Text Summarisation

Author

Raktim Bijoypuri IIT2018125

Introduction

- With the present explosion of data circulating the digital space, which is mostly non-structured textual data, there is a need to develop automatic text summarization tools that allow people to get insights from them easily.
- Social Media, News articles, emails, text messages (the list goes on..), generate massive information and it becomes cumbersome to go through lengthy text materials.
- For example, if you are looking for specific information from an online news article, you may have to dig through its content and spend a lot of time weeding out the unnecessary stuff before getting the information you want.
- To provide a solution to this problem we are trying to build a deep learning model that will create summary of text leaving out inessential and insignificant data

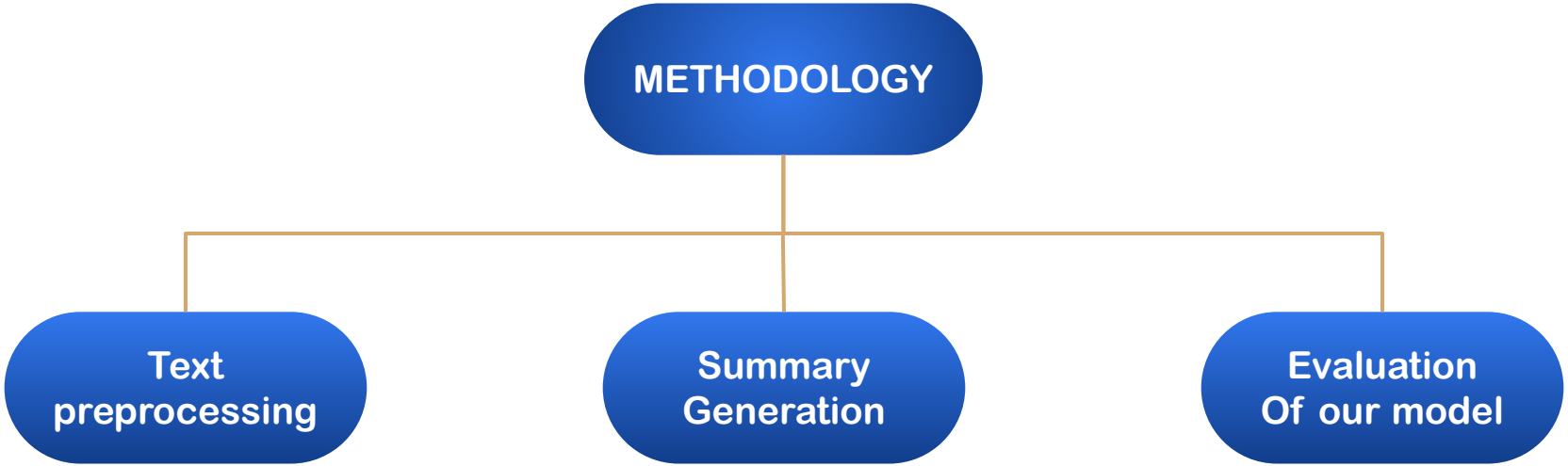
Problem Statement

- So our problem is to generate summary from some given text documents.
- This is known as Automatic text summarization, it is a common problem in machine learning and natural language processing (NLP).
- Given a wikihow article we will be generating its summary using deep learning methods.
- This is a Many-to-Many Seq2Seq problem so for given sequence of long text we need to generate a smaller output sequence(summary).
- Generally, variants of Recurrent Neural Networks (RNNs), i.e. Gated Recurrent Neural Network (GRU) or Long Short Term Memory (LSTM), are preferred as the encoder and decoder components
- Further details are discussed in the proposed methodology section.

Dataset Description

- We will work on the wikihow dataset that contains around 230,000 long sequence pairs of articles and their headlines. This dataset is one of the large-scale datasets available for summarization with the length of articles varying considerably. These articles are quite diverse in their writing style which makes the summarization problem more challenging and interesting.
- Each article consists of multiple paragraphs and each paragraph starts with a sentence summarizing it. By merging the paragraphs to form the article and the paragraph outlines to form the summary.

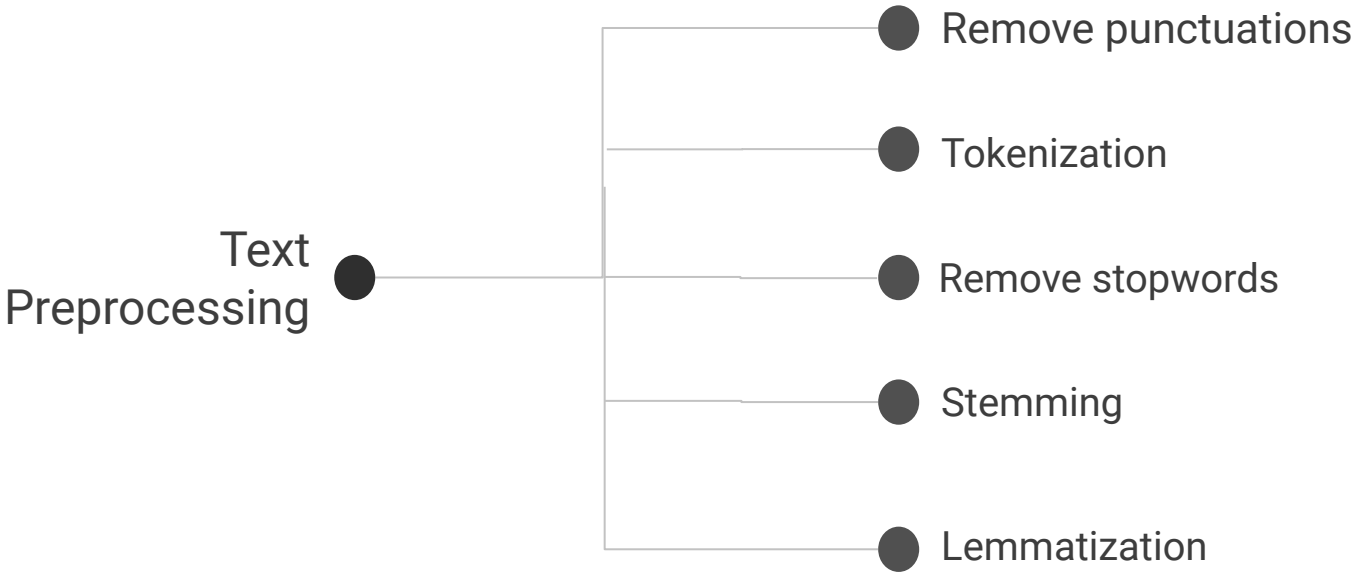
METHODOLGY


METHODOLOGY
Text preprocessing
Summary Generation
Evaluation Of our model

Text preprocessing

- Why it is important?
- Normally, the text data available is often noisy. It contains a lot of punctuations, emotions, different case of letters. Besides when we think about human language there are different language to express the same thing.
- So it is important to preprocess out text data before we feed that to our model. And convert it into its basic form where it is easier for our model to get trained on.

Continued...


Text Preprocessing
Remove punctuations
Tokenization
Remove stopwords
Stemming
Lemmatization

1. Remove punctuations - In this step all the unnecessary punctuations, irrelevant characters, stray whitespaces will be removed.
2. Tokenization - It is the process of breaking the text data into smaller pieces called tokens. Words, numbers can be considered as tokens. It helps us to create the term-document matrix.
3. Remove Stopwords -
    a. Stop words contain common pronouns, prepositions, articles etc. which are very frequent in text data but do not provide any valuable information.
    b. These words and their abundance may cause problems to our model training as the weight for each stop word will be very large although presence of these words contribute very little to the similarity of two documents

Continued...

4. Stemming - Stemming is a crude heuristic process that chops off the ends of words in order to get to their base form. For example run, running, runs, ran, derived from the same word as run. It is important to consider those kind of words as same since it strongly points that the related document may be similar

5. Lemmatization - The stemmatization technique is not so efficient as most of the time it chops of important words. To solve this problem we need lemmatization. Lemmatization essentially does the same thing as stemmatization but in a more efficient way.It takes into account both the inflectional and derivational morphology.

Encoder-decoder network

- Generate word embeddings for words in the vocab.
- Use the encoder-decoder architecture to make a seq-2-seq model.
- Encoder reads the input sequence(paragraph) and summarizes the information in something called the internal state vectors.
- The outputs generated by the encoder are discarded and only the context vector is passed over to the decoder.
- The decoder unit generates an output sequence based on the context vector.
- The output sequence is the generated abstract summary.
- We can also evaluate the results of our method using metrics such as BLEU.

Contributions

- We will experiment with several word embeddings.
- We will use attention layers/transfromers and experiment on small changes to architecture.
- We will use both Extractive Summarization and Abstractive Summarization

THANK YOU