Volume 9, No. 3, March 2018
Journal of Global Research in Computer Science
RESEARCH PAPER
ISSN: 2229-371X
Available Online at http://www.jgrcs.info

SUMMARISATION OF LONG TEXT EXTRACTED FROM ARTICLE IMAGES BY INTEGRATING EXTRACTIVE AND ABSTRACTIVE TEXT SUMMARISATION METHODS

Jayaraj Balagopal[*1], Cicci Maria Xavier[2], Krishna Priya P[3] and Rahul V Reji[4] and Anjali S[5]
*[1] U.G. Student, Department of Computer Engineering Model Engineering College,
Thrikkakara Cochin, India
jayarajbalagopal.mec@gmail.com[1], ciccimariaxavier.mec@gmail.com[2]
krishnapriyapadikkal.mec@gmail.com[3]rahulreji.mec@gmail.com[4],anjalisivakumar@gmail.com [5]

Abstract: People spend too much time reading huge articles, gist of which might be quite small. They also have tendencies to skip articles with essential content leading to not properly understanding the idea that the writer presents. Some articles also have a complex word which makes the articles difficult to perceive. This paper aims at extracting text from images of articles and summarising the latter using concepts in machine learning. The paper also identifies complex words from the document and substitutes simple words for the same. Sentence reconstruction is also performed to shorten long sentences to increase the scale of summarisation. The user can take images of the article he wishes to shorten and upload it to the server. The proposed application will extract the text from the image and provide the summarised version of the article to the user.

Keywords: Extractive summarisation, Abstractive summarisation, Parzen-window density function, LexRank, Encoder-Decoder Model, Attention Mechanism

INTRODUCTION

"Information Overload" is one of the major issues we face today.The rapid development of the Internet has brought in massive information on the web. The major challenge is to efficiently access the information available.

Automatic Text Summarisation allows to process information and compress them to produce concise and information-rich content. Automatic Text Summarisation can be broadly classified into extractive and abstractive text summarisation methods[2]. Extractive summarisation selects the best sentences from the original text, whereas , abstractive summarisation requires understanding the document and producing the summary taking into consideration the similarity aspects between sentences. Thus, an abstractive summary will be more condense and information-rich. But, it is difficult to develop a program that incorporates natural language generation technology to produce concise summaries. Most of the summarisers available use extractive text summarisation methods. RNN based sequence-to-sequence learning(seq2seq) can be used in various natural language processing tasks.

In this paper, we present a three-phase approach to long text summarisation. The 3 major phases are sentence extraction and clustering the document , generating abstractive summaries of clusters and extractive summary generation.

The image or pdf of the article can be uploaded and the valid text is extracted from it.In the clustering phase, we divide the document into clusters by using Parzen-window density function. Then,

we evaluate each of the clusters and produce abstractive summary for each cluster. We use the modified-LexRank algorithm on the generated summary sentences to produce the final summary.

RELATED WORK

Widely used text summarisation techniques are Automatic text summarisation techniques. They are classified into extractive and abstractive summarisation techniques[2]. Extractive summarisers select the best sentences from the document to form the summary, while abstractive summa

risers use methods to understand the document and generate new summary sentences.

Extractive summarisers use statistical and linguistic features of the sentence to assign score to each sentence[3]. The word-level features like content-word feature, title word feature, cue

phrase feature, uppercase word feature along with the sentence-level features like sentence location feature, sentence length feature and paragraph location feature are used to determine the rank of a sentence. Extractive text summarisation methods can be classified into supervised and unsupervised methods. TextRank and LexRank are graph based approaches to text summarisation

TextRank uses voting-based weighting algorithm, whereas, LexRank uses cosine transform based weighting algorithm to determine the score of a sentence. The sentences are sorted according to the scores obtained and a threshold or length cutoff is used to limit the size of the summary.

Sentence Clustering groups sentences or short texts into logical groups. After encoding sentences into vectors, clustering algorithms are applied to group them according to their relative similarity.

Abstractive summarisation retrieves information from documents to generate precise summary of information. It develops new sentences from text documents. An abstractive summariser summarises information into an easily readable and grammatically correct form.

Remarkable success is achieved using deep learning techniques which suggests a feasible framework for abstractive summarisation, with RNN based seq2seq model. seq2seq combines a representation learning encoder and a language modelling decoder to perform mappings between two sequences and is constructed on the encoder-decoder framework.

A basic seq2seq model consists of two recurrent neural networks: an encoder that processes the input and a decoder that generates the output.Encoder and decoder can share weights or use a different set of weights. Every input has to be encoded into a fixed-size state vector, and it is passed to the decoder. An attention mechanism is introduced here. Such approaches offer a fully data-driven solution to both semantic and discourse understanding.

PROPOSED MODEL

In this section, we define the architecture of the proposed model and the tasks involved in the three phases that follow.

Model Overview

The proposed model combines the advantages of extractive and abstractive summarisation methods, so that the overall performance is enhanced.

The system architecture of the proposed model is shown below.

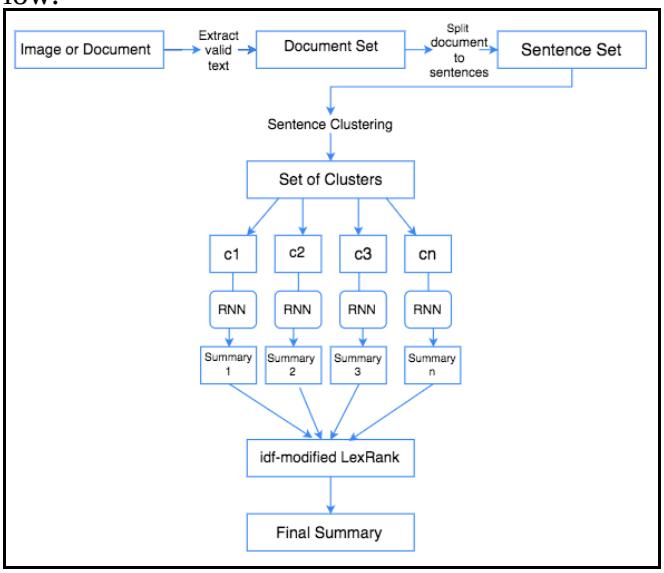

Figure 1. System Architecture.

First, the image or document is uploaded by the user. The text extractor extracts valid text from the input provided. The set of sentences in the extracted text are fed into the clustering phase. Similar sentences are clustered. Abstractive summarisation is performed in each cluster to produce a single line summary. The summaries generated from each cluster are fed into the extractive phase to generate the final summary.

Phase 1 : Sentence Extraction and Clustering

In this phase, the image or document uploaded by the user is passed through text extraction algorithm and the sentences extracted from the set of documents are returned.

From the sentence set, similar sentences are grouped into clusters.[4] This is performed by the clustering algorithm which uses the Parzen-window density function.

Clustering Algorithm

1: Input: The sentence set S.
2: Output: Sentence Cluster set C.
3: Compute all word vectors
4: Compute sentence vectors by taking average of all word vectors of the sentence
5: for all sentence s in S do
6: Add Vs to sentence vector set Sv
7: End
8: For all sentence vector Vs in Sv do
9: Calculate parzen density p
10: Add p to set P
11: For each point
12: Find closest point with highest density
13: Find all points that is not outlier an add it to the set of centres C (clusters)

14: For each point that is not in centre find the cluster that it belongs to and add it to that cluster.

15: return C.

Phase 2 : Abstractive Summarisation

Each cluster passes through the abstractive summariser to produce a single summary sentence. In this phase, we use a RNN-based encoder-decoder model with attention mechanism to produce the abstractive summary.

The sentences in a cluster are fed into an RNN based encoder-decoder model, in order to get a one sentence summary for the cluster. Gated RNN alternatives like GRU or LSTM provides better performance. The sentences in cluster are tokenised and each word in the sentences are converted to a fixed length vector c. Creating word vectors lets us to analyze words mathematically. Either word2vec or GloVe can be used to create a vector representation of words. GloVe is count based algorithm which uses a co-occurance matrix.

GloVe is simple than word2vec and can be used to create the initial embedding matrix. For every word outside the embedding matrix we find the closest word inside the matrix by measuring the cosine distance of GloVe vectors.

