Catch Phrase Extraction From Legal Documents Using Deep Neural Network

Sourav Das
Indian Institute of Engineering Science and Technology,Shibpur
Howrah, West Bengal
sourav.dd2015@cs.iiests.ac.in

Ranojoy Barua
Indian Institute of Engineering Science and Technology,Shibpur
Howrah, West Bengal
baruaranojoy1@gmail.com

ABSTRACT

This paper is based on finding and extracting important key phrases (catchphrase) from a document from which the the document can be summarized. This is important as this will reduce time consumption in summarization of documents. This work is realized with the help of deep neural network to train an model for recognizing such important key phrases based on various calculated parameters.

1 INTRODUCTION

The legal system depends on citation of previous cases which allows better judgment but with a huge number of cases to study, the search for suitable cases becomes difficult. This problem can divided into two parts. First is key phrase extraction and secondly finding suitable matches based on the key phrases found in the document. The main motive of the paper is to find efficient way for key phrase extraction. In this approach, deep neural network provides an elegant way to extract catchphrases, which then can be used to take reference from while searching for previous similar cases.The features used include grammar, Tf-idf, position in a document etc. Thus extracting most important key words in the document, which can then be used further as requirement. This approach can be used to minimize the required human effort. The words are further divided based on their weights to determine it's importance in the document.

2 METHOD

2.1 Data
All files are legal documents recorded by the Supreme Court of India. A total of 400 documents are used during the course of the experiment, out of which 100 are having gold standard catchphrase (catchphrase by human) which are used for training, other 300 are used to generate output.

2.2 Procedure
In this experiment for each file a set of potential meaningful phrases are created and then are classified using deep neural network. Steps involved
1. Preprocessing
2. Create potential meaningful phrases based on common grammar of phrases
3. Feature selection
4. Label the vectors
5. Classification
6. Training the model

2.2.1 Preprocessing
Read the whole file remove stop words, punctuations, non-ASCII characters and numbers. Store the modified file for future use.

2.2.2 Phrase generation
Generate all potential meaningful phrase based on common grammars of different phrase.

2.2.3 Feature selection
Different features are extracted on these phrases
1. Total summation of the Tf-idf value of all the words in the phrase. Tf-idf value is calculated to find the important words from the train documents.

Note : All the gold standard catchwords are combined and will be referred as super gold standard

2. Find all the different type of part of speech present present in super gold standard and create a standard vector with respect to which find the different part of speech in each phrase also keep a trace number of times each part of speech occurs.

3. Now multiply each part of speech in the vector created with its particular weight calculated from super gold standard

Weight = (number of occurances of unique POS)/(total number of phrase in UGS) * 100

4. Find if the phrase exactly matches with any phrase from super gold standard. If exact match is found find number of times exact match occurs.

5. Find the number of times the unique words word of the phrase matches with an word in the super gold standard file also keep a track of how many individual words of the phrase found a match in the super gold standard.

Now combine all these features to create a large feature vector.

Note : Not all the feature mentioned are used as it will lead to a large feature vector and some features cover the other features so only a set of these features is used as the final feature vector

2.2.4 Labeling

We intend to apply supervised learning but presently we have an feature vector without any label. So, we need apply labels to apply supervised learning. We label the data in two class i.e., phrase eligible for catchphrase and not eligible for catch phrase.
So the criteria for labeling are
1. Should have Tf-idf value greater than 0.0
2. The phrase should hold one part of speech belonging to super gold standard.
3. The phrase should have at least one word matching with super gold standard.
4. May or may not have exact match with a phrase with super gold standard.

Conditions for labeling
1. If all conditions satisfies labeled as valid.
2. If only condition-4 satisfies labeled as valid.
3. If all other condition satisfies other than condition-4 it is valid.
4. Else it will be not valid.

2.2.5 Classification
For classification purpose we have used deep neural network, the network have three layer deep. It have two internal layer each having 28 nodes and an output layer having 2 nodes.
Architecture of each layer

Output = input . (weight) + bias

Then sigmoid function is applied to squash all the values between 0 and 1.
The model is trained for 200 epochs. During training gradient descent optimizer is used to optimize the result. Softmax layer is applied on the output generated by the output layer to obtain the final result.

3 RESULT

Accuracy of the model is calculated by divided the 100 available samples in a set 70-30. 70 are used for training and rest for testing and the accuracy ranges from (76-82) percentage.

Final result obtained from the evaluation produces
1. Mean R precision : 0.0262223166667
2. Mean Precision at 10 : 0.0246666666667
3.Mean Precision at 20 :0.0208333333333
4. Mean Recall at 100 : 0.0868031271116
5. Mean Average Precision : 0.0618723522608
6. Overall Recall : 0.160995639731

Some of the way by which result can be improved by increasing the number of epochs, getting more features, combine the result of multiple run, using Adam Optimizer instead of Gradient Descent optimizer.

4 CONCLUSION

In this work we have developed a framework where if the network is trained by using previous cases then it will produce catchphrase which in turn will help to find precedent much faster than human can do.

REFERENCES

[1] A. Mandal, K. Ghosh, A. Bhattacharya, A. Pal and S. Ghosh. Overview of the FIRE 2017 track: Information Retrieval from Legal Documents (IRLeD). *The LaTeX Companion.* In Working notes of FIRE 2017 Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017, CEUR Workshop Proceedings. CEUR-WS.org, 2017.

[2] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. *The LaTeX Companion.* OReilly Media Inc.

[3] Martín Abadi et al. TensorFlow *The LaTeX Companion.* Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.