INSIGHT EXTRACTION FROM REGULATORY DOCUMENTS USING TEXT SUMMARIZATION TECHNIQUES

by

Kshirabdhi Tanaya Patel, Bachelor of Technology, SOA University, India, 2015

A Major Research Project
presented to Ryerson University
in partial fulfillment of the requirements for the degree of

Master of Science
in the Program of
Data Science and Analytics

Toronto, Ontario, Canada, 2020

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A MAJOR RESEARCH PROJECT (MRP)

I hereby declare that I am the sole author of this Major Research Paper. This is a true copy of the MRP, including any required final revisions.

I authorize Ryerson University to lend this MRP to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my MRP may be made electronically available to the public.

Kshirabdhi Tanaya Patel

INSIGHT EXTRACTION FROM REGULATORY DOCUMENTS USING TEXT SUMMARIZATION TECHNIQUES

Kshirabdhi Tanaya Patel

Master of Science 2019

Data Science and Analytics

Ryerson University

ABSTRACT

Legal documents are hard to understand and generally requires special knowledge to be able understand and gain information from it. In such situation it is hard to find and follow acts and regulations which are suitable for our business, jobs or other work. Sometimes hiring a person who understands it and can helps us costs hundreds of dollars. So, in current time there is a need of some technology which can help us to overcome these problems and recommend us a list of acts and regulations which are suitable to us and also provide summary which can make us understand those legal texts. To address the discussed situation here we are developing a NLP framework to automatically extract relevant documents as per the user's requirements and give an summary report of the regulations. The dataset which was used here is Canadian Government Regulation and Acts. This dataset was made public for the use of data science community in the year of 2018 by Canadian Government.

Key words:

Legal Text, Acts, Regulations, Text Summarization, Convolutional Neural Network, Semantic analysis, Word Embedding, TF-IDF.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Saman Hassanzadeh Amin from the Department of Mechanical and Industrial Engineering at Ryerson. Throughout my graduation project, he was always there whenever I needed help. His guidance in research and writing thesis was so much help and my completion of the project would not have possible without him. I would not have found a better supervisor for this project work. I am glad that he agreed to accept my project proposal. Through this project, he provided me an opportunity to look at the real-world data and provided continuous guidance to process and gain insight from the data.

Once again Thank you very much Dr. Saman.

Table of Contents

1. INTRODUCTION

This document provides details of our research on how the famous Convolutional Neural Network can be used for legal text summarization task. We will also address the problems we faced and the solutions found after numerous experiments. We will start by describing the data and later give the detail of exploratory data analysis steps, model building, and result extraction steps. To understand this paper basic knowledge of Natural Language Processing and Deep Learning are required.

2. BACKGROUND

For the major research project, we choose to use legal text data because of the less research done in this field, and this field is challenging especially because of very little publicly available data for experiments. Though the legal text is commonly written in the language we speak and read, still common people not having a background in the legal domain find it difficult to understand. To make people's life little easier while following government Acts and Regulations, we are trying to come up with a model which will retrieve and summarize the Acts and regulations relevant to a user. For this project, we will be using Canadian Acts and Regulation data provided by the Canadian government. This includes 2600 regulations which come under 800 Acts. For the project, we will be studying document similarity, information retrieval, semantic similarities for document extraction.

For Convolutional Neural Network implementation, our study attempts to replicate the work of Y. Zhang et al. (2016) and for document similarity and document extraction we have created the model of our own on top of CNN.

3. LITERATURE REVIEW

The following section provides an overview of the study done before starting this project. These papers have greatly helped us to troubleshoot all the problems we faced while doing the project.

In an article, Buitelaar et al. (2017) studies different topic modeling techniques for information retrieval and ranking the documents according to the relevance of query topics which is closely related to identifying regulations related to the interest of the user. Although topic models have wide applications in the field of Natural language processing, they do have limitations like topic proportions, which are relevant to individual documents and fail to find similarities between documents representing the same topic. P.Xing and Xie (2013) in their paper described a new way of integrating topic modeling and document clustering to get the clusters of similar documents. They projected the vocabulary of the documents to the topic dimensions and used topic score as document features to find the documents clusters. The results of the model were quite impressive as the clusters were successful to identify coherent clusters but had problems identifying regulations belonging to multiple clusters due to hard clustering. Replacing hard clustering with hierarchical clustering can eliminate the problem.

The papers so far discussed were built on the bag of words representation of the vocabularies and this type of representation often fails to capture semantic and syntactic properties of the words. In an article, Chen et al. (2017) proposed a probabilistic model combining word embedding and LDA (WE-LDA), which was powerful enough to capture the document level semantic of the word and hence producing a more coherent topic in the documents. For word embedding, they used skip-gram model and this also helped to project words to a lower-dimensional space enabling dimensional reduction and increasing computational efficiency of the topic extraction. When we have thousands of regulations we require a model that can output relevant regulations in almost real-time and for the purpose of multi-document extraction using embedded topic model, this article is a great help.

Document summarization is a challenging and demanding task. This summarizes large documents into small text that can help users gain access to useful data over a short period. One of the most interesting research done by H.P. and Luhan (1958) says text summarization has a high impact on the way we select our words and perform text cleaning like word stemming, stop word removal, and lemmatization. Also through experiments on huge text corpus, the conclusion they extracted was pretty surprising and also exciting. They conclude that around 85% of the data, the main subject was cited in the first few lines of the paragraphs within the documents, and focusing a little bit on these proportions can greatly improve our regulation extraction process.

After grouping documents and learning features the last step is to extracting document wise summary. Numerous projects have been done in this domain but all the methods perform well when there is an availability of plenty of data. For best document summarization models so far discovered require the availability of a large volume of data with document summary pair for training the models. However, In the legal domain like Act, Regulations, etc. no such publicly available data exist to date. Li and Manor (2019) presented an unsupervised approach of text summarization which eliminated the disadvantages of the requirement of handcrafted summaries. The most interesting and relevant part of the article for me was: in their work they have proposed the methodologies to find data that can be utilized for building the legal text summarization model without using legal text. They performed their experiment on text corpus like Wikipedia, CNN/Daily Mail, and achieved an F1 score of 0.892. Akcayol et al. (2019) represented how handcrafted parameters like term frequency, inverse sentence frequency, sentence position, sentence length, sentence–sentence similarity, phrases of the sentence, proper nouns, n-gram co-occurrence, and length of the document can be utilized to build a text summarization model by overcoming the data availability issue.

4. EXPLORATORY DATA ANALYSIS

In this section we will be discussing the detail of data and insights got from the data by exploring it, which are later used for modelling.

4.1 Datasets

Our primary dataset is download from justice Canada FTP server and they have the files in XML formats. Before doing any analysis it is important for us to build a XML parser which can extract the content of the file in simple text format. So we created a xml parsing function in python to retrieve the data in the form of Data Frame.

we have used 2 more datasets to support our work, which are DUC and CNN/Daily mail. The Document Understanding Conference (DUC) dataset has become a standard in text summarization task to achieve state of art performance. It is composed by news articles in English from the National Institute of Standards and Technology's Text Retrieval Conferences (NIST TREC). The CNN/Daily mail dataset contains online news articles (781 tokens on average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on average). The processed version contains 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs.

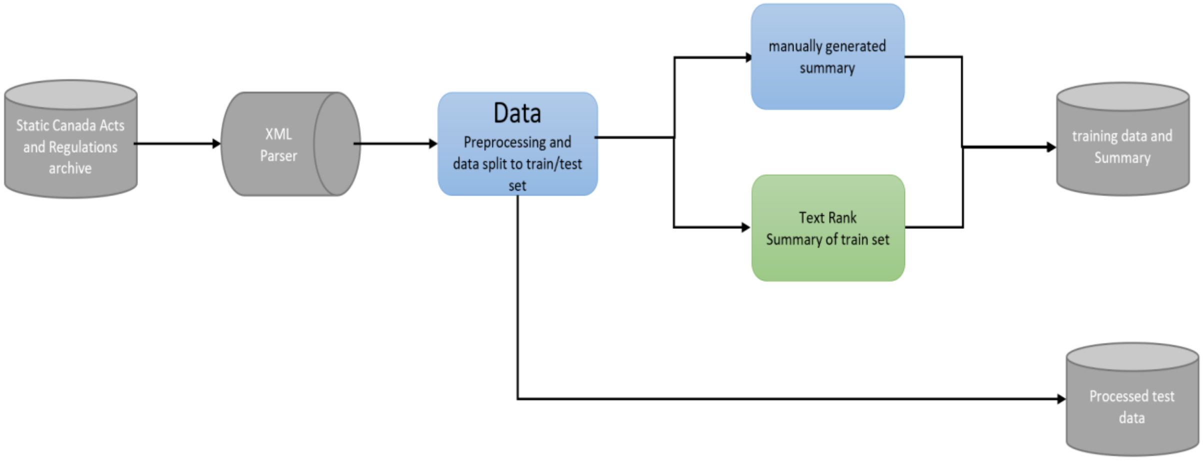

Static Canada Acts and Regulations archive
XML Parser
Data
Preprocessing and data split to train/test set
manually generated summary
Text Rank Summary of train set
training data and Summary
Processed test data

Fig. 1: The flow diagram shows the data extraction pipeline.

4.2 Data Extraction

After parsing the data, we ended up getting 6 columns which are long title, registration year, consolidation year, regnal years, xrefxternal, xrefinternal, and content. Some of the columns have more than 90% of missing data and also not relevant for our analysis. So we eliminated those fields and the selected columns are; content, xrefxternal, and longtitle.

Fields Description:

1. Content – Actual regulation text.
2. Xrefxternal – This tells about the enabling act (Act under which the regulation is enabled).
3. Long title – The actual title of the regulation

In the regulation data frame, each row of data represents a document which is also a regulation in our case, and the data frame “content” column is a text corpus/regulation corpus for further analysis. The DUC and CNN/Daily mail datasets are preprocessed and already available in the TensorFlow environment. So we did not require any extra efforts for that. we used TensorFlow command to import them as pandas data frame.

4.3 Textual Content Analysis

The regulatory corpus contains 2062 documents and each of them describes a regulation. Each regulation has a different length. Few of them are too small and few are too big. NLP models are highly sensitive to the size of the text corpus and also to the feature dimension. So we decided to have a quick look at the length of each document to decide the maximum length of the term-document matrix.

Fig. 2 clearly explains that each document has an average of 80 unique words and Fig. 3 explains that the total number of words in documents ranges from a few hundred to a few thousand. Form these graphs, it is clear that we have a limited vocabulary, and words are repeated again and again in the documents. The total number of unique words in the regulation corpus are 35552 and total numbers of words including repeated words are 1844666.

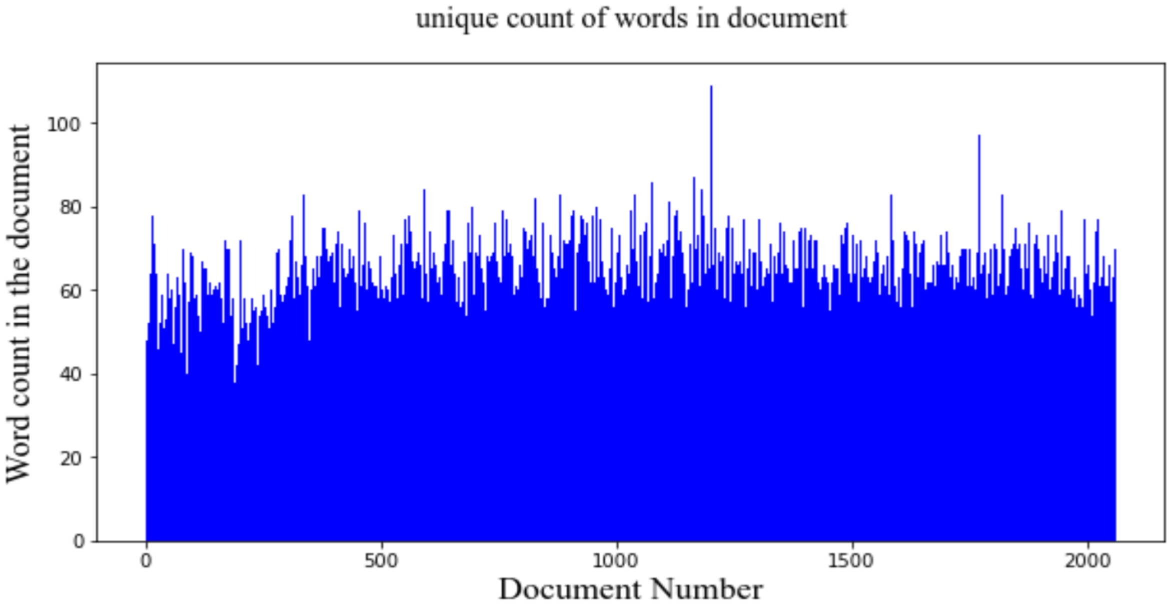


Fig. 2: The bar chart shows the number of unique words each document have.

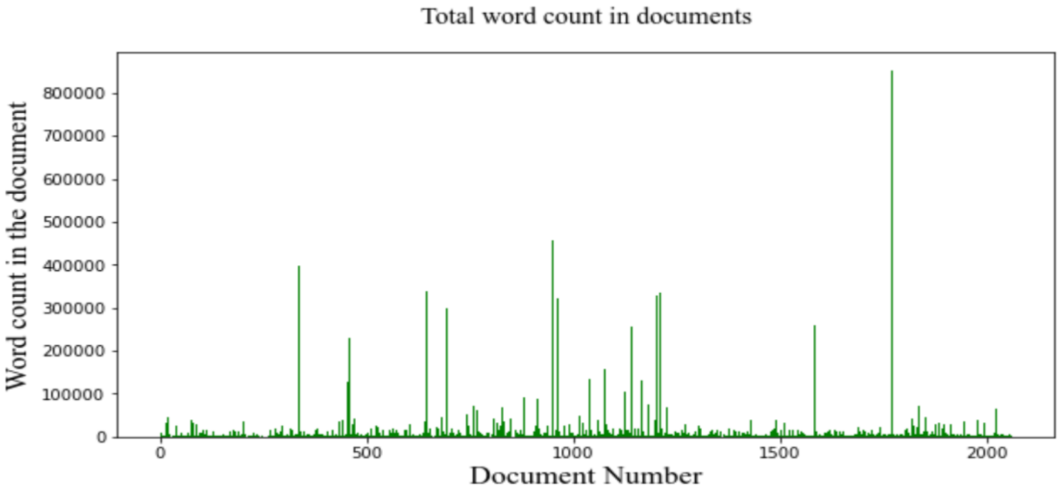


Fig. 3: The bar chart shows the total words count each document.

After looking at the word distribution over the documents, let's move on to the sentence level analysis. This analysis helps to choose the word level and sentence level embedding dimensions for preparing data for neural networks. One of the biggest strengths and also weaknesses of neural networks is the data. We require a large number of labeled data to train a model because of the number of parameters of network. In our case, we do not have both the large volume of data and labels. So we have to come up with similar datasets having the same kind of sentence distribution as ours. DUC and CNN/Daily mail are two famous datasets for text summarization tasks, so we will be doing some sentence-level analysis of these datasets to know whether we can use these datasets for our model building or not.

From Fig. 4 we can see that most of the regulations have sentences in the range of 0 to 250. To get a clearer idea about the sentences, we can have a look at the summary statistic of the number of sentences shown in Fig. 5.

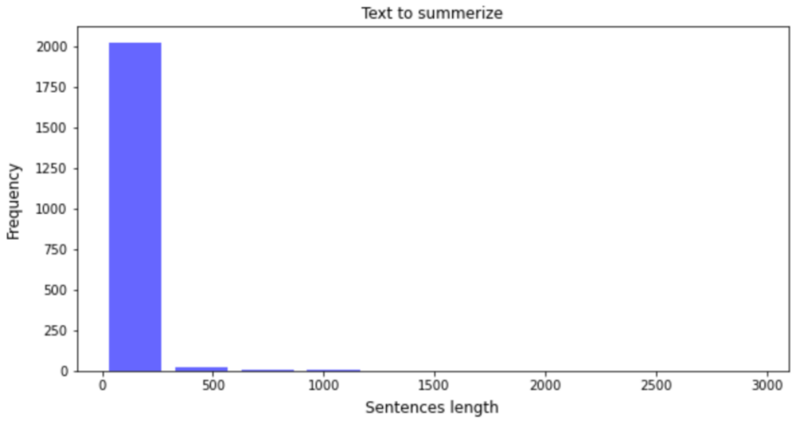
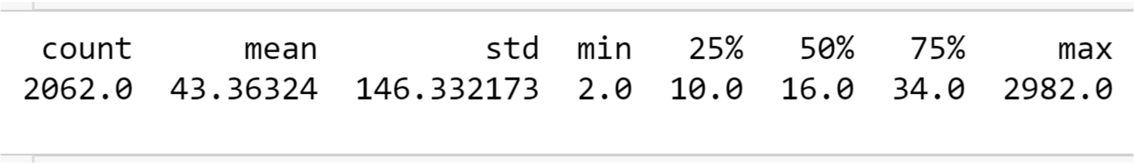


Fig. 4: The histogram plot shows the sentence count in documents.



```
 count       mean          std  min   25%   50%   75%     max
2062.0   43.36324   146.332173  2.0  10.0  16.0  34.0  2982.0
```

Fig. 5: Summery statistic of length of documents

Now we will check the average number of sentences the DUC and CNN/Daily mail datasets have in each document. For CNN/Daily mail dataset maximum documents have the sentence length ranging between 5 to 90 and summary length between 2 to 6 (fig 6). We can see the same type of distribution for DUC dataset but here the summary length has little broader distribution ranging between 3 to 10 (fig 7). From the summary statistic (fig 4) we can see the average number of sentences in a regulation text is ~44. So from our analysis, we can conclude that using these datasets we can come up with a base model which can be used for transfer learning and fine-tuning.

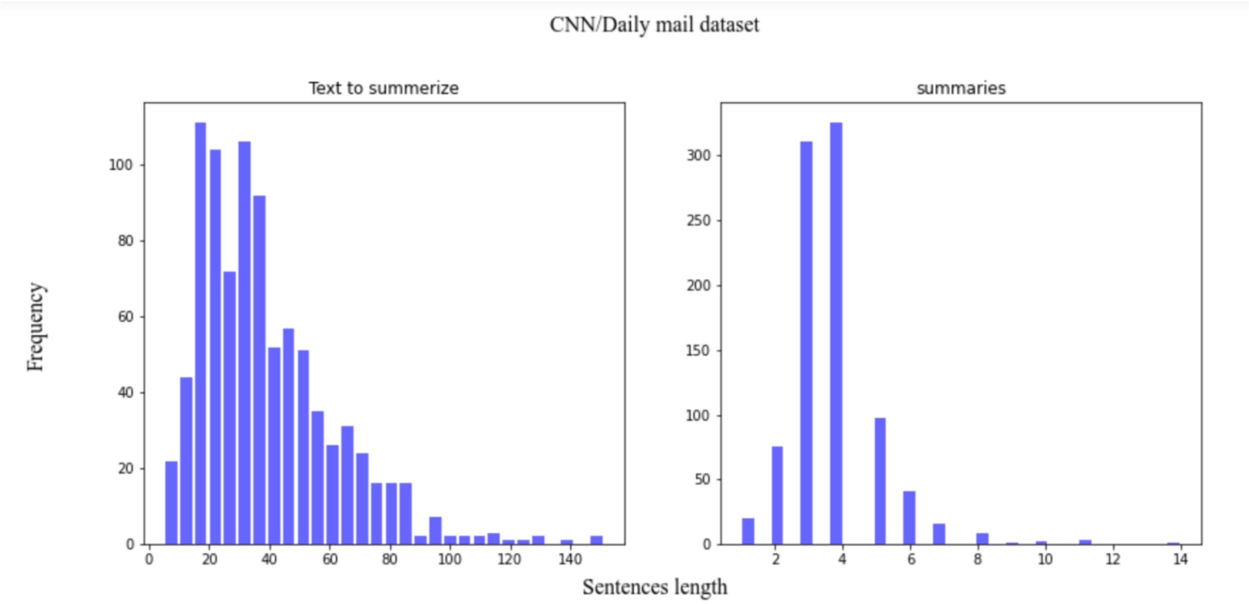


Fig. 6: Histogram plot of average number of sentences in CNN/Daily mail dataset

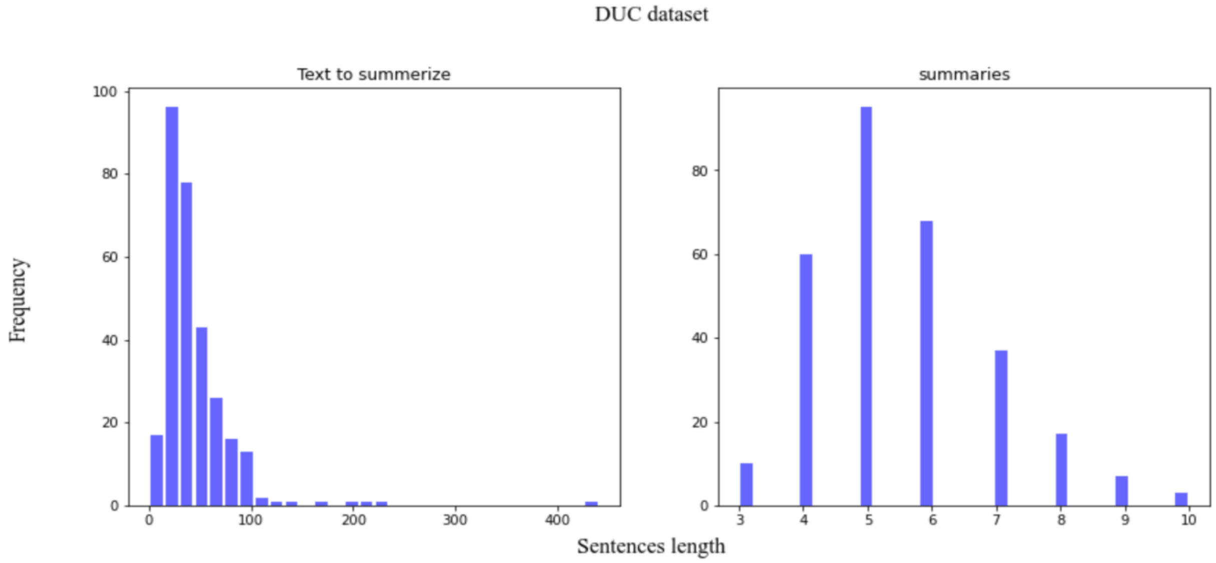


Fig. 7: Histogram plot of average number of sentences in DUC dataset

4.4 Text Preprocessing:

Sentence level text cleaning:

1. Part of speech Tagging : We have tagged each sentence with its grammatical form like noun, verb, adjective, etc. and removed words that does not convey any meaning but help to make a sentence complete and meaningful. Such examples are is, am, are, etc.
2. Tokenization and lemmatization : We have performed 2 level tokenization. Document to sentence and sentence to word to get individual words for further analysis. Each and every word are lemmatized to bring it back to its base form.
3. Stop words Removal : Special characters like semicolon, brackets, parenthesis, etc. are removed as they don’t add any value to the models.

Table 1 shows an example of text processing steps performed before proceeding to word to embedding.

Original Text: In these Regulations, aircraft means any machine used or designed for navigation of the air but does not include a machine designed to derive support in the atmosphere from reaction against the earth’s surface of air expelled from the machine
Processed Text: ['Regulations', 'aircraft', 'mean', 'machine', 'use', 'design', 'navigation', 'air', 'do', 'not', 'include', 'machine', 'design', 'derive', 'support', 'atmosphere', 'reaction', 'earth', 'surface', 'air', 'expel', 'machine']

Table 1: Shows the original text vs processed text

For word embeddings, we have used Google's pre-trained word2vec, which has embeddings of 3 million words at k = 300 latent dimensions. After word embeddings, the embeddings are stacked to form sentence embedding and again stacked to form a tensor of documents. For target variable salience score was calculated. Zhang et al. (2016) define a salience score, in order to rank sentences according to their similarity with the ground truth summary. This makes sense for extractive summarization since the more similar a sentence is with the summary, the more likely it is that the summary contains it.

$$y_i = \alpha \cdot R_1 + (1 - \alpha) \cdot R_2 \in \mathbb{R}$$

R1 = ROUGE -1 score

R2= ROUGE-2 score

α = coefficient to perform weighted average

5. METHODOLOGY AND EXPERIMENTS

5.1 Aim of study

This study has two main goals. The first is to train a Convolutional Neural Network for text summarization task that could be used for transfer learning and fine-tuning. We need this pretrained network because of having a very small volume of labeled data. The regulation data we have was downloaded directly from the Canadian government website and no summarized view of it was available; to train a CNN, we require a data set comprised of independent and response variables and manually generating summaries of more than 1000 documents is not possible. The second is to build a hybrid model that will incorporate machine learning algorithm with neural network to extract summaries of semantically similar documents.

5.2 Independ, dependent variables and formatting

For the task of text summarization, we are training a Convolutional Neural Network with one max pooling layer, one dropout layer, and 2 fully connected layers. As the models are built on Keras which is built on top of TensorFlow, We had to convert the inputs into tensors. Below are the steps for converting data into tensors.

- Let us assume the input is X then the form of it would be X ∈ RS ×N×K , where S is the number of sentences, N is length of sentences which is a hyperparameter to decide, K is word embedding latent dimension which is 300 in our case.
- The input tensor shape will be in NHWC data format. (number of sentences, sentence length, word embedding width, Convolutional channel). In our case the channel is 1.
- The output is the salience score and is a NumPy array of length equal to number of sentences.

5.3 Model Building/Experimental Design

5.3.1 Network Architecture

- *1D Convolutional layer :* In order to mimic the process of RNN, the filters need to convolve through multiple words in the sentence. So we decided to have one 1D convolution with 100 filters. The window size of the filter is 5✕300, Since our word embedding dimension is 300 it does not make sense to split it, which made our filter width to 300. The activation function used here is RELU.
- *Max pooling layer :* Max pooling layer was added to keep the important features and remove the rest.
- *Dropout layer :* To avoid overfitting of the model a dropout layer was added, which help to regularize the model
- *Fully connected layer :* Two dense layers were added on top with sigmoid activation function to produce a scalar output.
- *Loss function and optimizer :* We used binary cross entropy as loss function to minimizing training error. The optimizer used was adadelta since it helped to converge faster.

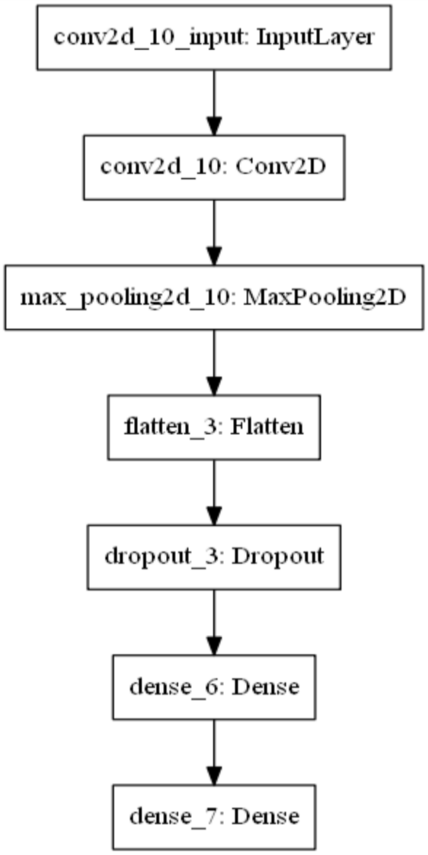


Fig. 8: Shows the CNN network architecture for text summarization task.

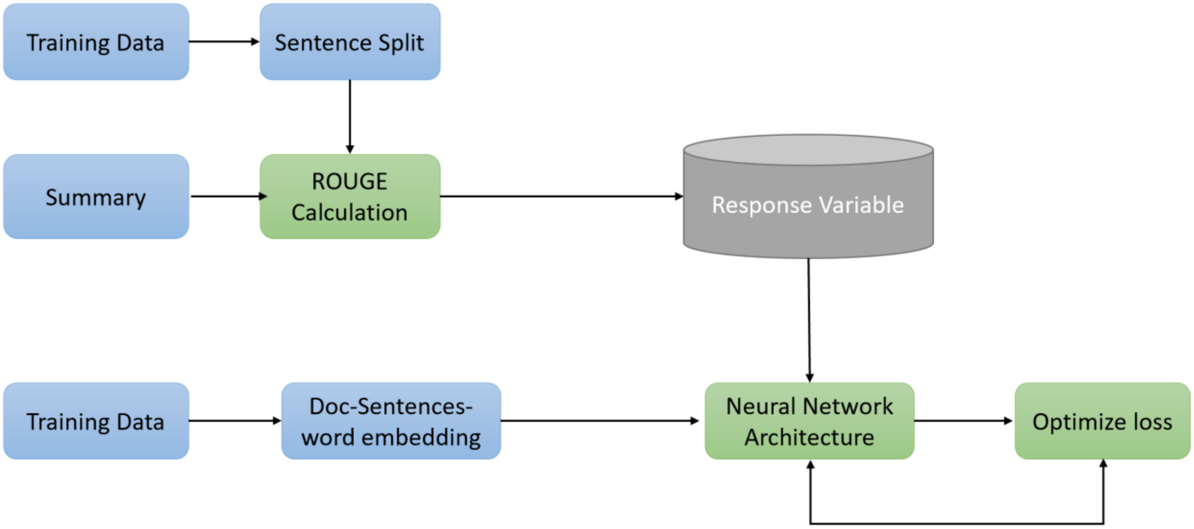


Fig. 9: The data flow diagram with training architecture.

5.3.2 Model Training and Validation

The model was trained in 3 phases for both the dataset. At each stage, performance of the model was observed and modified for better results.

- The model was trained with DUC 2001 and CNN/Daily mail datasets where the batch size is 256 and the number of iterations performed is 2000. The trained model was saved and utilized to fit the preprocessed legal documents. The ROUGE score was calculated between the original summary and model generated summary as the performance measure.
- In the second phase, we tried transfer learning. The first layer of the neural network was freeze and rest of the layers were trained with legal document dataset for 50 epochs and model performance was observed
- In the third phase, the model was fine-tuned with legal documents dataset and the performance was observed.

5.4 Inference from the model

After experimenting with varieties of models, the best model was chosen by looking at the ROUGE score between the original summary and the model generated summary. For the best performance of the models, all the documents having maximum number of sentences greater than 250 were eliminated because most of the documents in the training dataset were having sentence length less than 250.

For inference, the following steps were performed.

- Input the keywords to the document database. The TF-IDF score between the words and documents are precalculated and stored in the database. In our case, it is a data frame instead of database where the words are the data frame indexes and each column refers to a document/regulation and the cell value corresponds to the TF-IDF score.

- When a matching word is found, it extracts the documents with the best TF-IDF score
- The documents are then processed to fit the neural network and after that, the summaries of the corresponding documents are generated.

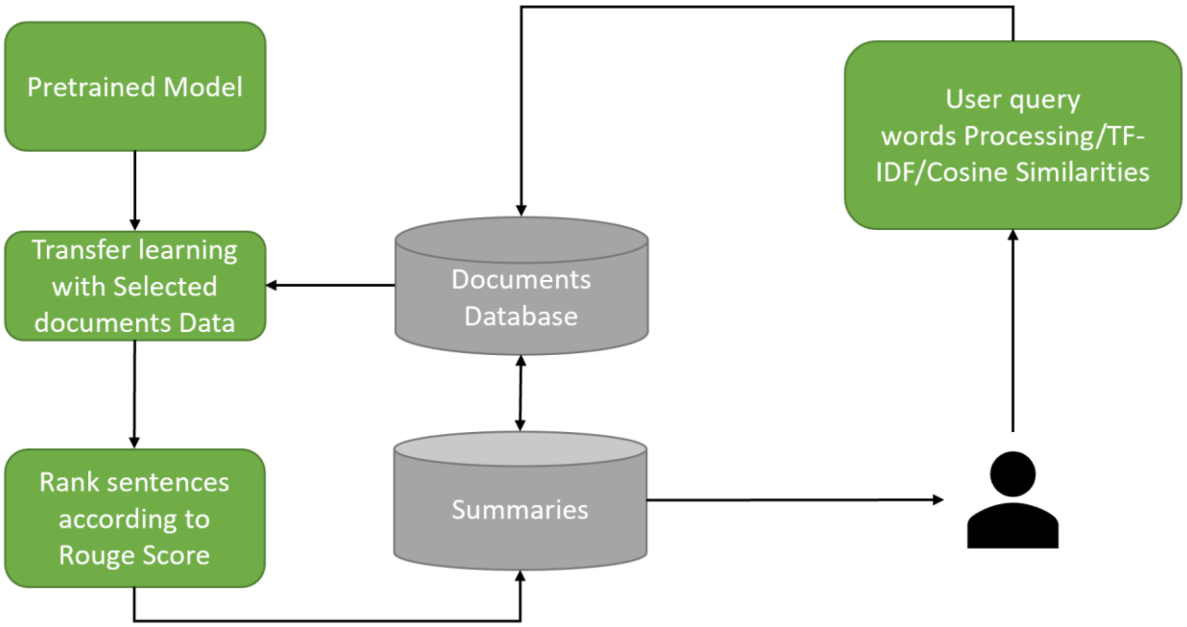

Pretrained Model
User query words Processing/TF-IDF/Cosine Similarities
Transfer learning with Selected documents Data
Documents Database
Rank sentences according to Rouge Score
Summaries

Fig. 10: The inference network the result/summary extraction.

6. RESULT AND DISCUSSION

From the various stage of the project, we got various type of results and insights which further helped us to plan our next step. We will discuss all the results step by steps in below paragraph.

6.1 Exploratory Data Analysis Results

- The exploratory data analysis of the three datasets provided some useful facts which greatly helped to decide our neural network dimensions. For example, While tokenizing, we found out that, we have a very limited vocabulary which are used very frequently. This helped us to decide whether to use bag of word representation of words or word embedding. Since repeated use of a word can enhance its weight and introduce bias to the model, we decided to use word2vec embedding representation of words.

- While analyzing document length we found maximum documents have number of sentences less than 250. So while creating embedding for sentences we decided to keep the document embedding shape 250*300 and eliminate documents having number of sentences greater than 250. As we had very few documents having sentences greater than 250 and among those documents, few documents were having sentences greater than 2000 too. So creating an embedding size 2000*300 was increasing model complexity and was not improving performance of the model.

6.2 Neural Network Results:

Our neural network architecture has so many hyperparameters and to comp up with the best model we had to go through multiple experiments. Few parameters were fixed through multiple experiments and few parameters were selected after learning multiple papers. We have decided to select tanh and sigmoid activation function by learning other research papers and hyperparameters like numbers of filter and window size were decided through multiple experiments, which we will discuss in below paragraphs.

6.2.1 Selection of window size :

The window size is one of the important hypermeter of the experiment because a big number will encapsulate an entire sentence might overfit to training data and too small window size may lose the meaning of the sentence. So, to know what window size will be best fit for our experiments, we ran our training model with different settings and observed the loss and error continuously and found out 3 is the best suited number.

6.2.3 Selection of filter size :

While experimenting with number of filters keeping window size constant at 3, we observed that with increase in number of filters the loss and error also decreased but as the number of filters become larger the execution time increased exponentially. So to maintained balance between both the factors we came up with a number after which the loss and error decreased slowly and the best achievable number for us is 300.

6.2.4 Result of the best model :

In our initial model, the training and validation error was not converging and validation error was much higher. We started tuning each and every hyperparameter but had almost no impact on the model. The model showed great changes while changing filter size and window size. Fig 11 shows the error and loss values before turning hyperparameters and Fig 12 shows constant decline in loss and error after selecting the window size and filter size.

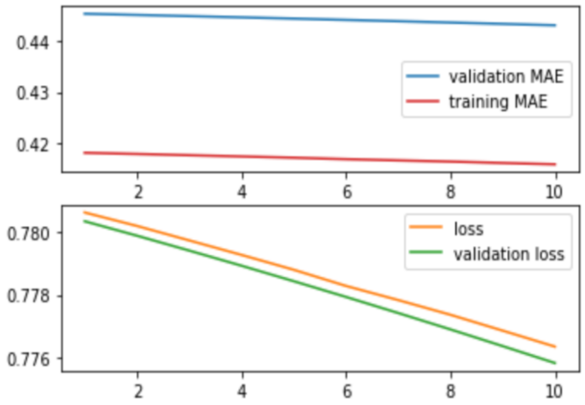


Fig. 11: Mean average error and loss on training and validation data when window size = 10 and number of filters = 100

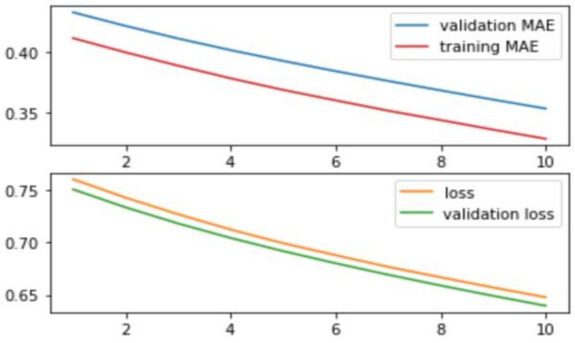
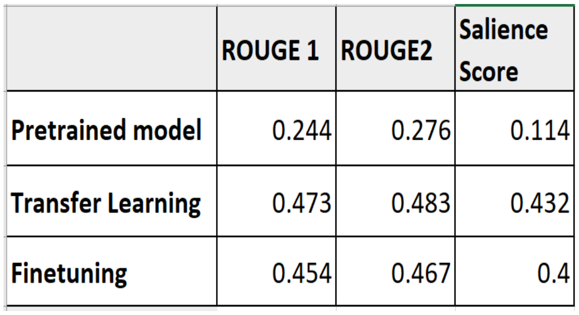


Fig. 12: Mean average error and loss on training and validation data when window size = 3 and number of filters = 300

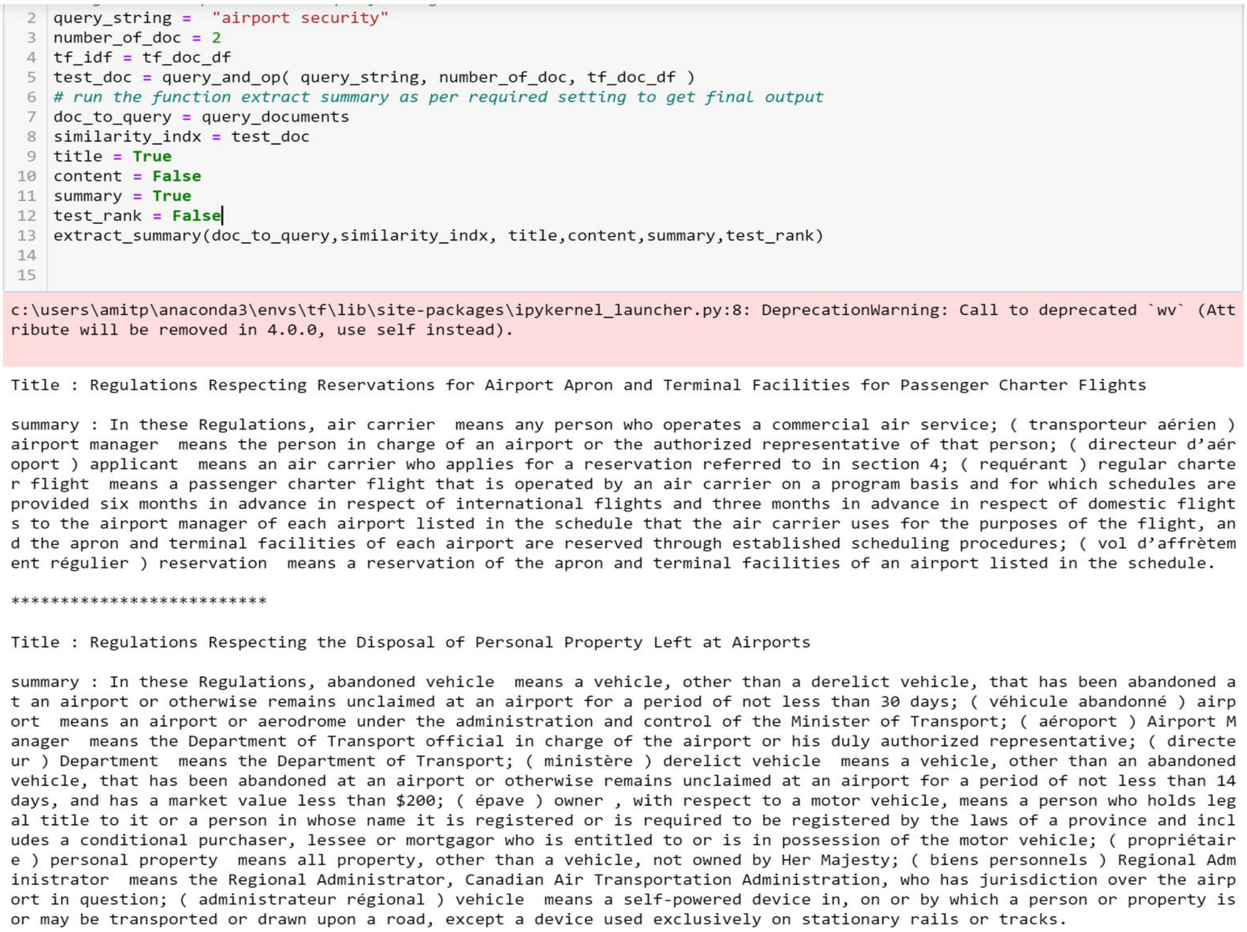
After training the model, the model performance was calculated with the test set. Since our goal is also to eliminate small data set problems, we conducted transfer learning by freezing the weights of 1st layer of network and training the rest of the layers and achieved a high increase of ROUGE score which can be seen from Table 2. We then tried fine tuning on regulation data by releasing all the weights and training for few iterations but we could not achieve ROUGE score higher than transfer learning. This happens sometimes because while training again it messes with the pattern learn by a previously trained model.



	ROUGE 1	ROUGE2	Salience Score
Pretrained model	0.244	0.276	0.114
Transfer Learning	0.473	0.483	0.432
Finetuning	0.454	0.467	0.4

Table 2:Scores of models trained with CNN/Daily mail dataset

Fig. 13 shows the summary extracted by our text summarization model. The query words send by the user to the model are "airport" and "security", numbers of document requested are 2 and from the title of the document, we can see that extracted documents are related to airport security and the summary contain the definition of the words mentions on the title of each legal text.



```
query_string =  "airport security"
number_of_doc = 2
tf_idf = tf_doc_df
test_doc = query_and_op( query_string, number_of_doc, tf_doc_df )
# run the function extract summary as per required setting to get final output
doc_to_query = query_documents
similarity_indx = test_doc
title = True
content = False
summary = True
test_rank = False
extract_summary(doc_to_query,similarity_indx, title,content,summary,test_rank)
```

```
c:\users\amitp\anaconda3\envs\tf\lib\site-packages\ipykernel_launcher.py:8: DeprecationWarning: Call to deprecated `wv` (Att
ribute will be removed in 4.0.0, use self instead).

Title : Regulations Respecting Reservations for Airport Apron and Terminal Facilities for Passenger Charter Flights

summary : In these Regulations, air carrier  means any person who operates a commercial air service; ( transporteur aérien )
airport manager  means the person in charge of an airport or the authorized representative of that person; ( directeur d'aér
oport ) applicant  means an air carrier who applies for a reservation referred to in section 4; ( requérant ) regular charte
r flight  means a passenger charter flight that is operated by an air carrier on a program basis and for which schedules are
provided six months in advance in respect of international flights and three months in advance in respect of domestic flight
s to the airport manager of each airport listed in the schedule that the air carrier uses for the purposes of the flight, an
d the apron and terminal facilities of each airport are reserved through established scheduling procedures; ( vol d'affrètem
ent régulier ) reservation  means a reservation of the apron and terminal facilities of an airport listed in the schedule.

**************************

Title : Regulations Respecting the Disposal of Personal Property Left at Airports

summary : In these Regulations, abandoned vehicle  means a vehicle, other than a derelict vehicle, that has been abandoned a
t an airport or otherwise remains unclaimed at an airport for a period of not less than 30 days; ( véhicule abandonné ) airp
ort  means an airport or aerodrome under the administration and control of the Minister of Transport; ( aéroport ) Airport M
anager  means the Department of Transport official in charge of the airport or his duly authorized representative; ( directe
ur ) Department  means the Department of Transport; ( ministère ) derelict vehicle  means a vehicle, other than an abandoned
vehicle, that has been abandoned at an airport or otherwise remains unclaimed at an airport for a period of not less than 14
days, and has a market value less than $200; ( épave ) owner , with respect to a motor vehicle, means a person who holds leg
al title to it or a person in whose name it is registered or is required to be registered by the laws of a province and incl
udes a conditional purchaser, lessee or mortgagor who is entitled to or is in possession of the motor vehicle; ( propriétair
e ) personal property  means all property, other than a vehicle, not owned by Her Majesty; ( biens personnels ) Regional Adm
inistrator  means the Regional Administrator, Canadian Air Transportation Administration, who has jurisdiction over the airp
ort in question; ( administrateur régional ) vehicle  means a self-powered device in, on or by which a person or property is
or may be transported or drawn upon a road, except a device used exclusively on stationary rails or tracks.
```

Fig. 13: The shows our summarization model's output.

6.3 Discussion

In this project, we tried to replicate the behavior of Convolutional Neural Network on image classification in the text summarization task. CNN requires tuning of multiple parameters and also requires generous amount of time to train. To achieve the best performing model we need to train the model again and again and it requires a large amount of time. For training 500 documents for 200 iterations in 1GPU core and 32 GB RAM, it took 10 hours to complete the training. While dealing with millions of documents it will require months of time which is computationally expansive and time consuming. Another limitation of the model is the fixed document size. During training we took the maximum number of sentences in the document as 250. In the test set, if a document appears with number sentences greater than 250 then the summary of the document could not be calculated.

One of the most point to note here is the ROUGE score. It does not compare semantic meaning of the sentence and score the sentence based on exact match. If a sentence with same meaning but with different wording present in the sentence then it will give it a low score which will greatly impact the model performance.

7. CONCLUSION AND FUTURE WORKS

Implementing the concept of image recognition in text summarization was a difficult task. To support the project very few previously done work found. So, implementing this project was a very challenging and exciting opportunity. The project also taught us the importance of pre processing of data and hyperparameters. Since our computation capacity was limited, a major learn from this project was to get state of art result with few iteration of training.

Since our data set was quite complex and was not normally distributed, deciding best number parameters were little difficult. In future we would like to take more pre-preprocessing steps for selecting important sentences to fit it into the fixed sentence size. ROUGE score can be replaced with semantic similarity scoring technique for best prediction of summary. Word embedding is one of the important factor for capturing the meaning. In this project we used Google’s pretrained embedding which lack legal text. In future we want to build our own embedding model which would be a best fit for legal text.

8. REFERENCES

[1] Cecile Robin, James O' Neill, Leona O' Brien, Paul Buitelaar. An Analysis of Topic Modelling for Legislative Texts. 16th International Conference on Artificial Intelligence and Law (ICAIL 2017), London, UK, 16 June

[2] Eric P.Xing, Pengtao Xie. Integrating Document Clustering and Topic Modeling. Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)

[3] Liang Yao, Yin Zhang, Qinfei Chen, Hongze Qian, Baogang Wei, Zhifeng Hu. Mining coherent topics in documents using word embeddings and large-scale text data. Engineering Applications of Artificial Intelligence, Volume 64, September 2017

[4] Luhn, H. P. The automatic creation of literature abstracts. IBM Journal of research and development (1958)

[5] Laura Manor, Junyi Jessy Li. Plain English Summarization of Contracts. Natural Legal Language Processing (NLLP) co-located with NAACL 2019 in Minneapolis, Minnesota, USA

[6] Begum Mutlu, Ebru A. Sezer, M. Ali Akcayol. Multi-document extractive text summarization: A comparative assessment on features. Knowledge-Based Systems, Volume 183, 1 November 2019, 104848

[7] Y. Zhang et al, Extractive Document Summarization Based on Convolutional Neural Networks, IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, p. 918-922, 2016.

[8] Yong Zhang, Joo Er Meng, Mahardhika Pratama Extractive Document Summarization Based on Convolutional Neural Networks Y. Zhang et al.(2016)

1. APPENDIX

1. **Code for insight extraction from regulatory documents using text summarization techniques**

   https://github.com/kshirabdhip/Legal_text_summarization.git

2. **Dataset download link**

   ftp://205.193.86.89/data.zip

3. **Convolutional Neural Network implementation reference**

   https://github.com/alexvlis/extractive-document-summarization

4. **CNN implementation of TensorFlow tutorial**

   http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/