Knowledge Graph and Deep Neural Network for Extractive Text Summarization by Utilizing Triples

Amit Vhatkar
Student / IIT Bomaby, India
asvhatkar@iitb.ac.in

Pushpak Bhattacharyya
Professor / IIT Bomaby, India
pb@iitb.ac.in

Kavi Arya
Professor / IIT Bombay, India
kavi@iitb.ac.in

Abstract

In our research work, we represent the content of the sentence in graphical form after extracting triples from the sentences. In this paper, we will discuss novel methods to generate an extractive summary by scoring the triples. Our work has also touched upon sequence-to-sequence encoding of the content of the sentence, to classify it as a summary or a non-summary sentence. Our findings help to decide the nature of the sentences forming the summary and the length of the system generated summary as compared to the length of the reference summary.

1 Introduction

Extractive summaries contain the most informative sentences from the input text. The ordered pair of Subject(S), Verb(V) and Object(O) i.e. [1]< S, V, O> represent the content of the sentence. We form a knowledge-graph(KG) by considering words in the triples. Our novel methods choose the informative sentences based on the count of frequencies calculated using generated KG. We also have implemented machine Learning(ML) and Deep Neural Network(DNN) based models. These models make use of the KG based features which try to capture information available. We are making use of dataset made available for FNS-2020 shared task by El-Haj et al. (2020). [2]We have used SpaCy library for extracting triples. We have used python implementation of Rouge package made available by [3]PyPI, which implements ROUGE described by Lin (2004).

2 Implemented Approaches

In general, we pose extractive summarization as a sentence classification and a triple classification task. We perform this classification using algorithms like SVM, SVR, Neural Network(NN) and Long Short-Term Memory(LSTM/DNN). This section describes our implemented approaches in details.

2.1 Labelling and Feature Extraction

	Summary Sentences	Non Summary Sentences	Total Sentences
Train Set	0.3M	2.6M	2.9M
Validation Set	0.051M	0.663M	0.714M

Table 1: Distribution of Summary, Non-Summary Sentences Extracted from Training and Validation Set

FNS-2020 Shared task training and validation dataset comes with up to seven gold summaries. All the sentences present in the gold summary are extractive in nature. All available gold summaries of the specific document are used for labelling the sentences in the given text.

[1]Referred to as triple

[2]https://spacy.io/

[3]https://pypi.org/project/rouge/

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 130–136
Barcelona, Spain (Online), December 12, 2020.

We have considered features as, *Position of the sentence*- each sentence is marked according to its position (i.e. 1-Introductory, 2-Concluding and 3-Explanatory), *Length*- number of words present in the sentence, *Thematic Words*- number of most [4]frequent words present in the sentence, *Indicator Words*- Count of words present in the available [5]synset(i.e. the group of synonymous words) of words 'conclusion' and 'summary', *Uppercase*- number of uppercase words present in the [6]sentence, *Important Word Feature*- It represents quotient of the available triple in the sentence to the total triples in the text file, and *KG File Feature*- It represents quotient of the available triples in the sentence represented in terms of [7]lookup frequencies to the number of total triples in the text file. We also have used pre-trained 100-dimensional GloVe(Pennington et al., 2014) embedding for DNN based approaches.

2.2 Triple Frequency-based Models(TFM)

Subject, verb and object(<S, V, O> i.e. triple) are main content words available in any sentence. Individual count of S, V and O present in the document fails to represent content available the sentence. To generate a content-aware extractive summary, TFM chooses sentences containing the highest scoring triples. Based on score computation, we have defined three different models.

2.2.1 Plain Frequency Model(PKG)

This is the simplistic approach to generate extractive summaries by making use of Triple Frequencies. This method fails to identify important sentences in the case of equal distribution of triples.

1. Extract all triples available in the text document and maintain it's count
2. Generate the score of the triple by considering its count
3. Generate extractive summary by selecting the sentences containing top-K triples

2.2.2 Updated Frequency Model(UKG)

PFM fails when the majority of the extracted triples gets an equal score. We tried to remove the equal scoring by considering frequencies of the subject and object present in the triples for scoring the triples.

1. Extract all triples available in the text document
2. Generate the score of the triple using the following formula,

$$triple\,score = Frequency\,of\,triple + frequency\,of\,subject + frequency\,of\,object \quad (1)$$

3. Generate extractive summary by selecting the sentences containing top-K triples

The formula 1, helps to break the uniformity of the scores occurring in the PFM by giving importance to the subjects and the objects available in the sentence.

