Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Speech Summarization using Essence Vector Modeling

Aathiswaran S
Department of Computer Science
Dr. Mahalingam College of Engineering and Technology

Gokulapriya S
Department of Computer Science
Dr. Mahalingam College of Engineering and Technology

Bharath Kumar D
Department of Computer Science
Dr. Mahalingam College of Engineering and Technology

Gayathri S
Department of Computer Science
Dr. Mahalingam College of Engineering and Technology

***Abstract*— Speech is the most natural and effective method of communication between human beings. It is not easy to review and retrieve the information present in the spoken document instantly when the speech is recorded and stored as an audio signal. Speech summarization extracts important information and saves time for reviewing speech documents and improves the efficiency of document retrieval. Representation learning and paragraph embedding methods have emerged as new active research subjects because of their excellent performance in many applications. In this research, we have implemented an unsupervised paragraph embedding method called Essence vector (EV) modeling for Speech Summarization, which strives at distilling the most characteristic information from a paragraph and also includes the general background information to produce a more informative low-dimensional vector representation of the input. Speech is converted into text and provided as a high dimensional vector input to the Essence Vector (EV) model which in turn summarizes the input in a low dimensional space. We have enhanced the performance of the EV model by using LSTM, a state-of-the-art neural network architecture, to handle the imperfect and erroneous Speech conversions made by the speech to text module and also added Attention mechanism to the EV model to obtain a better and relevant summary. The proposed method is evaluated using ROUGE metrics against a summarization corpus.**

Keywords—Essence Vector(EV), Long Short Term Memory(LSTM)

1. INTRODUCTION

Speech is the most fundamental and effective method of communication found between human beings. Humans have a remarkable ability to understand speech quickly and are able to focus on the meaning of speech of a particular speaker. This speech can be stored in the form of digital audio signals or text. The use of spoken language processing (SLP) technology has grown to become a major adjunct to the rapidly growing use of digital technologies involving spoken data.

Since Speech is the most valuable source of information, and with the rapid development of digital applications, unprecedented volumes of data such as lecture recordings and meetings have become an integral part of our everyday life, it is crucial to represent such large data concisely and precisely including only the important points and giving the overall theme of the data. Using speech summarization, one can efficiently digest the amount of information present in the spoken documents with minimal human interference.

Our project focuses on summarizing a large amount of spoken data using natural language processing techniques and thus minimizing the time taken for manually extracting meaningful information from large speech-related data. Our project aims to create an Automatic Speech Summarization System using state of the art neural network architectures to summarize spontaneous speech into text.

2. LITERATURE SURVEY

2.1 An Information Distillation Framework for Extractive Speech Summarization

In the context of text summarization, Kuan-Yu Chen and Shih-Hung Liu proposed a novel unsupervised paragraph embedding method, named the Essence Vector (EV) model [1], which aims at extracting the most representative information from a paragraph. The proposed method eliminates the general background information to produce a more informative low-dimensional vector representation for the input paragraph. According to [1], Classical paragraph embedding methods infer the vector representation by considering all of the words occurring in the paragraph, and Therefore the stop words that occur frequently in the paragraph might drift the theme of the content into producing an irrelevant summary. Each paragraph can be assembled by two segments: the paragraph specific information and the general background information. The proposed method consists of three modules: a paragraph encoder which converts the paragraph-specific information into low-dimensional vector representation; a background encoder, which converts general background information into a low-dimensional vector representation; and a decoder that can restore the original paragraph by combining the paragraph representation and the background representation. Secondly, an extension of the EV model named the denoising essence vector (D-EV) model is proposed, which not only acquires the advantages of the EV model but also can infer a more robust representation for a given spoken paragraph against imperfect speech recognition. An additional module, a denoising decoder is introduced on top of the EV model to equip the ability to extract true information from a given spoken paragraph.

