Automatic Text Summarization Using Machine Learning

[1]Sanjay H Pillai, [1]Raheek Mohammed, [1]Yadhu Krishna, [1]Sangeeth S Krishnan, [2]Jithin Jacob

[1]*UG Scholar, Department of Computer Science and Engineering*
[2]*Asst. Prof, Department Of Computer Science And Engineering, UKF College of Engineering and Technology*

---***---

Abstract- *Our project combines human and machine-based approaches to text summarization using a multi-stage extractor-abstractor network. We use an initial abstract generated by Google's Pegasus abstraction model as a reference to create an initial extraction generated by using a novel statistical extraction model that we created. The method uses word frequency and likelihood of occurrence in the reference summary to generate an accurate extract from the original document, this is then used as an input to the abstractor. This method allows us to create an accurate extracted input to the abstraction stage, this improves the efficiency and speed of the abstractor thereby improving the overall performance of the system. This method is similar to human abstraction method, we first extract important parts of the document before it is shortened to create an abstract sentence.*

Key Words: Summarization, Extraction, Abstraction

1. INTRODUCTION

An effective text summarization technique is an exhaustively researched area in the field of Natural Language Processing. Text summarization requires shortening of text documents while preserving their intended meaning. This is usually achieved in two ways, either using extraction or abstraction.

The extraction method [1] [2] [3] involves selection of important sentences from the original document such that the total number of sentences in the final document is less than that of the original document. In this method there is no rephrasing of the sentences from the original document as they are included without change to the final summary. This method provides a fast but primitive abstract.

The abstraction method [4] [5] actively rephrases the sentences from the original document in a concise manner before including them to the final summary. This method requires machine learning models that can rephrase sentences while their meaning remains unchanged. Even though abstractive method provides a more human like summary, the process is often slow and erroneous as sometimes the final summary may contain sentences that are semantically inaccurate or convey a different meaning that that is intended in the original document.

Although some models combine [6] the use of extraction and abstraction, they are not mainstream and applications that implement these methods for a large user base do not exist, therefore our summarization system aims to be accessible to a large user base via an online platform. The website will be an easy-to-use application that allows users to upload large documents and create summarised documents free of cost.

We propose a novel summarization technique which is a combination of both human and machine approach to text summarization, we first create a reference abstraction of the original document using Google's Pegasus abstraction model, this abstract is used just as a reference to determine the salient sentences from the original document. We use a likelihood-based extraction method in which we find the probability of adding a sentence from the original document to the extracted summary using the Pegasus abstract as a reference.

Our contributions therefore include:

A free summarization tool that will be widely accessible to a large userbase, a novel extraction mechanism that can provide accurate extracted summaries using a statistical approach, A novel abstract-extract-abstract system that can provide much more accurate results.

2. MODEL

The model works by first creating a reference abstract using the Pegasus model [7], this is then used to determine the salient sentences from the original document using a statistical approach, this shortened version is then passed through the abstractor again to create a final summary.

The initial document is first summarised abstractedly using the Pegasus model and some important words from the original document is appended to this abstract, this forms the reference document containing salient words from the initial document, now we make another separate document that contains unimportant words from the initial document. These two documents then act as reference points which is used to classify whether a sentence from the initial document should be added to the final extract or not. Once the extracted summary is obtained, we again use the Pegasus model to abstract the extracted summary, this method ensures that we get an accurate summary that includes all the salient sentences from the initial document that is rephrased so that the final summary is concise and semantically correct. This method therefore incorporates the advantages of both the extractive and abstractive approach.

3. RELATED WORKS

We find that the majority of the works done on automatic summarization is mainly focused on extraction or compression-based approaches. There are also some recent abstraction-based approaches using neural abstractive models [8] [9], RL based metric optimization [10], coverage [11] [12] [13], copy mechanism [14] [15], graph-based attention [16] etc. Reinforcement Learning is also an extensively used method in this field, Q-learning based RL for extractive summarization [17] is also an effective method. RL policy gradients have been used for producing accurate abstractive summaries.

Some works in this field also include use of selective gates [18] to improve attention in abstractive summarization. There are also works that attempted cascaded small non-recurrent networks on extractive QA, producing a scalable, parallelizable model [19].

4. EXTRACTOR

The extractor is a new sentence level selection mechanism that uses naïve bayes to estimate the maximum likelihood for a sentence to be added to the extracted summary. To achieve this, as mentioned above we first create a temporary reference abstract using the Google's Pegasus model. Then we use a statistical technique to find salient words from the initial document, for this we first determine the frequency of occurrence of each no filler word in the original document, then we use other parameters like capitalization and spacing to find other salient words that were not included in the initial abstract. These words are then appended to the abstract such the it contains almost all the salient words from the initial document while ignoring all the unimportant filler words from the original document. These filler words are then included into another file which is also used as reference to perform naïve bayes.

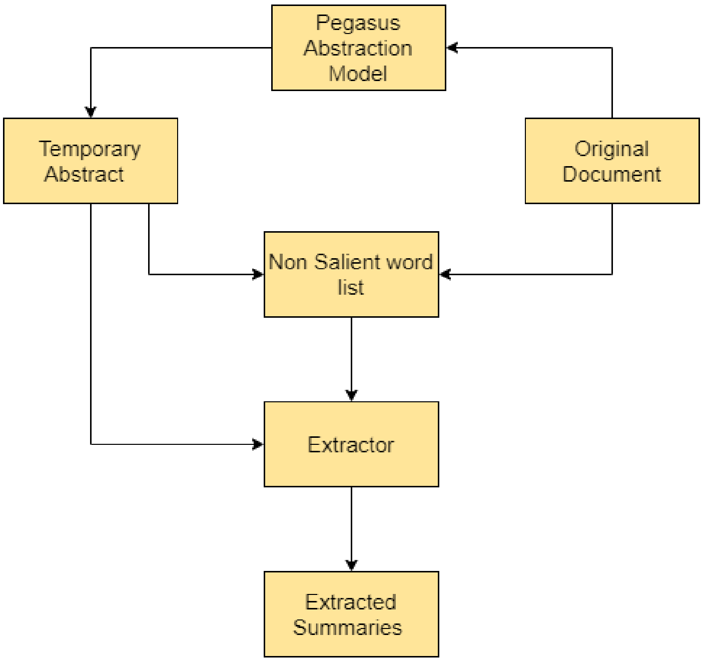

Fig -1: Model Schema

For each sentence in the original document, we find its likelihood to be included in the extracted summary by multiplying the probability of occurrence of each word in the sentence to the prior probability.

$$\mathrm{L} = P \cdot \Pi_{i=1}^{n} P(x_i)$$

Likelihood **'L'** is given by multiplying the probability of occurrence of each word in the given sentence by the prior probability **'P'**. If the calculated likelihood is lower with respect to the temporary reference abstract, then the sentence will not be included in the extracted summary and vice versa.

5. WORD WEIGHTAGE

We estimate the weight or saliency of a word to be include to the reference abstract using a statistical approach. We find the frequency of occurrence of non-filler words and use that as the first indication of saliency. In some cases, there might be salient words which might not have a high frequency of occurrence, in such cases we use other parameters like capitalization and spacing to determine the importance of such words. Capitalized words and nouns are given a higher weightage and has a higher chance to be included in the temporary abstract.

6. ABSTRACTOR

We use the Pegasus model for the two abstraction stages within our project. The first stage abstraction produces a larger file that tries to include more salient sentences, this is used as reference along with the weighted words selected from the original document to act as reference for initial extraction. Once the initial extraction is completed, we use the extracted output as the input for the second stage of abstraction, since the input of the second stage of abstraction is more concise than the original document, the second stage of abstraction will be much faster and provide much more accurate results as the input is tailored during the initial extraction stage. We intend to use our on novel abstraction system that produces faster results than Pegasus system as a future scope of our project, as of now we decide to continue with the Pegasus system as it provides concise and accurate abstracts that works well with our new extraction mechanism.

7. USER INTERFACE

The UI is an easy-to-use web application that allows the user to select and upload the documents to be summarized directly to our servers which then employs our summarization algorithm to produce an accurate summary and then returns a downloadable link to the user. The web application is developed using HTML, CSS, JavaScript front end stack. For testing purposes, we host the web application initially on a local Linux server using apache2.

8. HUMAN VALIDATION

To ensure the accuracy and semantic consistency of our summarization algorithm, we conduct human evaluation of the output at various stages to ensure that the output meets the various criteria required. We also compare the output of our summarization techniques to the output produced by other commonly used summarization algorithms to compare speed and accuracy. Human validation is only conducted during the testing phase and is not required after the deployment of the application.

9. CONCLUSION

We find that the summarization technique proposed has the following contributions -

1. A free summarization tool that will be widely accessible to a large userbase
2. A novel extraction mechanism that can provide accurate extracted summaries using a statistical approach
3. A novel abstract-extract-abstract system that can provide much more accurate results.

We find that our summarization technique is slightly slower than some state-of-the-art fast abstraction and extraction systems due to our three-stage process but we find that due to this process we are able to produce much more accurate results consistently.

REFERENCES

[1] Hongyan Jing and Kathleen R. McKeown (2000), cut and paste based text summarizer, which uses operations derived from an analysis of human written abstracts.

[2] Kevin Knight and Daniel Marcu (2000) Summarization beyond sentence extraction: A probabilistic approach to sentence compression.

[3] Andre F. T. Martins and Noah A. Smith 2009; Summarization with a Joint Model for Sentence Extraction and Compression.

[4] Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation, Association for Computational Linguistics.

[5] David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In HLT-NAACL 2004 Document Understanding Workshop, pages 112–119, Boston, Massachusetts.

[6] Yen-Chun Chen and Mohit Bansal UNC Chapel Hill 2018; Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting.

[7] Peter J. Liu and Yao Zhao, Software Engineers, Google Research 2020; PEGASUS: A State-of-the-Art Model for Abstractive Text Summarization.

[8] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-tosequence rnns and beyond.

[9] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In ICLR.

[10] Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In ICLR.

[11] Jun Suzuki and Masaaki Nagata. 2016. RNN-based encoder-decoder approach with word frequency estimation. In EACL.

[12] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In IJCAI.

[13] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer generator networks.

[14] Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In EMNLP.

[15] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning.

[16] Swabha Swayamdipta, Ankur P. Parikh, and Tom Kwiatkowski. 2017. multi-mention learning for reading comprehension with neural cascades.

[17] Sebastian Henß, Margot Mieskes, and Iryna Gurevych. 2015. A reinforcement learning approach for adaptive single- and multi-document summarization.

[18] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarizer: A recurrent neural network-based sequence model for extractive summarization of documents.

[19] Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization.