March - April 2020
ISSN: 0193-4120 Page No. 11666 - 11670

GUI Application for Summarizing and Optimizing Audio and Textual Data

[1]Suyash Khare, [2]Ashwin Rajpurohit, [3]R.Brindha
[1,2]Computer science and engineering, SRM Institute of Science and Technology, Chennai, India
[3]Assistant professor, Computer science and engineering, SRM Institute of Science and Technology, Chennai, India
[1]Suyashkhare_ra@srmuniv.edu.in, [2]Ashwin_ravindra@srmuniv.edu.in, [3]brindha.ra@ktr.srmuniv.ac.in

Article Info
Volume 83
Page Number: 11666 - 11670
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 16 April 2020*

Abstract
Text summarization is one of those implementations of nlp which have been popularized greatly over time. Earlier versions were simple and sometimes were based on only one basis. nowadays summarization can broadly be divided into two main categories to classify it. They are abstract and extractive summarizers. the implementation of seq2seq model for summarization of textual data using tensor flow/ keras and demonstrated on amazon or social response reviews, issues and news articles. Text summarization basically works by cutting out the excess data and giving only the required data. LSTM summarization which we used in this project of ours trains the machine to make meaningful sentences based on the dataset given. So, our aim is to compare spacy, gensim and nltk summarization technique by the input requirements and try to prove that the implemented LSTM based summariser is more efficient and better.

Keywords: *Recurrent neural network; natural language processing; LSTM; neural networks and tensorflow*

1. INTRODUCTION

Computers do not think like humans in an abstract manner they need numbers and facts to work with every thing is interpreted as strings of numbers. Thus it becomes very difficult to make such a machine understand the complexities and difficulties of the natural language spoken by the people. Languages written or textual have been developed over the span of hundreds over thousands of years, each year adding its own complexities to the language. To make computers understand such languages it is necessary to map the information into numbers and data but this also comes with its own share of difficulties. Natural language processing is used to make these machines understand the language used by us humans and tells them how to decode it. There are a lot of machine learning algorithms and mechanism developed for us to implement but the sheer amount of information we have to decode for the computers when it comes to natural languages are huge to say the least. We have text data, written data and speech and where speech may vary person by person.

We can train models in accordance with the information we have and over time the machine becomes proficient but it never comes close to human level as at the end of the day what machine is doing is only giving data based on the data which means nothing to it. It can learn to differentiate between nown, verbs and parts of speech after enough training. We will be focusing basically on the summarization part of the nlp branch which is further complicated as the machine has to understand the text to summarize the data. Though many models have been implemented based on extractive methods it fails to give a proper meaningful summary like a human. Abstractive summarizers are more efficient as they give meaningful sentences instead of just cutting out the sentences and lstm based models can be very efficient for such methods as they use long range dependencies so they can make meaningful sentences more easily.

2. RELATED WORK

Lstm is a very widely used method when it comes handling data with long term dependencies, But there is very little work done in the text summarization field when it comes to the use of lstm to make the summaries better. Very few research works show the benefit of using such advanced machine learning techniques and the impacts it has.

Lstm networks work in a different way when compared to the rnn network, lstm is used when we have to find a long

March - April 2020
ISSN: 0193-4120 Page No. 11666 - 11670

term relation between two data, it gets the dependencies the data have over the long range and gives the output based on the data it acquires. Lstm is very useful when it comes to summarization as it kind of works in the same way a human thinks and processes data. Various studies have been done on this feature of the lstm based models where the researchers have compared the human and lstm on the basis of their thinking and summarizing capabilities. This test was conducted by Takahashi and Tanaka-Ishii.

Although the study can not prove that lstm works on the same level as the human it need to be noted that the test was done in a very limited time period and the data given to the computer was also limited due to time and budget in most of the cases.

As a matter of fact, when compared to the rest of the fields very limited amounts of time and study have been put into a proper analysis and to a methodological evaluation of the similarities between textual material created by machines and by humans.

In this paper algorithm used are, Gensim summarization (It is based on text rank algorithm), Spacy summarization, Nltk summarization and LSTM summarization.

Text summarization is basically a process of identifying the most important meaningful parts in a document or set of related documents and compressing them into a shorter version preserving its overall meanings by using nltk algorithm. Our analysis provides a comprehensive guide to sensitivity analysis of model parameters with regard to comparing gensim summarization, spacy summarization and nltk summarization with evaluation of GUI based application results.

3. SYSTEM ARCHITECTURE

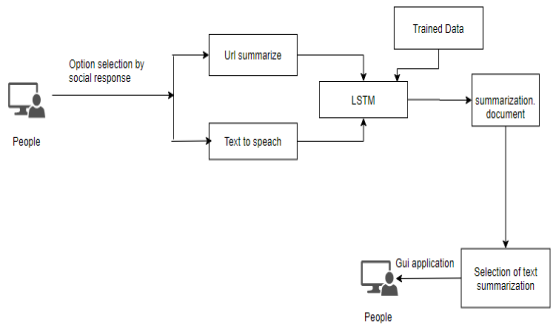

Figure 1.1: Architecture diagram

System architecture model as the name suggests defines the structure of the project in a diagrammatic form. It informs about the processes and the components involved in the operation. Our project as shown in the diagram starts with the user, then it takes the information in the url, text or speech format. The data is then passed through our lstm model which uses the trained data. The text is then summarised and sent to the gui to compare to the nltk based summariser results and allow the user to see and check the differences. The Output in the presented in the audio format is the user desires so by the gui. The final output is delivered back to the user thus completing the cycle shown.

4. METHODOLOGY

NLTK Summarization Technique

Create the word frequency table

The word frequency table creates a dictionary for the word frequency table from the text. For this, we should only use the words that are not part of the stop words array. Tokenize the sentences. It split the text string in a set of sentences, uses the inbuilt method from the NLTK. Score the sentences: Term Frequency method to score each sentence. Find the threshold Threshold is considered as the average score of the sentences. It is also possible to use other methods to calculate the threshold. The sentence in this summarization is only considered if the score it got crosses a specified certain score.

Spacy Summarization

Spacy is one of the most used python libraries used when dealing with the nlp problems. It comes to the play when we have to process a large amount of data. Processing in Python. Spacy is used mainly for production use. It can be used to in deep learning for pre-processing.

LSTM Networks

Lstm networks work in a different way when compared to the rnn network, lstm is used when we have to find a long term relation between two data for example when we have to find what language a person speaks based on the country he used to live in. This kind of problem becomes very complicated for the rnn to complete as it can not access the data so far removed. Lstm was fine tuned for this kind of problem only and face no problem whatsoever in solving these kind of problems.

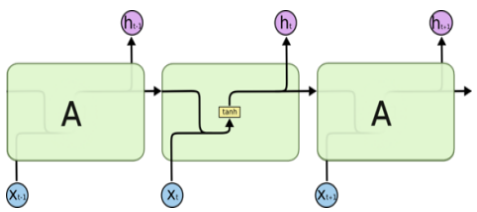

Figure 1.2: Lstm simple model

Shown below is the complete working of the lstm and different modules interacting with each other. The input is

March - April 2020
ISSN: 0193-4120 Page No. 11666 - 11670

passed through a tan based function and in such a way the all the data is always somehow accessible

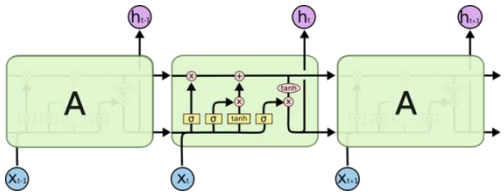

Figure 1.3: The repeating module in LSTM

5. IMPLEMENTATION

In this project we will be summarizing our data using the amazon reviews downloaded from amazon. this paper will basically train the lstm machine based on the data we obtained from the dataset, we will score the data based on the ranking given in the dataset and will map the 5 and 4 star data as 1 and 3,2,1 star review as 0. We will be training the machine first by using 70% data and test using the 30%. It will then study the performance analysis graph to check the efficiency of the project. The main motive is to use that lstm based summarizer and compare with the rest of the available nltk modules and get a better result.

It will also design a specific gui based interface which will give the user the flexibility to work with different formats of inputs and outputs. We have used the speech recognition modules to convert the audio(wav) to text and google gpi module to convert the text back into the audio. The projects focused on the flexibility of the process which the user can achieve even while using lstm concepts and also gives the user a side by side comparison with the rest of the modules. The main training and testing part is done in python language amd we have used deep learning modules like tensorflow and keras to implement the lstm based summarizer

Recurrent Neural Networks

$$g_t=sig_g(v_f.x_t+s_f.h_{t-1}+z_f)$$
$$h_t=sig_g(v_i.x_t+s_i.h_{t-1}+z_i)$$
$$i_t=sig_g(v_o.x_t+s_o.h_{t-1}+z_o)$$
$$j_t=f_t*p_{t-1}+i_t*sig(p)(v_cx_t+s_ch_{t-1}+z_c)$$
$$k_t=o_t*sig_h(p_t)$$

Recurrent neural network or rnn is used for training the machine to think like how humans do. when we humans do the thinking we keep the things we studied or learned earlier in mind to correlate the data and accurately fill in the gaps. Machines on the other hand don't know how to implement such technique and thus get stuck when faced with the problem which requires it to have beforehand information. This is where rnn comes into play rnn works by basically creating a loop thus allowing the data to not be thrown away. Using this looping mechanism the network can guarantee that the information will persist. This kind of efficiency was not available in the traditional neural networks. Figure shown below explains how the looping works in the recurrent neural network

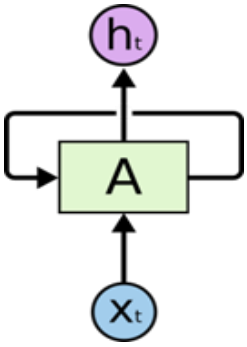

Figure 1.4: Rnn looping structure

In the diagram give above , a part of network, AA, As shown in the above diagram the input is taken as X_t and similarly the output is taken as h_t the input is passed through the A function to get the output and the process is done in a recurrent manner. We can understand it further when we unbind the loop, observing how the information is being passed after every cycle.

It is basically the same operation being repeated over and over again to make the data persist as shown in the figure below this what is what happens if we unroll the loop:

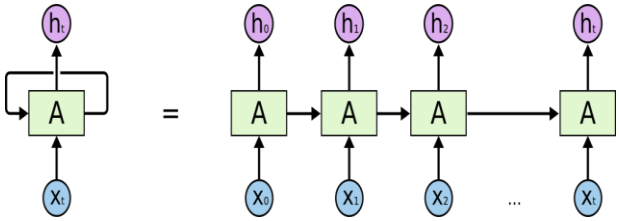

Figure 1.5: unrolled rnn model

As shown above the network closely represents a string like structure when unrolled. Rnn is very widely used in text recognition and speech recognition.

Comparison

SpaCy Spacy is one of the most used python libraries used when dealing with the nlp problems. It comes to the play when we have to process a large amount of data. Processing in Python. Spacy is used mainly for production use. It can be used to in deep learning for pre-processing

March - April 2020
ISSN: 0193-4120 Page No. 11666 - 11670

Gensim It is a Python library basically very useful for the vector based models and data preprocessing and working with bigger files becomes easier.
Keras Deep Learning library available in python language and works with tensorflow for neural networks.
Nltk It's set of libraries which is used in natural language processing and is mainly used in research purposes, it make use of tokenization and sequence scoring for summarization

6. RESULTS

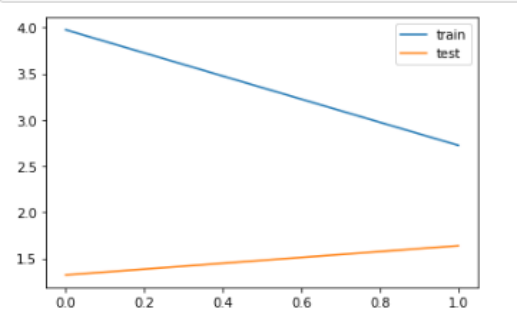

Figure 1.6: Plot of loss after second epoch

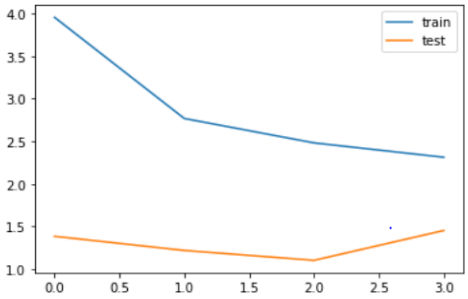

Figure 1.7: Plot of loss after fourth epoch

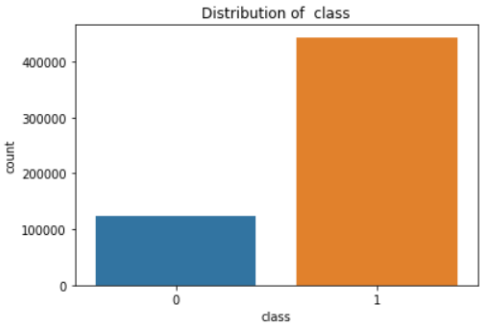

Figure 1.8: bar graph of the qualified and not qualified data

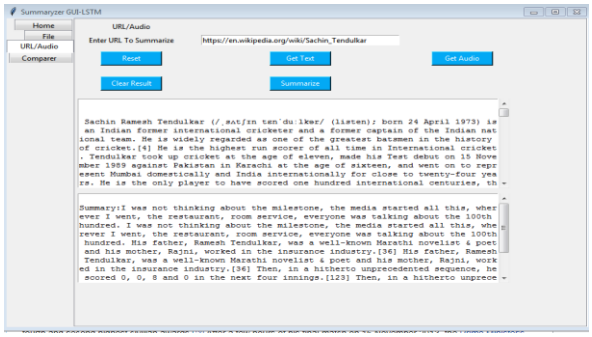

Figure 1.9: project screenshot

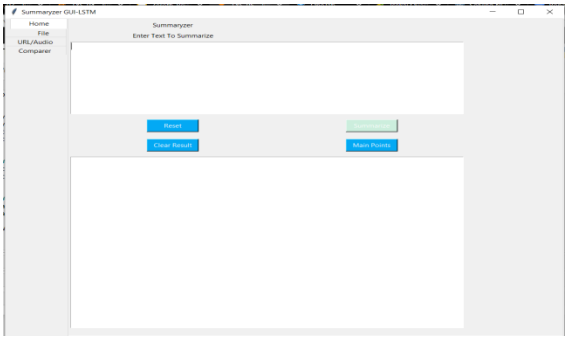

Figure 1.10: project screenshot

7. CONCLUSION

This paper presented in the text summarization of comparison results by input and output requirements with the foundation of knowledge in functional grammar, our method contains various phases and comparison of various techniques: like Spacy summarization, NLTK summarization, Gensim summarization, and custom LSTM based summariser the paper strives to prove that the lstm based approach taken gives more optimum result further work can be put in the efforts to further improve the projects, other datasets can be used along with better machines to implement the project better. In future with given advancement in technology the process can be totally automated.

REFERENCES

[1] M. A. Boden, "Creativity and artificial intelligence," Artif. Intell., vol. 103, nos. 1–2, pp. 347–356, 1998.
[2] E. Reiter, R. Dale, and Z. Feng, Building Natural Language Generation Systems, vol. 33. Cambridge, MA, USA: MIT Press, 2000.
[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, May 2015.
[4] Y. Bengio, "Learning deep architectures for AI," Found. Trends Mach. Learn., vol. 2, no. 1, pp. 1–127, 2009.

[5] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 3128–3137.
[6] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence–video to text," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 4534–4542.
[7] X. Zhang and M. Lapata, "Chinese poetry generation with recurrent neural networks," in Proc. EMNLP, 2014, pp. 670–680.
[8] P. Potash, A. Romanov, and A. Rumshisky, "GhostWriter: Using an LSTM for automatic rap lyric generation," in Proc. EMNLP, 2015, pp. 1919–1924.
[9] W. Ebeling and A. Neiman, "Long-range correlations between letters and sentences in texts," Phys. A, Stat. Mech. Appl., vol. 215, no. 3, pp. 233–241, 1995.
[10] M. A. Montemurro and P. Pury, "Long-range fractal correlations in literary corpora," Fractals, vol. 10, no. 4, pp. 451–461, 2002.
[11] E. G. Altmann, G. Cristadoro, and M. D. Esposti, "On the origin of long-range correlations in texts," Proc. Nat. Acad. Sci., vol. 109, no. 29, pp. 11582–11587, 2012.