6515 Machine Learning
Assignment 4
By

Yahu Wang (B00757055) yh288977@dal.ca
Anqi Chen(B00838586) an883236@dal.ca

1.Background

Automatic text abstracts received attention as early as the 1950s. In the late 1950s, Hans Peter Lunn published a research paper entitled "Automatic Generation of Literature Abstracts", which extracted important sentences from the text by using the features of word frequency, phrase frequency and so on.

The text summarization main has 3 big steps: Intermediate representation, Score sentences, and Select summary sentences.

For intermediate representation, it must use that representation to summarize and identify what is important. The topic representation method converts text into an intermediate representation interpreted as the topic discussed in the text. Some of the most popular summarization methods rely on topic representation, which exhibits impressive differences in complexity and presentation power. For example, the frequency, TF. IDF method and topic lexical method. Topic representation is composed of a simple word list and its weight. The higher the weight is, the more it can represent the topic. Lexical chain methods, in which a thesaurus such as WordNet is used to find semantically related words of the topic or concept and then give the concept weight.

For score sentences, each sentence has a score indicating its importance. Scores are often related to how well sentences express some of the most important topics. In intermediate representation, the weight works the same as the score so we do not repeat it.

For select summary sentences, the famous methods are Greedy Approaches and Global Summary Selection. In greedy approaches, we always choose the sentences which get the highest score, which usually works well in most cases. The global optimization algorithm is used to solve the new summary formula and choose the best overall summary. If some constraints are imposed on the summary, such as maximizing information, minimizing duplication, and matching the length of the summary, then the task is to select the best summary. The difference between two methods is if there are constraints for summary or not
.

2.Models

This assignment aims to conduct a research on the subject of Text Summarization Techniques and find two methods of Text Summarization. In our case, we used a frequency-driven approach and latent semantic analysis to make the summary.

2.1 Frequency-Driven Approach

This method mainly focuses on the words appearance in the whole text file. Its criterion for concluding sentences is the probability of important words appearing in a sentence. In other words, if one sentence contains many important words, then it is the summary sentence. The standard of important words is how many times it appears in a text file. And we give every sentence a score and use the average score as a threshold. The sentences whose score higher than the threshold are the summary sentences.

In general, this approach mainly has four step:

2.1.1 Get the frequency of words

```
freq_table = frequency_table(text)
print(freq_table)
```

```
{'mental': 41, 'models': 31, ':': 3, 'train': 2, 'brain': 3, 'think': 4, 'new': 6, 'ways':
```

We use dict() in python which works as a hashtable, which the words is the key and the value is the number of times it appears in the text file. In the example here, the word "mental" appears 41 times in the whole text file, and so for the rest of words.

2.1.2 Get the score of sentence

As example above, the "mental" appears more often than other words so the sentence which contains "mental" will have a higher score. And we process all words in one sentence and give it a final score. We use the formula p(w) = c(w)/N, which p(w) is computed from sentence, c(w) is the occurrence of words, N is the length of sentence.

2.1.3 Find the threshold according to the average score

After we get all the scores, we calculate the average score of sentences for the text file and use it as a threshold.

2.1.4 Use the threshold to find summary sentence

And finally, the sentence whose score is higher than threshold, we consider it as the summary sentence.

2.2 Latent Semantic Analysis

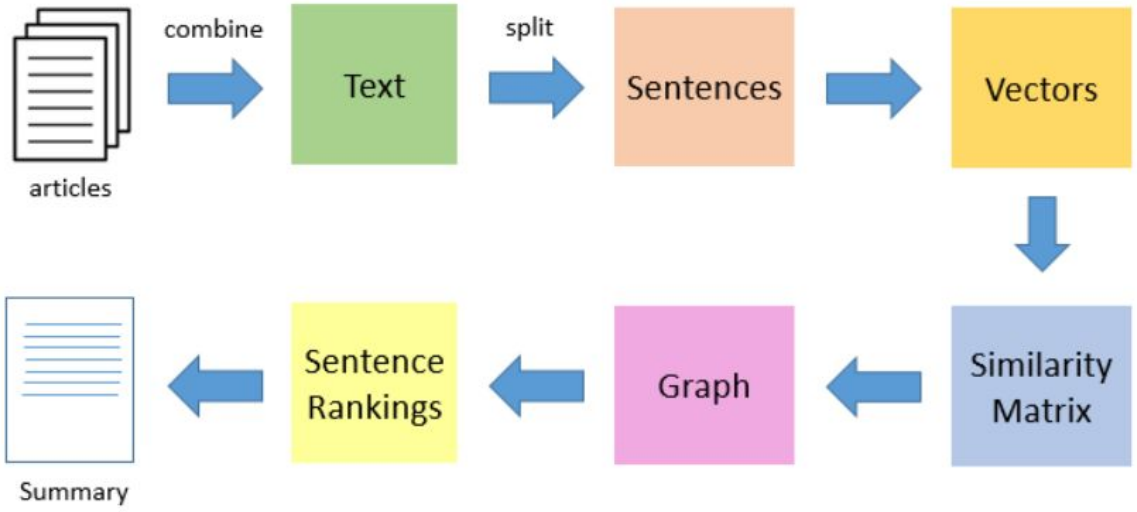
This method works different from frequency driven method, but it also have mainly four steps:

1.Clean the sentences,including punctuations, numbers, stopwords and special characters(This is required for Glove Word embedding)

2.Create vector for each sentences

3.Use consine similarity to find the similarities between each sentence. Build the similarity matrix. According to the article ***A SURVEY OF TEXT SUMMARIZATION TECHNIQUES :*** "*Building the topic representation starts by filling in a n by m matrix A: each row corresponds to a word from the input (n words) and each column corresponds to a sentence in the input (m sentences). Entry of the matrix corresponds to the weight of word i in sentence j. If the sentence does not contain the word, the weight is zero, otherwise the weight is equal to the TF*IDF weight of the word."[4]* Therefore, the higher the weight, in our case, score, the higher rank of sentences. That means these sentences have a strong connection to other sentences. In other words, they might be talking about the topic.

4.Apply textRank algorithm by using library networkx



In general, we use Glove Word Embedding to let sentences become vectors, and we use the cosine similarity approach to build the matrix and initialize this matrix with cosine similarity scores of the sentences. Next, we transform the similarity matrix into a graph. The nodes in the figure represent sentences, and the edges represent similarity scores between sentences. In this figure, we will apply the PageRank algorithm to get the ranking of sentences. Finally, we pick the top 10 sentences as the summary sentences.

2.3 Seq2Seq Model

We've also explored the way that RNN algorithms work by building a Sequence-to-Sequence model to summarize these articles. The Seq2Seq model can be divided into 2 major

components: encoder and decoder. The encoder will create context vectors and pass all hidden states to the decoder as output. The decoder receives these states and is responsible to transfer them as the target vectors. The attention layer can be added in the process to take a look at the relevant parts and define the more important ones.

After data cleaning and preprocessing in the first step, we used Tokenizer and pad_sequences in Keras.processing to convert text sequences into meaningful numerical form. Before feeding into the model, the special tokens for differentiate the start and end of each sequence are also added.

To implement the model, we mainly used Keras.layers to do this work. The model is built with several layers: input layer, embedding layer, 3 stages of LSTM layers, attention layer, concatenate layer and a dense layer. In each layer, we set up some parameters to make it more effective. After building up the model, we compile the model using an optimizer. We tested for RMSProp, Adam, and Momentum and chose RMSProp for our dataset. These data are trained for prediction, so next in the inference phase they are used by the decoder to generate summaries -- the next word with the maximum probability will be chosen and put into the target sequence.

A sample output is shown below:

```
Text: train brain think better one best ways expand set mental models use think let explain mean sharing story world
class thinker first discovered mental model useful right one could reading story richard feynman famous physicist fey
nman received undergraduate degree mit ph princeton time developed reput
Original title: mental models how to train your brain to think in new ways you can train your brain to think better o
ne of the best ways to do this is to expand the set of mental models you to think
Predicted summary:  is better the need set to new mental direction but you can do one do expand you better how you ex
pand set models
```

3.Results

The main focus of a frequency-driven approach is on the Frequency of each word. As far as it is concerned, when a word is mentioned repeatedly, it means that it is an important word that needs to be emphasized repeatedly in an article. So this word is usually used in a summary statement. For example, in the first sample article, “mental model” are the words that appear most frequently, and the whole article also describes things about the mental model. So for this method, the statement that contains these two important terms is the summative statement. The advantage of this approach is that you can quickly find topic Word. You don't need a lot of articles to do this. In a nutshell, it's fast and effective. But he also has a shortcoming that he can't deal with short sentences well. For example, the summary of the first article includes the sentence "He had a broader set of mental Models". The reason is that when the sentence is too short and contains important words, the length N of the whole sentence will be very small, leading to a high score obtained by the formula PW = CW /N. And we can see clearly that this is not a good summary sentence.

But for Latent Semantic Analysis, it would hardly have such problems. The reason for this is that it mainly focuses on the relevance of one sentence to another according to the

model trained by large amounts of text, and there is nothing to do with word frequency in sample articles. If a sentence has a high degree of correlation with other sentences, prove that the other sentences are described around that sentence. A sentence like this is usually a concluding statement. So no matter how long the sentence is, as long as it is related to the other sentence, it is a summary sentence. This is also one of his advantages over Frequency Driven Approach. However, its own problem lies in word embedding. In our case, we use the Glove Word Embedding method and this model requires lots of text data to build the vectors for each word. Because the sample article itself is not long enough to build a vector accurately so we downloaded nearly 823M of extra text data from Google.

When it comes to the Seq2Seq model, the main limitation is that the structure is more complex than other ways and the predicted summary results depend on the training process. For our datasets which only contain 3 articles (not lengthy), the results are not as good as what we expect. One of the benefits of this method is that the model is trainable, which means it can be more powerful when the training sets have enough word vocabulary and related word pairs. Another benefit is that it learns from the patterns of vectors and generates a sequence of new text as summary for us, not simply pick a line in the article. It can cope with not only short sequences but also long sentences with the help of attention mechanism. The reconcurrent networks offer a lot of flexibility to do different types of text summarization, one-to-one, many-to-one or many-to-many. Here, in our example, we generated three summaries for these articles at once. Also, once we have a well-performanced network model, we can feed new data and reuse it.

4.Conclusion

So in general, the advantage of Frequency Driven Approach is that the summary sentence can be found quickly and without a lot of text data. the advantage of Latent Semantic Analysis is that it finds a summary sentence that is more relevant to the topic of the article and other important points but it requires a large amount of text data.

If the end user input only has a small amount of text and contains many long and complicated sentences such as personal article, email or tweet, we would recommend the frequency driven approach.

If the end user input wants to prepare a bullet-point summary by scanning through multiple articles online, we would recommend the Latent Semantic Analysis.

If the end user input has a large number of dataset, for example a list of news or reviews, and want to generate summaries for each record to get a better understanding of its main idea or even do some translation, we would recommend the Seq2Seq approach.

5.Personal Contribution

Yahu Wang: Manipulation of frequency driven approach and Latent Semantic Analysis

Anqi Chen: Exploring and implementation of Seq2Seq Model

6.References

[1] *Entropy: Why Life Always Seems to Get More Complicated*
https://jamesclear.com/entropy

[2] *Mental Models: How to Train Your Brain to Think in New Ways*
https://jamesclear.com/feynman-mental-models

[3] *First Principles: Elon Musk on the Power of Thinking for Yourself*
https://jamesclear.com/first-principles

[4] A. Nenkova and K. McKeown, "A Survey of Text Summarization Techniques," in Mining Text Data, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 43–76.
https://link.springer.com/chapter/10.1007%2F978-1-4614-3223-4_3

[5] "Seq2Seq Model | Sequence To Sequence With Attention," Analytics Vidhya, Mar. 15, 2018.
https://www.analyticsvidhya.com/blog/2018/03/essentials-of-deep-learning-sequence-to-sequence-modelling-with-attention-part-i/ [Accessed Nov. 28, 2020].

[6] "Text generation with an RNN | TensorFlow Core," TensorFlow.
https://www.tensorflow.org/tutorials/text/text_generation [Accessed Nov. 28, 2020].

[7] "Text Summarization | Text Summarization Using Deep Learning," Analytics Vidhya, Jun. 10, 2019.
https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/ [Accessed Nov. 28, 2020].

[8] Text Summarization using NLTK: TF-IDF Algorithm
https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3

[9] An Introduction to Text Summarization using the TextRank Algorithm (with Python implementation)

https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/