2.2.3 Five Fold Cross Validation Model(FKG)

This approach considers two disjoint sets of documents to generate a score of the triples, 1- Train fold: used to extract and score the triple, 2- Test fold: used to generate the summaries and check the performance.

1. Extract and pre-compute frequencies of triples based on all documents present in the training folds and extract triples from the document present in test fold
2. Generate the score of the triple extracted from test document by considering its count which is computed (after considering all documents in remaining folds) in Step-1.

[4]Based on occurrence in the document
[5]We have used wordnet made available in nltk library for getting synset
[6]Excluding word 'I'(most commonly occurring uppercase word)
[7]Entire dataset is divided in five disjoint folds of which four-fold forms training set i.e. lookup

3. Generate extractive summary by selecting the sentences containing top-K triples

This method helps us to gain an insight over the presence of general sentences related to the topic or domain and the presence of the sentences specific to the document. Table 2 represents the extraction statistics of FNS-2020 training set. We have considered 2580 training documents for extraction.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
# Important Words	186505	186526	186249	188363	186631	186854
# Triples	758701	758657	760248	767476	761487	761313

Table 2: Fold wise Extraction Statistics of FNS-2020 Shared Task Training Set

2.3 Machine Learning based Summarization

We have implemented two machine learning-based approaches, 1-SVM, 2-SVR. Implementation of SVM for extractive summarization roughly follows the method used by Chali et al. (2009); where the task of extractive summarization is posed as a binary classification task. SVR for extractive summarization is based on the discussion by Li et al. (2007). SVR for extractive summarization assigns the score to the sentence. Summary sentences are scored as 1 and non-summary as 0. SVR approach predicts the score for sentences in the document and we select top-k scoring sentences as a summary sentences. SVM and SVR make use to the features mentioned in section 2.1. Both of the models are trained on 0.6M sentences with an equal mix of the classes.

2.4 Deep Neural Network based Approaches

Along with KG and ML-based approaches, we have implement DNN based approaches to generate extractive summaries by performing binary class classification.

2.4.1 Neural Network Model

We have trained a feed-forward neural network to classify a given sentence. Input layer consumes the features mentioned in the section 2.1. The model's architecture is the input layer followed by a dense layer with eight neurons followed by an output layer. *ReLU* and *Sigmoid* activation functions were used with *Binary Cross Entropy* as a loss function and *Adam* as an optimizer. Model Performs best when we set batch size as 32, Train-Validation split as 70-30% and train for 150 epochs.

2.4.2 S-LSTM for Extractive Summarization

We have trained LSTM models to classify the **sentence** as a summary and non-summary sentence. The encoder uses the entire sequence of the words present in the sentence to capture content information, for that we have used pre-trained 100-dimensional GloVe embedding which does not evolve during the training phase. The architecture is made up of an embedding layer followed by LSTM layer (with 2% dropout) followed by a softmax layer. Categorical cross-entropy(a generalized form of binary-cross entropy) with Adam optimization technique is used for training.

Models	Training Instances	Validation Instances	[8]LR in %	Epoch	Batch Size	[9]MSL
S-LSTM	0.48M Sentences	0.12M Sentences	1	5	32	35
T-LSTM	0.8M Triples	0.2M Triples	1	15	65	5

Table 3: Training Parameters Used to Train S-LSTM and T-LSTM Model

[8]Learning Rate
[9]Maximum Sequence Length

2.4.3 T-LSTM for Extractive Summarization

In this approach, words present in the **triples** were passed to the LSTM encoder, unlike S-LSTM where all the words in the sentence get passed. Based on the presence in the summary sentences, triples are marked as a summary or a non-summary triples. We have trained this model on positional embedding. All words in all triples are used to generate positional embedding. Embedding of the words presented in the triples are concatenated and is passed as input to the encoder of LSTM. The total number of words in the triples are less than the total number of words in all of the documents. Positional embedding tries to get an exact representation of the content of the sentence represented by the triple. Architecture details of this model remain the same as S-LSTM. Table 3 represents parameters used for S-LSTM and T-LSTM and table 4 represents extraction statistics related to T-LSTM model.

	Total Triples	Non Summary Triples	Summary Triples
Training Set	1389758	1158854	230904
Validation Set	352435	315418	37017

Table 4: Extraction Statistics of Triples from Sentences from FNS-2020 Shared Task Dataset

3 Results and Analysis

We have implemented eight different approaches while considering TextRank (Mihalcea and Tarau, 2004) as the baseline approach. In this section, we discuss the performance of the implemented models on the validation set followed by a performance on the test set.

3.1 Validation Set

The validation set of FNS-2020 Shared Task consists of 363 documents each having up to seven gold summaries. We are comparing results obtained over *single* gold summary of the specific document. System generated summaries were constrained to have 1000 words and our approaches select *first* 1000 words because, After segmenting given text in three parts each containing equal portions of the text we have found that in the training set 96%(i.e. 0.28M out of 0.29M) and in the validation set 95%(i.e. 49K out of 51K) of the summary sentences of the gold summaries are present in the first part of the text.

Model Name	*ROUGE-1 with respect to Single Full Length Gold Summary*			*ROUGE-1 with respect to Single Limited Length Gold Summary*		
	F	P	R	F	P	R
T-LSTM	0.3815	0.528	0.3162	***0.4888***	***0.4829***	***0.5013***
S-LSTM	***0.3911***	***0.5898***	*0.3238*	0.4288	0.4205	0.4434
NN	0.362	0.5527	0.3009	0.4471	0.4404	0.4604
SVM	*0.3114*	*0.435*	*0.2615*	*0.4368*	*0.4292*	*0.451*
SVR	0.2883	0.4045	0.2423	0.3665	0.3532	0.3875
PKG	*0.2891*	0.2546	*0.3768*	*0.426*	*0.4489*	*0.4126*
UKG	0.2689	*0.3837*	0.2213	0.3891	0.3848	0.3975
FKG	0.1933	0.3049	0.151	0.3413	0.3364	0.3498
[10]TextRank	0.2886	0.2535	**0.3778**	0.4244	0.4122	0.4438

Table 5: ROUGE-1 Score Comparison of All Implemented Models concerning Single Full Length Gold Summary and Single Limited Length Gold Summary

Table 5 represents ROUGE-1 score of all implemented models when reference gold summary is allowed to contain all of its text(i.e. Full Length) and when it is allowed to contain first 1000 words(i.e.

[10]Our Baseline Model

Limited Length) of its text. The ROUGE score values in the table are averaged over the averaged(over 3 runs) ROUGE score of all documents in the validation set. In the result table, overall highest score are **bold-faced** while highest score among specific category is *italicized*.

Considering the full-length gold summary S-LSTM performs the best amongst all approaches. In the setting of limited length gold summary T-LSTM approach performs best. All approaches perform best in the limited length gold summary setting. Therefore, to obtain better results, length of system generated summary should be equal to the length of the reference summary.

Even after being rule-based models, TFM models have performed comparably well. Nature of text causes UKG to give an equal score to the triples affecting its performance. In the FKG model, generic triples get a higher score as the effect of considering all documents in training fold. This leads to a summary containing generic sentences. However, as PKG and UKG performance better than FKG, the summary should contain sentences specific to the document

3.2 Test Set

We have generated summaries of the 500 documents present in the test set using our [11]NN, S-LSTM and an SVM approach. Gold summaries of the documents in the test set are used by the organizers as a reference summaries to compute the results. The ROUGE values in the table 6 are published by the organizers of the shared task. Organizers also had computed the ROUGE scores of their baseline approaches(as mentioned in table 6) employing SUMM-TL-MUSE, LEXRANK-SUMMARY, SUMM-BL-POLY, TEXTRANK-SUMMARY.

Model Name	ROUGE-1 F	ROUGE-2 F	ROUGE-L F	ROUGE-SU F
[12]Best Performing	0.466	0.306	0.456	0.318
NN	0.445	0.246	0.318	0.242
S-LSTM	0.438	0.243	0.317	0.245
SVM	0.438	0.247	0.312	0.248
SUMM-TL-MUSE	0.433	0.234	0.407	0.253
LEXRANK-SUMMARY	0.264	0.12	0.218	0.14
SUMM-BL-POLY	0.274	0.105	0.205	0.135
TEXTRANK-SUMMARY	0.172	0.07	0.206	0.079

Table 6: ROUGE Score Comparison of KG-based Approaches with Top Scoring Approaches and Baseline Approaches, on Test Set, Computed by the Organizers using the Gold Summary

When compared on ROUGE-1, our NN based approach is among top-5 while SVM and S-LSTM approaches have secured 9th and 10th position respectively. SVM, NN and S-LSTM ranked 10th, 11th and 12th respectively on ROUGE-2 metric. Our approaches perform quite well as compare to the baseline approaches. No one approach outperformed the others in all ROUGE metrics.

4 Conclusion

We have successfully generated extractive summaries using our novel methods of triple scoring which are based on KG generated by the words in the triples. We have also proposed novel DNN based approaches for extractive summarization, where summarization is carried by performing binary classification after sequence-to-sequence encoding(either sentence or triples) content present in the input text. From our discussion in section 3.1, we can conclude that the summary should contain sentences specific to the document. We have seen that, in order to get better results, length of system generated summary should be equal to the length of the reference summary. We also have seen that KG-based Triple Frequency models perform comparably well than baseline models and possess scope of the improvement.

[11]Only up to 3 summaries per document are allowed by organizers of FNS-Shared Task

[12]Different systems performed well on different ROUGE metric

References

Yllias Chali, Sadid A Hasan, and Shafiq R Joty. 2009. A svm-based ensemble approach to multi-document summarization. In *Canadian Conference on Artificial Intelligence*, pages 199–202. Springer.

Mahmoud El-Haj, Ahmed AbuRa'ed, Nikiforos Pittaras, and George Giannakopoulos. 2020. The Financial Narrative Summarisation Shared Task (FNS 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020*, Barcelona, Spain.

Sujian Li, You Ouyang, Wei Wang, and Bin Sun. 2007. Multi-document summarization using support vector regression. In *Proceedings of DUC*. Citeseer.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

A Web Service

We have created web service for end-users to summarize their text either by uploading a file or by type-in text functionality. Figure 1 depicts the landing page of web service where users can see the summarized text along with the [13]KG that gets generated and used for summarizing the input text. The web portal also mentions the percentage reduction in terms of the number of words and ROUGE score calculated using TextRank as a reference summary. Web service displays PKG, UKG summaries along with TextRank summary. It also has provision to host a dataset and it currently hosts part of CNN-Daily mail dataset. Currently, the web service is running in a private domain.

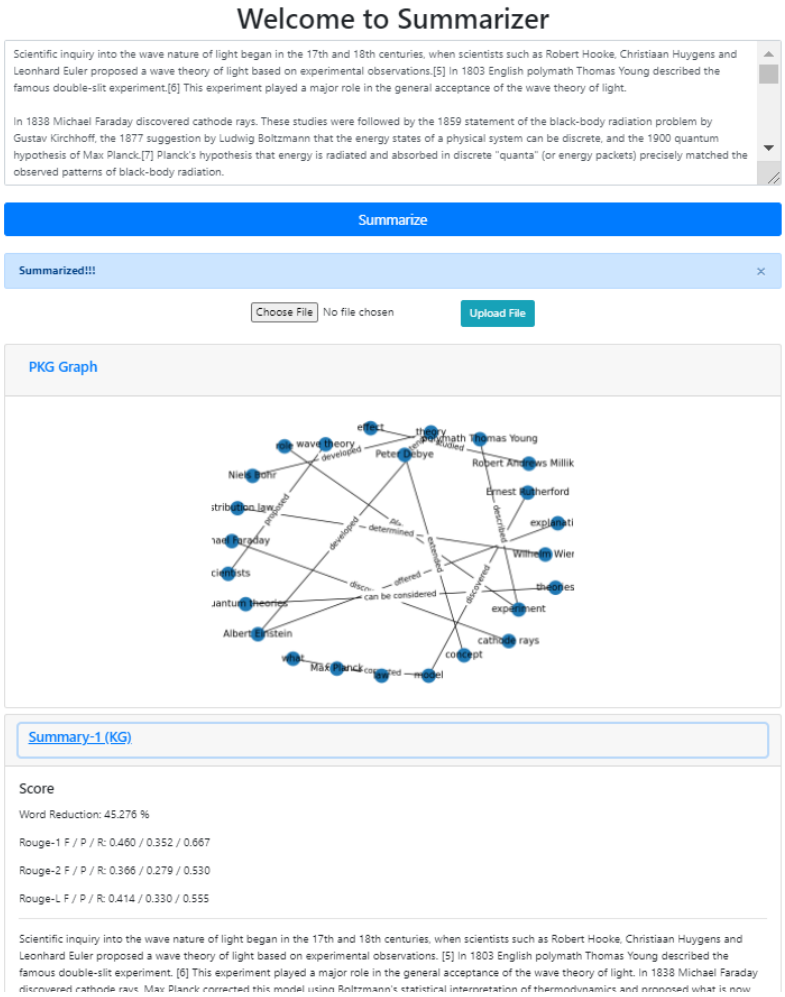

Figure 1: Web Service for Extractive Text Summarization by Triple Frequency-based Approaches

[13]Graph shown does not capture the notion of frequency