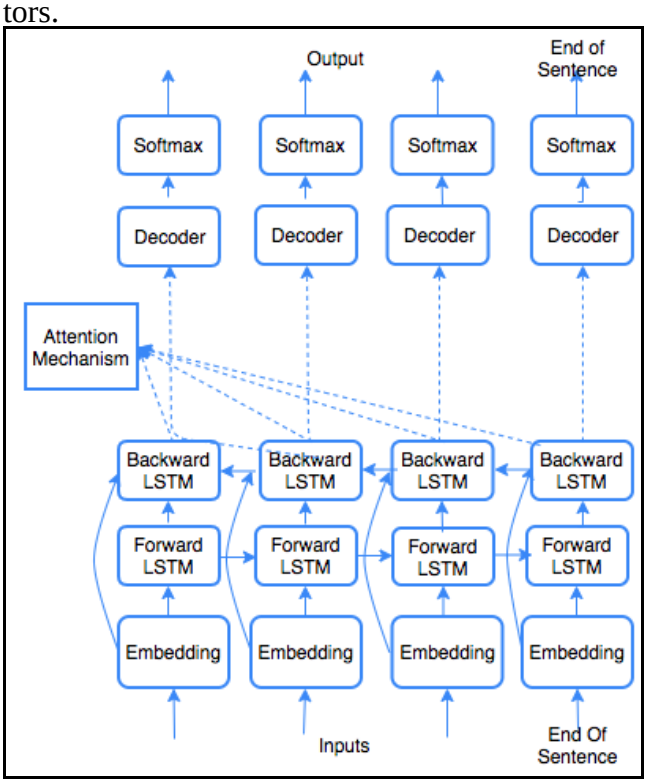

Figure 2. Encoder-Decoder Architecture.

Encoder-decoder architecture: The model uses 2 recurrent networks. First is the encoder network, it takes an input sequence and creates an encoded representation of it.

The second is the decoder network. When building encoder network the weights of the first layer can be set as the pre-trained embeddings.Embedding layer is used to turn input integers into fixed size vectors. Newspaper articles and their corresponding headlines can be used to train the network.The decoder network takes as input the vector representation. At first it will create its own representation using its embedding layer. The next step is to convert these representation into words.

Attention mechanism: The model uses attention mechanism while outputting each word in the decoder. For each output word, it computes a weight over each of the input words that determines how much attention should be given to that input word. All the weights sum up to 1 and are used to compute a weighted average of the last hidden layers generated after processing each of the inputted words. The model takes the weighted average and input it to the softmax layer.

Phase 3 : Extractive Summarisation

The set of abstractive summary sentences are fed into the extractive summariser. Extractive summarisation is performed using idf-modifed LexRank algorithm to produce the final summary. The summary generated will be according to the compression rate provided by the user.

Extractive Summarisation

1: Input: Sentence set S(Output of abstractive summarisation phase),cosine threshold t
2: Output: Final Summary
3. Array SimilarityMatrix[n][n];
4. Array Degree[n];
5. Array EigenVector[n];
6. Extract sentences from the sentence set S.
7. Calculate TF and IDF score.
8. for i from 1 to n do
9. for j from 1 to n do
10. SimilarityMatrix[i][j]=0;
11. for i from 1 to n do
12. for j from 1 to n do
13. if(i!=j)
14. SimilarityMatrix[i][j]=tfidfcosine(i,j);
15. end
16. end
17. end
18. for i from1 to n do
19. for j from 1 to n do
20. if SimilarityMatrix[i][j] > t then
21. SimilarityMatrix[i][j] = 1;
22. Degree[i] + +;
23. end
24. else
25. SimilarityMatrix[i][j] = 0;

```
26. end
27. end
28. end
29. for i from 1 to n do
30. for j from 1 to n do
31. if(Degree[i]!=0)
32. SimilarityMatrix[i][j] = SimilarityMatrix[i][j]/
        Degree[i];
33. end
34. else
35. SimilarityMatrix[i][j]=0;
36. end
37. Lexrank=LexRank(sentences,SimilarityMatrix,
        DampingRatio,Degree);
38. Score=lexrank.PowerMethod(errorfactor);
39. Select the sentences in decreasing order of Score.
40. End.
```

COMPARATIVE STUDY

A comparative study of the existing text summarisation methods is given below.

I. Comparative Study

Sr. No.	***Title***	***Overview***	***Advantages***	***Disadvantages***
1	Multi-document abstractive summarization using chunk-graph and recurrent neural network[4]	A Chunk Graph is a word-graph is constructed to organize all information in a sentence cluster. Beam search and character-level RNNLM is used to generate summaries from the Chunk Graph for each sentence cluster.	RNNLM is a better model to evaluate sentence linguistic quality than n-gram language model. Chunk Graph is able to filter lots of sentence paths in low linguistic quality	The algorithm design is complex.
2	K Nearest Neighbor for Text Summarization using Feature Similarity[5]	The task of text summarization is considered as a binary classification task where each paragraph or sentence is classified into the different groups. Efficient results are obtained using text classification and clustering.	Represents data items more compactly.	Many features are required for encoding texts. So it covers only a certain domain.
3	The Mixture of TextRank and Lexrank Techniques of Single Document Automatic Summarization Research in Tibetan[6]	The summarization is done combining the advantage of keyword processing based on TextRank and processing of the relationship between sentences based on LexRank algorithm.	Provides better accuracy than LexRank alone.	The research was performed centred around Tibetan documents.So it is not beneficial for summarising articles written in English.

4	A Survey on Abstractive Text Summarization[7]	Two abstractive summarisation methods are structured based approach and semantic based approach. This method summarises and deciphers the various methodologies, challenges and issues of extractive summarisation and make use of the semantics of the text to produce better results.	It produces highly cohesive, coherent, less redundant summary.The summary will be information rich.	It is difficult to summarise long text by employing abstractive method alone.
5	Text Summarisation Using Sentence-Level Semantic Graph Model	The relevance values between sentences are calculated using semantic analysis and the values are taken as weights of edges while sentences' values are scored by a variant of the traditional PageRank algorithm.	This method is feasible and effective.	This system works on an assumption that all the sentences are anaphora resolved

VI. CONCLUSION

In this paper, we proposed a new algorithm for long text summarisation by integrating extractive and abstractive text summarisation methods. The method combines the advantages of both abstractive and extractive summarisation methods. Idf-modified lexrank algorithm was used in the extractive phase and a seq2seq RNN model combined with attention mechanism was used in the abstractive phase. In order to generate multi-line summary and increase the accuracy of the generated summary, after preprocessing, clustering of similar sentences is done.

VII. ACKNOWLEDGMENT

This work has been done as a part of final year B.Tech Degree project work and we would like to take this opportunity to thank our institution , Govt. Model Engineering College, Thrikkakara.

VIII. REFERENCES

[1] Shuai Wang, Xiang Zhao, Bo Li, Bin Ge and Daquan Tang, "Integrating Extractive and Abstractive Models for Long Text Summarisation" , 2017 IEEE 6th International Congress on Big Data.

[2] Akshi Kumar Aditi Sharma, Sidhant Sharma and Shashwat Kashyap and Daquan Tang, "Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarisation" ,2017 International Conference on Computer, Communications and Electronics(Comptelix)Manipal University

[3] N. Moratanch and S. Chitrakala, "A Survey on Extractive Text Summarisation" , IEEE International Conference on Computer, Communication, and Signal Processing (ICCCSP-2017).

[4] Jianwei Niu, Huan Chen, Qingjuan Zhao, Limin Sun and Mohammed Atiquzzaman, "Multi-Document Abstractive Summarisation using Chunk-graph and Recurrent Neural Network" , IEEE ICC 2017 SAC Symposium Big Data Networking Track.

[5] Taeho Jo, "K Nearest Neighbor for Text Summarisation using Feature Similarity." ,2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan.

[6] Ailin Li, Tao jiang, Qingshuai Wang, Hongzhi Yu, "The Mixture of TextRank and LexRank Techniques of Single Document Automatic Summarisation Research in Tibetan", 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC).

[7] N. Moratanch,Dr. S. Chitrakala, "A Survey on Abstractive Text Summarisation" ,2016 International Conference on Circuit, Power and Computing Technologies [ICCPCT].

SHORT BIODATA OF ALL THE AUTHOR

Jayaraj Balagopal is currently pursuing B.Tech in Computer Science and Engineering from Govt. Model Engineering , Cochin , affiliated to Cochin University of science and Technology.

Cicci Maria Xavier is currently pursuing B.Tech in Computer Science and Engineering from Govt . Model Engineering , Cochin , affiliated to Cochin University of science and Technology.

Krishna Priya P is currently pursuing B.Tech in Computer Science and Engineering from Govt. Mode l Engineering , Cochin , affiliated to Cochin University of science and Technology.

Rahul V Reji is currently pursuing B.Tech in Computer Science and Engineering from Govt. Model Engineering , Cochin , affiliated to Cochin University of science and Technology.

Anjali S Has Done Her B.Tech In Computer Science And engineering from college of engineering. Adoor affiliate d to cochin university of science and technology and m.tech in computer science and engineering from amc college of engineering affiliated to visvesvaraya technological university. She has 1.7 years of industria l experience in company called isigma inc. She worked as an assistant professor in ise department of the oxford college of engineering, bangalore for 2.5 years and she is currently working as an assistance professor in cse department of govt. Model engineering college thrikkakara, kochi since 1.6 years.