2.2 Abstractive Text Summarization using Sequence to sequence RNNs and Beyond

According to [2], Abstractive text summarization is not merely an extraction of a few sentences from the main content, but a compressed representation of the main content, combined using vocabulary different from that of the original document. The main contributions in this paper include

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

implementing off-the-shelf attentional encoder-decoder RNN that was initially developed for machine translation to summarization and improving the model performance against significant problems in summarization such as modeling keywords and omitting rare or unseen words in the input document. This work shows that many of the other proposed models contribute to further improvement in performance.

The baseline model includes an encoder consisting of a bidirectional GRU-RNN, while the decoder consists of a unidirectional GRU-RNN with the same hidden-state size as that of the encoder. An Attention mechanism is applied over the source-hidden states and a soft-max layer is used over the target vocabulary to generate words. The key entities in the input document were identified by capturing linguistic features such as part-of-speech tags, named-entity tags, and TF and IDF statistics of the words combining them into an embedding matrix. The omitting of rare and unseen words was solved by introducing Tokens to point the location of those words in the input document enabling the decoder to generate a pointer to the word-position in the source document. The state-of-the-art performance in Abstractive text summarization was achieved by using Attentional Encoder-Decoder Recurrent Neural Network.

2.3 A Deep Sentence Embedding Using LSTM

In this paper [3], Hamid Palangi et al. proposed a novel method for sentence embedding, using recurrent neural networks (RNN) with Long Short-Term Memory (LSTM) cells. In this proposed method [3], RNN is used to sequentially accept each word in a sentence and map it into a vector space along with the historical information. At the end of the input sequence, the hidden activations form a natural embedding vector for the contextual information present in the sequence. LSTM cells are incorporated into the RNN model to facilitate long term memory learning in RNN. The main idea behind using RNN for sentence embedding is to capture the context information in the sequence using recurrence and to find a dense low dimensional semantic representation by processing each word in a sequence and mapping it into a low dimensional vector. Backpropagation Through Time is used to train the LSTM-RNN model. To attenuate unimportant information, the hidden states and cell states gradually absorb contextual information and evolve. The input gates emerge in such a way that it attenuates the unimportant information and extracts important information from the input sequence. During Analysis, by focusing on significant activation changes, the LSTM-RNN model was found to extract important keywords. LSTM-RNN model not only extracts important keywords but also allocates them properly to different cells according to the topics. The proposed model [3] was robust to noise since it mainly embeds keywords in the final semantic vector representing the whole sentence.

3. LSTM BASED ESSENCE VECTOR MODELING

In order to remove the effects of incorrect speech transcription and stop words in the summarized content and also to increase the efficiency of the existing EV model, the existing traditional multilayer neural network is replaced with a gated recurrent network known as long short-term memory (LSTM) model. Gated RNNs are based on the idea of creating paths through time that have derivatives that don't fade or collapse. Instead of applying an elementwise nonlinearity to the inputs, LSTM recurrent networks have “LSTM cells” that have internal states, in addition to the outer recurrence of the RNN. Each cell has more parameters than the ordinary recurrent unit with the same inputs and outputs. Each LSTM Cell has a system of gating units that can collect or free memory on the go by using the gating mechanism.

An LSTM cell has three gates, an input gate which uses a sigmoid function with a tan-h activation function, to control the information added to the current state, an output gate which controls the information transferred to the next state and a forget gate, which decides whether to store or erase the information from the previous state. LSTM networks learn long-term dependencies more easily than the other RNN architectures thus providing a more relevant and abstract summary.

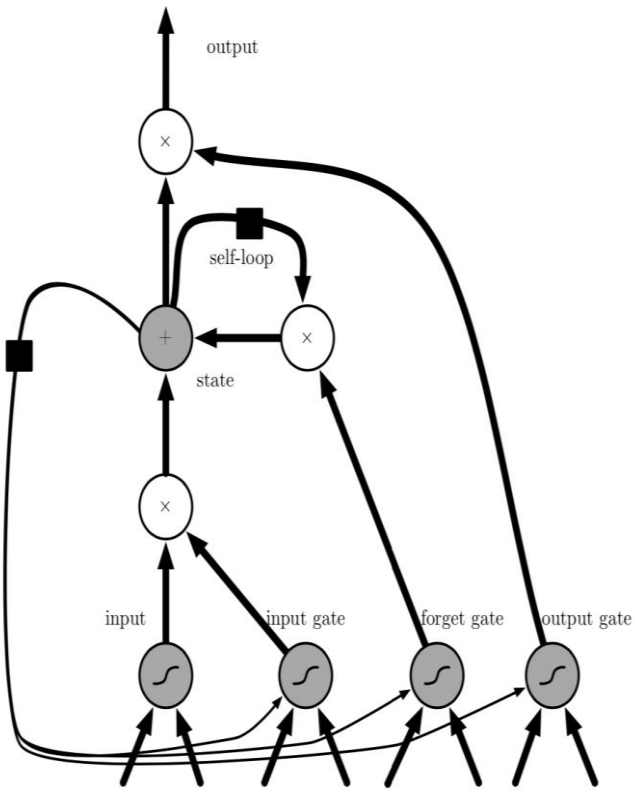

Figure 1: Block Diagram of an LSTM Cell

4. METHODOLOGY

3.1 Speech to Text

Initially, the Spoken document is converted into Text Data using Pocket Sphinx, a light-weight speech recognition engine for transcribing spontaneous speech. Speech is a continuous audio stream where stable states are mixed with dynamically changed states known as phones. Pocket Sphinx uses three models to transcribe spontaneous speech, an acoustic model which contains the acoustic properties for the phones, a phonetic model which contains the word to phone mapping and a language model which defines which word could follow previously recognized words and significantly restrict the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

matching process. Pocket Sphinx extracts phonemes from the audio source using an acoustic model and then maps those phonemes to sequences of words using a language model thus transcribing the spoken data into text.

3.2 Preprocessing of Input Data

Given a set of input paragraphs, Standard NLP pre-processing techniques like removing stop words and punctuations and Stemming are performed. During the training phase, each paragraph that is to be summarized will be represented by two input data, the paragraph data that is to be summarized and the background data that provides the general background information about the input paragraph. During the inference phase, only the paragraph data will be used. The corresponding Vector representations of the input paragraphs are obtained by using Glove, an unsupervised learning algorithm which stands for "Global Vectors", that captures both global and local statistics to obtain word vectors. Pre-Trained Glove Vectors trained using the Wikipedia text data are used to represent each word in the paragraph into a high dimensional word embedding of size 100. The word vectors are padded to eliminate variable-length inputs. Start-of-Sequence (SOS) and End-of-Sequence tokens are introduced for handling arbitrary length data in the Encoder-Decoder Architecture. The proposed method consists of three modules: an encoder module where the inputs are summarized, an interpolation module where the inputs are interpolated based on attention and a decoder module, where the summaries are decoded.

3.3 Encoder – LSTM

The proposed method is based on the Seq2Seq model also known as Encoder-Decoder architecture which maps fixed-length input to fixed-length output where the length of input and output data differs. The Proposed method has two encoders-a paragraph encoder f(.) made of a stack of LSTM Layers which summarizes the input paragraph information into internal state vectors known as thought vectors and a Background Encoder g(.) also made of LSTM layers which summarizes the general background information into thought vectors. The LSTM Cell has two states, an internal hidden state, and a cell state. Each LSTM cell reads the data one sequence after another such that if the input is a sequence of length 'x', then the LSTM Cell reads it in 'x' time steps. In the final time step, the hidden state and cell state vectors of an LSTM layer are combined to form the thought vectors. Paragraph encoder and Background encoder should have the same architectures since during interpolation the output of the two encoders (thought vectors) should have the same dimensions.

