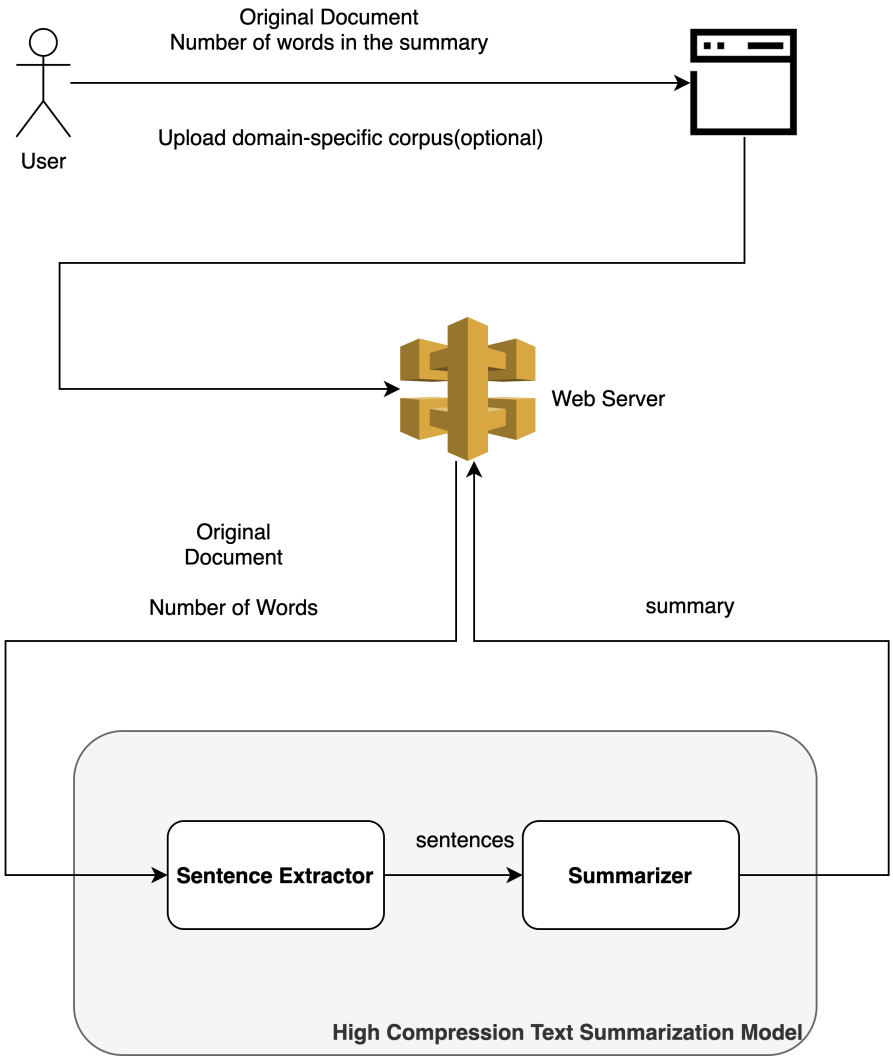

Figure 5: Architecture: High-Compression Text Summarization Website