Given a set of training input paragraphs Dt along with the Background information BGt, the inputs are then pre-processed to obtain vector representation of words in each paragraph. Each input paragraph is represented by Glove high dimensional Vector PDt, where each element corresponds to a high dimensional vector of a word in Glove. Then the paragraph encoder f(.) is applied to summarizes the information in the paragraph vectors into thought vectors.

$$f\left(P_{D_t}\right) = v(D_t) \quad (1)$$

At the same time, the background encoder g(.) is applied to the vector representations of the general background information to obtain the summarized thought vectors. During the training phase, both the encoders are used and in the inference phase only the paragraph encoder is used.

$$g(P_{BG}) = v_{BG} \quad (2)$$

3.4 Interpolation

Interpolation is a type of estimation method used for constructing new data points within the range of a set of known data points. In theory, interpolation is used to extricate data about situations using known experiences to expand knowledge into areas that are unknown.

$$\text{Interpolation} = \alpha_{D_t}.v_{D_t} + (1 - \alpha_{D_t}.v_{BG}) \quad (3)$$

where α_{Dt} is known as the interpolation weights which can be determined by an Attention Function *q(·,·)* which is a trainable neural network, that helps to focus on specific input sentences by assigning probability score to each input sequence.

$$\alpha_{D_t} = q(v_{D_t}, v_{BG}) \quad (4)$$

When *αDt* is high, then the interpolated output will be biased towards the output of the paragraph encoder and when *αDt* is less, then the interpolated output will be biased towards the output of the background encoder. At the end of training phase, the paragraph encoder will be able to identify and extract the useful information from the input sequences.

3.5 Decoder – LSTM

The Decoder h(.) behaves little differently during training and inference phases. Encoder will scan the input sequence word by word. Similarly, the Decoder will generate the output sequence word by word. A Technique called 'Teacher Forcing' developed as an alternative to backpropagation for training RNN architectures is used to train the Decoder, where the input word at each time step is given as the actual Output word from the previous time step. During the inference phase, the Decoder LSTM initialized with the output of the encoder, is called in a loop generating one word at each time step. The Output generated during the current timestep is given as the input to the next time step.

5. EXPERMINETS

5.1 Dataset

The CNN/Daily mail summarization dataset is used for training and testing the proposed method. This dataset contains the documents and summaries from the news articles of CNN. There are two features: - article: text of news article, used as the document to be summarized - highlights: joined text of highlights with and around each highlight, which is the target summary.

5.2 Evaluation Metric

For the assessment of summarization performance, we adopt the commonly used ROUGE metric by taking ROUGE-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

1, ROUGE-2 and ROUGE-L (in F-scores) as the main measures. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation.

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \epsilon S_H} \sum_{g_n \epsilon S} C_m(g_n)}{\sum_{S \epsilon S_H} \sum_{g_n \epsilon S} C(g_n)} \quad (5)$$

Where,

S_H is the set of manual summaries

S is an individual manual summary

g_n is a N-gram and $C(g_n)$ is the number of occurrences of g_n in the manual summary and automatic summary.

5.3 Expermient

The capability of the Essence vector model for speech summarization is investigated by experimenting against the CNN stories dataset. During the Training phase, standard preprocessing techniques are applied to both training and validation datasets. The inputs are vectorized using the Glove Pretrained model. The dimension of the embedding is chosen as 100. Keras functional API backed by TensorFlow is used to build the encoder-decoder architecture. Paragraph encoder and background encoder can have the same or different architecture. In this experiment, both the paragraph and background encoders consist of an LSTM layer with 100 hidden units, with a dropout. After the encoder reads the entire input sequence, the output of the encoders is fed into an Attention Layer for Interpolation. Merge Layers are used to interpolate the thought vectors and the resultant vector is used to initialize the decoder. The decoder architecture consists of an LSTM layer and a highly connected dense layer with a linear activation function. Categorical cross-entropy is chosen as the loss function and the model is optimized using Root Mean Square optimizer. The EV Model is trained using a dual-core CPU processor. During the inference phase, the pocket sphinx module is configured, and the audio file is converted to text. The converted text is vectorized using Glove and given as input to the paragraph encoder. The thought vector from paragraph encoder is used to initialize the decoder. The Start-of-sequence token is given as the initial input to the decoder. At each time step, the current state of the decoder is preserved and is used to initialize the decoder for the next timestep. At each timestep, the current predicted output is fed as the input to the next time step. The Process is repeated until the end-of-sequence is reached. The outputs of the decoder are preserved and combined to generate the required summary.

5.4 Result

ALGORITHM	ROUGE
Gensim Text Rank	0.27837
Lex Rank	0.32530
Latent Semantic Analysis	0.29515
Luhn Algorithm	0.20789
Edmudson Summarizer	0.20789
Essence Vector Model	**0.34343**

Table 1: ROUGE Scores of Summarization Algorithms

1) Experimental results indicate that the performance of Essence Vector modeling implemented using sequence-to-sequence architecture for speech summarization is likely on par with other classical unsupervised summarization techniques like Lex Rank and Latent Semantic Analysis.

2) It is concluded from the experiment that, use of the "Teacher Forcing Algorithm" for training the decoder can result in poor prediction due to the generation of error compound in the output context.ie., Sequence generated during training will be different from that of the sequence generated during testing.

3) Due to the closed-loop model of sequence to sequence model architecture, the output is more of a prediction than that of compression.ie, the words present in the summary will be slightly different from the words present in the original context.

6. CONCLUSION

In this work, we have extended the Essence Vector modeling algorithm using sequence-to-sequence architecture for speech summarization. The efficiency of the Essence vector model is increased by using LSTM's and the proposed method is successfully evaluated against a summarization corpus. Thus, we have concluded from the experiments that the proposed system is likely in par with other classical unsupervised summarization techniques like Lex Rank and Latent Semantic Analysis.

REFERENCES

[1] Chen, Kuan-Yu, et al., "An information distillation framework for extractive summarization.", IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017.

[2] Nallapati, Ramesh, et al., "Abstractive text summarization using sequence-to-sequence rnns and beyond.", arXiv:1602.06023, 2016.

[3] Palangi, Hamid, et al., "Deep sentence embedding using long short-term memory networks: Mikolov, Tomas, et al, "Efficient estimation of word representations in vector space.", arXiv:1301.3781, 2013.

[4] Bengio, Yoshua, et al., "A neural probabilistic language model.", Journal of machine learning research, February 2003.

[5] Le, Quoc, and Tomas Mikolov, "Distributed representations of sentences and documents.", International conference on machine learning, 2014.

[6] Kågebäck, Mikael, et al., "Extractive summarization using continuous vector space models.", Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), 2014.

[7] Chen, Berlin, Hao-Chin Chang, and Kuan-Yu Chen, "Sentence modeling for extractive speech summarization." IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2013.

[8] Kikuchi, Tomonori, Sadaoki Furui, and Chiori Hori, "Automatic speech summarization based on sentence extraction and compaction.", IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, 2003.

[9] Liu, Yang, Shasha Xie, and Fei Liu, "Using N-best recognition output for extractive summarization and keyword extraction in meeting speech.", IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2010.

[10] Lee, Hung-yi, et al., "Unsupervised domain adaptation for spoken document summarization with structured support vector machine." IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013.

[11] Liu, Tzu-En, Shih-Hung Liu, and Berlin Chen, "A Hierarchical Neural Summarization Framework for Spoken Documents.", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

[12] Zhang, Justin Jian, and Pascale Fung, "Learning deep rhetorical structure for extractive speech summarization.", IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2010.

[13] Devlin, Jacob, et al., "Bert: Pre-training of deep bidirectional transformers for *language understanding."*, arXiv:1810.04805, 2018.

[14] Tsai, Chun-I., et al., "Extractive speech summarization leveraging convolutional neural network techniques.", IEEE Spoken Language Technology Workshop (SLT), IEEE, 2016.

[15] Chen, Kuan-Yu, et al., "A recurrent neural network language modeling framework for extractive speech summarization.", IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2013.

BIOGRAPHY

Aathiswaran S.
Student
Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu

Bharath Kumar D.
Student
Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu

Gokulapriya S.
Student
Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu

Gayathri S.
Professor
Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu