Extractive Text Summarization using Neural Networks

A report submitted in partial fulfillment of the requirements for the

CS356 Mini Project / Industrial Training

Submitted by

Sourab Mangrulkar (14CSE1024)

Department of Computer Science and Engineering

National Institute of Technology Goa

Under the guidance of

Dr. Dileep A. D.

School of Computing and Electrical Engineering

Indian Institute of Technology Mandi

May-July, 2017

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The main objective here is to implement Single Document Extractive Text Sumarization using Neural Networks and Deep Learning. This included implementing Convolutional Neural Network(CNN) to train on CNN and Dailymail datasets. The features generated from Convolutional Neural Network(CNN) were used for implementing Linear SVM and Recurrent Neural Network(RNN) to train on CNN and Dailymail datasets. The tools used are Python3, Tensorflow and Scikit-Learn.

1.1 Text Summarization

Text summarization is an instance of Automatic summarization. It is also known as document summarization. Text summarization is the process of shortening a text document and generating a summary using some software. The summary generated by the software must contain all the main points of the original document. There are two approaches to text summarization, Extractive Text summarization and Abstractive Text Summarization. Based on number of documents used for generating a summary, there are two categories, Single Document Summarization and Multi-Document Summarization. The focus in this report is on Single Document Summarization. The evaluation measure used for text summarization models is ROUGE measure.

1.1.1 Extractive Text Summarization

Extractive methods in general context implement the idea of selecting a subset of given data such that the selected subset contains all the important information of the entire original data. Extractive summarization methods are popular for image collection summarization and video summarization.

Single Document Extractive Text Summarization methods involve selecting the most informative sentences from a given text document and generating a representative summary of the original document.

E.g. **CNN NEWS Article:**
cigarettes have been put out across the bars of New Orleans. ***cigars are welcome no more.*** *the city known for excess of everything – drinking , eating , dancing in the street until all hours – went smoke - free as tuesday became wednesday at midnight.* ***how can that be ? it turns out that the city known for its over-the - top Mardi Gras celebrations and incredible jazz fests (starting friday !) did n't want its waiters and musicians to have to breathe smoke to do their jobs anymore. the New Orleans City Council passed its ban against smoking in most places across the city – including bars , casinos and restaurants – in january , and the vote was unanimous , the New Orleans Times-Picayune reports.*** *bar owners worried about potential revenue loss , while puffing customers bemoaned the loss of their smoking spots. Harrah 's New Orleans and bar owners filed a lawsuit to stop the ban , and a hearing is scheduled in state court in a month , CNN affiliate WAPT reports.* ***fines start at $ 50.*** *luckily for us , none of CNN 's 15 New Orleans must - do 's – including touring Treme or eating a beignet – requires smoking.*

Extractive Summary:(Sentences selected from the article are shown bold in the above article)
cigars are welcome no more. how can that be ? it turns out that the city known for its over-the - top Mardi Gras celebrations and incredible jazz fests (starting friday !) did n't want its waiters and musicians to have to breathe

smoke to do their jobs anymore. the New Orleans City Council passed its ban against smoking in most places across the city – including bars , casinos and restaurants – in january , and the vote was unanimous , the New Orleans Times-Picayune reports. fines start at $ 50.

1.1.2 Abstractive Text Summarization

Abstractive Text Summarization methods build an internal schematic representation of the document and then use natural language generation techniques to generate the representative summary of the original document. Such a summary might have verbal innovations. Research till date shows much work in Abstractive Text Summarization(generating summaries similar to summaries generated by humans) as opposed to Extractive Text Summarization.

E.g. **Consider the CNN NEWS article from the sub-section 1.1.1.**

Abstractive Summary:

*New Orleans bars are smoke - free as of wednesday *morning*. a lawsuit by Harrah 's and bar owners seeks to *overturn* the ban.*

1.1.3 ROUGE(Recall-Oriented Understudy for Gisting Evaluation) Measure

ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. **The metrics compare an automatically produced summary or translation against a reference or a set of reference (human-produced) summaries or translation.** [4]

It is recall-based because importance is given to the presence of all the important information of the reference summary/summaries in the system generated summary. Recall can be computed with respect to unigram, bigram, trigram, or 4-gram matching. For example, *ROUGE-1 is computed as:*

$$\frac{number\_of\_overlapping\_unigrams\_between\_system\_summary\_and\_reference\_summary}{number\_of\_unigrams\_in\_reference\_summary}$$

If there are multiple references, the ROUGE-1 scores are averaged. ROUGE-2 and ROUGE-3 are similarly computed but instead of unigrams, bigrams and trigrams are used repectively.

There are some other different types of ROUGE measures such as ROUGE-N, ROUGE-L, ROUGE-S etc. Short descriptions of these ROUGE measures:

1. ROUGE-N: It computes n-gram recall between a system summary and a set of candidate summaries. It is computed as follows:

$$\frac{\sum\limits_{s\epsilon\{reference\_summaries\}} \sum\limits_{gram_n \epsilon s} Count_{match}(gram_n)}{\sum\limits_{s\epsilon\{reference\_summaries\}} \sum\limits_{gram_n \epsilon s} Count(gram_n)}$$

   where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in system summary and set of reference summaries [4].

2. ROUGE-L: LCS(Longest Common Subsequence) principle is used to find the ***longest common sequence*** of words between system summary and a set of reference summaries. An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order. Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length [4].

3. ROUGE-S: Skip-bigrams are all in-order word pairs in a sentence [4]. For example, in the sentence *"I like Neural Networks"*, the skip-bigrams are {*"I like", "I Neural", "I Networks", "like Neural", "like Networks", "Neural Networks"*}. **ROUGE-Si** measures the overlap of word pairs that have a maximum of **'i' (number)** gaps in between words. For example, ROUGE-S4 measures the overlap of word pairs that have a maximum of 4 gaps in between words.

The measure to be used depends on the specific problem being evaluated. ***As our focus is on extractive text summarization where verbosity is considerable for both system and reference summaries, the measures best suited for the evaluation are ROUGE-1 and ROUGE-L.***

1.2 Artifical Neural Networks

Artifical Neural networks is a programming paradigm which enables computer to learn from observational data. It is similar to the way biological neural networks make decisions by weighing the input evidences. Neural networks are being used in variety of tasks such as machine translation, speech recognition, computer vision, image captioning, text summarization, chatbots, language modelling and in many other domains.

1.2.1 Artifical Neurons and Neural Network

Artificial Neurons are the basic units in an artificial neural network(also referred as neural network for simplicity). An artificial neuron (also referred as neuron for simplicity) receives one or more inputs and computes the weighted sum of the inputs based on the weights assigned to inputs. This weighted sum is then passed through a non-linear function(activation function) which provides the final output. There are various types of activation functions such as sigmoid , tanh, ReLU(Rectified Linear Units) activation functions and so on. Figure 1.1 shows the structure of an artificial neuron.

A connected network of such neurons is known as neural network. A neural network is composed of layers of neurons connected with each other. A neural network is basically divided into three different parts, namely input layer, output layer and remaining layer(s) in between them known as hidden layer(s). Figure 1.2 shows a neural network having only one hidden layer. There are two types of neural networks depending on the direction of flow of information in neural network, namely Feedforward Neural Networks and

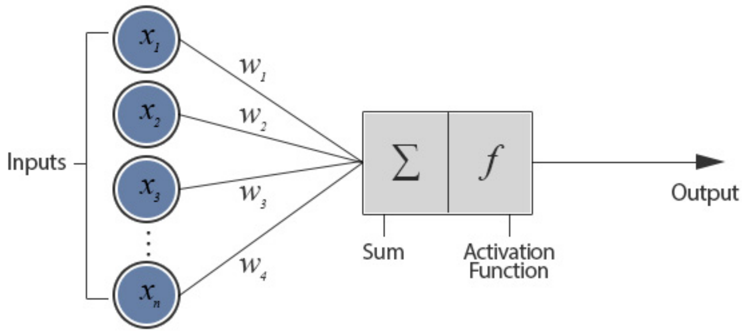


Figure 1.1: Artificial Neuron Structure (By Glosser.ca)

Recurrent Neural Networks. There is a class of neural networks known as Deep Neural Networks which can be defined as neural networks with a large number of hidden layers.

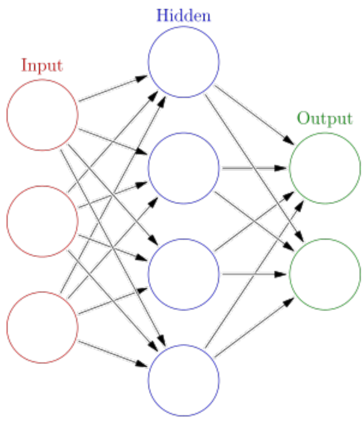


Figure 1.2: Example of a Neural Network (By Glosser.ca)

A Learning algorithm is used to train a neural network using the training dataset. Learning algorithm learns the parameters (weights and biases) such that the loss computed by the Cost function is minimum. In other words, learning algorithm optimises the parameters of the neural networks. There are various types of cost functions such as Quadratic cost function, Cross-Entropy cost function and log-likelihood cost function. There are many learning algorithms. the most widely used learning algorithm is Stochastic

Gradient Descent Algorithm which uses Backpropogation algorithm to compute the gradients of the loss with respect to the parameters of the neural network.

1.2.2 Feedforward Neural Network

A Feedforward Neural Network is a neural network in which information flow is only in a single direction, from input layer to output layer via hidden layer(s). There are no cycles or loops in the network. Figure 1.2 is an example of feedforward neural network. Feedforward neural networks are of two types, namely single-layer and multi-layer. Single-layer feedforward neural network has only two layers, one is input layer and the other is output layer. Multi-layer feedforward neural networks include hidden layers in addition to input layer and output layer.

1.2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network is a variant of deep feedforward neural network. They have been found to be best suited for analyzing visual imagery. They have applications in image and video recognition, recommender systems and natural language processing. They require minimal preprocessing and have translation invariance characteristics due to their architecture based on shared weights.

Hidden layers in a CNN can be either convolutional, pooling or fully connected. Figure 1.3 shows an example of a CNN having two convolutional layers (also includes pooling layers) and a fully connected layer.

1.2.3.1 Convolutional layer

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution approximates the response of an individual neuron to visual stimuli. Each convolutional neuron processes data only for its receptive field.The convolution operation reduces the number of free parameters and improve generalization.

1.2.3.2 Pooling Layer

Pooling layer combines output of neuron clusters at one layer into a single neuron in the next layer. For example, max pooling uses the maximum value from each of a cluster of neurons at the prior layer.

1.2.3.3 Fully Connected Layer

In fully connected layer, every neuron in one layer is connected to every neuron in the next layer

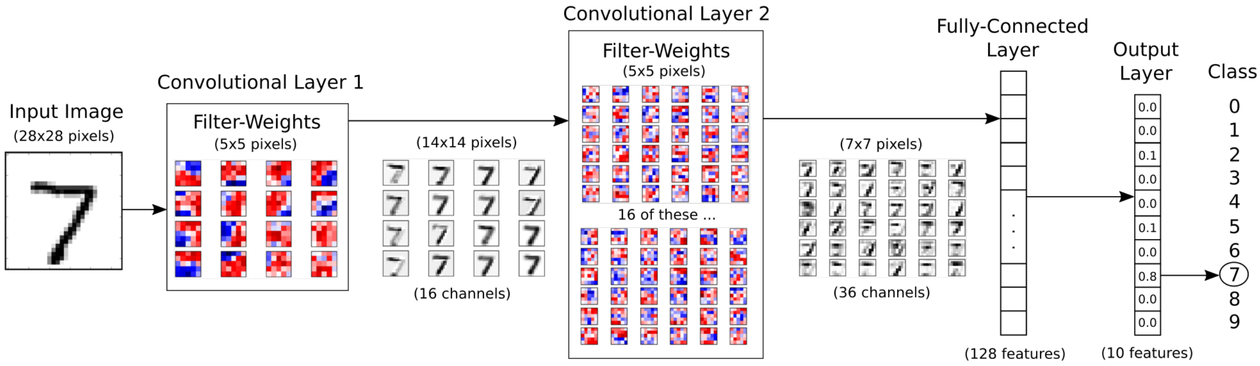


Figure 1.3: Example of a Convolutional Neural Network Source: Hvass

1.2.4 Recurrent Neural Network (RNN)

Recuurent Neural Networks are used in the tasks where output depends not only on the present input but also on the previous inputs and in some cases even on the future inputs. For example, to find a missing word in a sequence, one might have to look at both left and right context. RNNs can process sequential information. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a “memory” which captures information about what has been calculated so far. Figure 1.4 shows an example of a RNN and its unfolding in time.

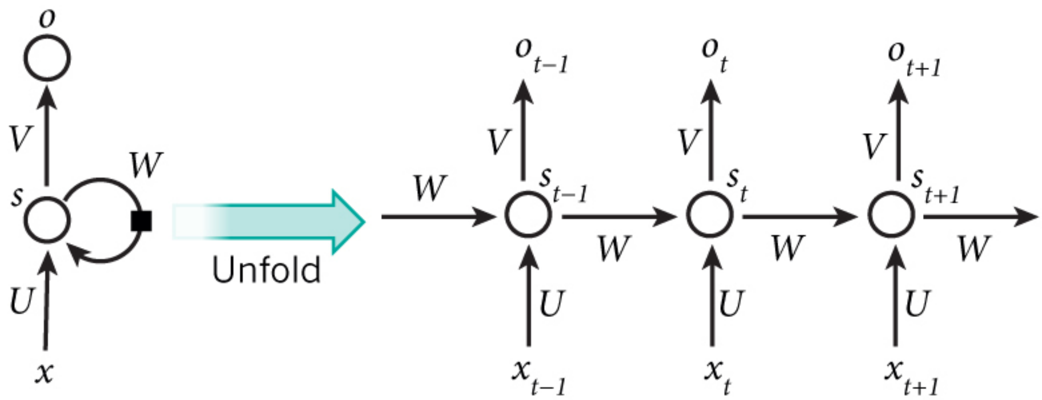


Figure 1.4: Example of a Recurrent Neural Network and the unfolding in time of the computation involved in its forward computation. Source: Nature

There are different types of RNNs such as Bidirectional RNNs, LSTM networks and GRU networks. Backpropagation Through Time(BPTT) algorithm is used to compute the gradients in case of RNNs. RNNs have many applications in Natural language Processing(NLP) tasks and have shown a great success. Some of the applications of RNNs in NLP tasks include language modelling, machine translation, image captioning, text summarization and speech recognition.

1.2.5 Tensorflow

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research.

We will be using Tensorflow to develop and implement neural networks.

1.2.6 Scikit-Learn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

We will be using Scikit-learn to implement linear support vector machine.

Chapter 2

Related Work

Extractive Summarization methods are classified into two categories, namely Unsupervised Learning methods and Supervised Learning Methods. Recent work has been in the unsupervised learning methods.

2.0.1 Unsupervised Learning Methods [1]

Unsupervised Learning Methods do not require training data, domain independent and portable. In many cases getting a lot of training data can be a costly affair and unsupervised learning methods overcome this issue but have a very sophisticated algorithms. Unsupervised learning methods for extractive text summarization can be broadly divided into the following four types:

- *Graph Based Approach*

  Graph based methods are very popular for text summarization due to their ability to efficiently represent the structure of documents [1]. LexRank [8] is an algorithm which uses the mechanism of random walks and eigenvector centrality to estimate sentence importance. The sentences in the document are represented as vertices of the graph and the edges between these vertices are the weighted cosine similarity values. PageRank [14] algorithm is applied on the graph constructed to rank the sentences. A summary is formed by combining the top ranking sentences, using a threshold or length cutoff to limit the size of the summary. TextRank is another graph based algorithm which is very similar to LexRank. There is an algorithm which uses external knowledge from Wikipedia and uses bipartite graph framework [7].

- *Concept Based Approach* [1]

  The basic steps in concept based methods are:

  1. From external knowledge base(Wikipedia, HowNet, WordNet) extract the concepts for the given text.
  2. Relationship between concepts and sentences is depicted by constructing conceptual vector or graph model.
  3. Sentences are scored by implementing a ranking algorithm.
  4. Based on the ranking scores of the sentences generate summaries

- *Fuzzy Logic Based Approach* [3] [9]

  The fuzzy logic approach mainly contains four components: defuzzifier, fuzzifier, fuzzy knowledge base and inference engine [1]. Figure 2.1 shows architecture of a fuzzy logic based approach for text summarization.

- *Latent Semantic Analysis Method* [11] [13]

  Latent Semantic Analysis(LSA) is a method which excerpt hidden semantic structures of sentences and words that are popularly used in text summarization task. It is an unsupervised learning approach that does not demand any sort of external or training knowledge. LSA captures the text of the input document and excerpt information such as words that frequently occur together and words that are commonly seen in different sentences. A high number of common words amongst the sentences illustrate that the sentences are semantically related. The

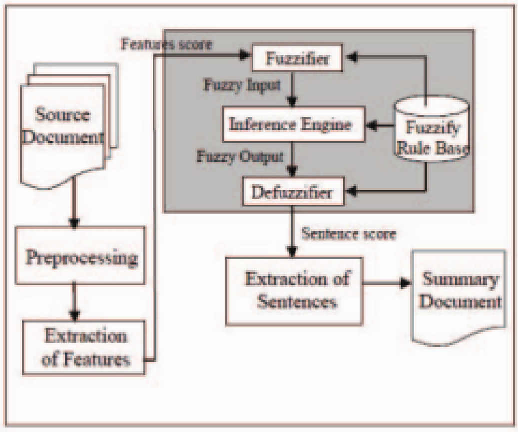


Figure 2.1: Example of a fuzzy logic based approach for text summarization [3]

advantage of adopting LSA vectors for summarization over word vectors is that conceptual relations as represented in the human brain are naturally captured in the LSA.

2.0.2 Supervised Learning Methods

Supervised Learning Methods require training data. They are based on the classification approach at sentence level where sentences are either labelled as "summary sentence" or "non-summary sentence". These methods learn to classify sentences among those two classes. Features of sentences that make them good candidates for inclusion in the summary are learnt by the supervised model using training data. Features might include the position of sentence in the document, the number of words in the sentence, etc. The main difficulty in supervised extractive summarization is that the known summaries must be manually created by extracting sentences so the sentences in an original training document can be labelled as "in summary" or "not in summary". This is not typically how people create summaries, so simply using journal abstracts or existing summaries is usually not sufficient. The sentences in these summaries do not necessarily match up with sentences in the original text, so it would be difficult to assign labels to examples for training. Note, however, that these natural summaries can still be used for evaluation purposes, since ROUGE-1 only cares about unigrams.

Some of the supervised learning methods are machine learning approach based on Bayes rule, neural networks based approach and Conditional Random Fields(CRFs) [1]. A two layer Recurrent Neural Network based Sequence Model is proposed in [2]. Figure shows the architecture of the model proposed in [2]. The bottom layer operates at word level within each sentence,while the top layer runs over sentences. Double-pointed arrows indicate a bi-directional RNN. The top layer with 1's and 0's is the sigmoid activation based classification layer that decides whether or not each sentence belongs to the summary. The decision at each sentence depends on the content richness of the sentence, its salience with respect to the document, its novelty with respect to the accumulated summary representation and other positional features [2].

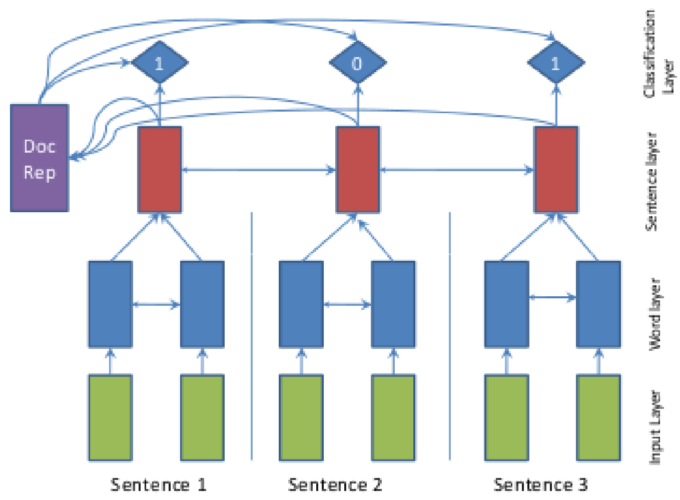


Figure 2.2: Architecture of Recurrent Neural Network based Sequence Model proposed in [2].

Chapter 3

Proposed method

This chapter is divided into following sections:

1. Description of Dataset
2. A Convolutional Neural Network Based Model
3. A Linear Support Vector Machine Based Model
4. A Bidirectional Recurrent Neural Network Based Model

3.1 Description of Dataset

The datasets used for the proposed models are:

3.1.1 CNN Dataset

Number of documents in training data folder: 83567
Number of documents in validation data folder: 1220
Number of documents in test data folder: 1093
Average number of words per sentence: 24
Average number of sentences per article:30
Vocabulary Size: 222436
Number of words in sentence after padding: 60

Each document has article with one sentence per line and labelled by a number in the set $\{0, 1, 2\}$. Sentences labelled 0 are to be included in the extractive summary, sentences labelled 1 are not to be included in extractive summary and sentences labelled 2 may or may not be included in the extractive summary. Each document also has an *human generated summary* which will be referred to as *abstractive summary or gold summary* from now on for simplicity.

3.1.2 Dailymail Dataset

Number of documents in training data folder: 193981
Number of documents in validation data folder: 12147
Number of documents in test data folder: 10346
Average number of words per sentence: 29
Average number of sentences per article:27
Vocabulary Size: 397505
Number of words in sentence after padding: 60

Each document has article with one sentence per line and labelled by a number in the set $\{0, 1, 2\}$. Sentences labelled 0 are to be included in the extractive summary, sentences labelled 1 are not to be included in extractive summary and sentences labelled 2 may or may not be included in the extractive summary. Each document also has an *human generated summary* which will be referred to as *abstractive summary or gold summary* from now on for simplicity.

3.1.3 Cases considered

The following cases are considered for generating training data, validation data and test data which will be used by the proposed models:

CASE 1: Sentences labelled 2 will be relabelled as 1. In other words, sentences labelled 2 are not included in extractive summary but will be part of article.

CASE 2: Sentences labelled 2 will be retain that label. In other words, this is a classification problem with 3 classes.

CASE 3: Sentences labelled 2 will be excluded from the article. In other words, sentences labelled 2 will not even be part of article.

3.2 A Convolutional Neural Network Based Model

This model is similar to the one proposed in [5]. In our task, each sentence is represented as a matrix. Each row corresponds to word2vec [10] word embedding of a word. So, for a sentence with 5 words and using a 100-dimensional word embedding, we get 5x100 matrix as input.

We will use filters that slide over the full rows of the input matrix(words). So the width of the filters are same as that of the width of the input matrix. The heights, or region sizes, that have been used for the proposed CNN model are 3, 5, 7 and 9. The architecture and components of the proposed CNN model is similar to the one shown in Figure 3.1. Here we depict three filter region sizes: 2, 3 and 4, each of which has 2 filters. Every filter performs convolution on the sentence matrix and generates (variable-length) feature maps. Then 1-max pooling is performed over each map, i.e., the largest number from each feature map is recorded. Thus a univariate feature vector is generated from all six maps, and these 6 features are concatenated to form a feature vector for the penultimate layer. The final softmax layer then receives this feature vector as input and uses it to classify the sentence; here we assume binary classification and hence depict two possible output states. Source: Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification [5].

The network we will build will be similar to the one depicted in Figure 3.2. We will model extractive text summarization problem as sentence

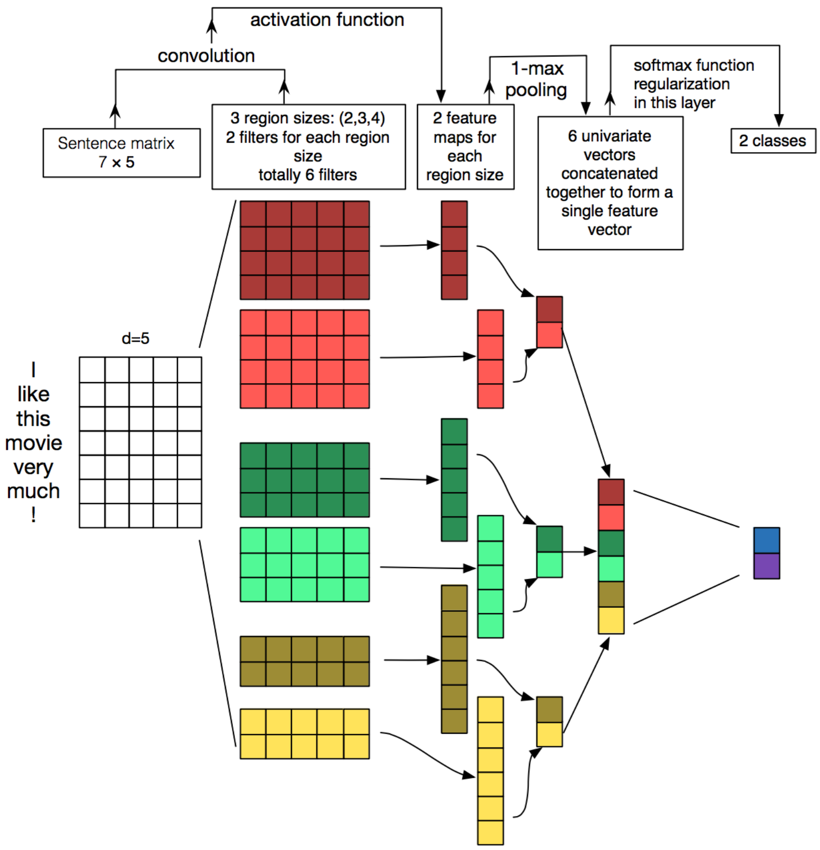


Figure 3.1: Illustration of a Convolutional Neural Network (CNN) architecture for sentence classification [5]

classification problem wherein inputs are sentences having labels. Here our assumption is that each sentence in an article is independent of other sentences. We find the labels for the sentences belonging to articles and based on those labels we pick the sentences and form extractive summaries for corresponding articles.

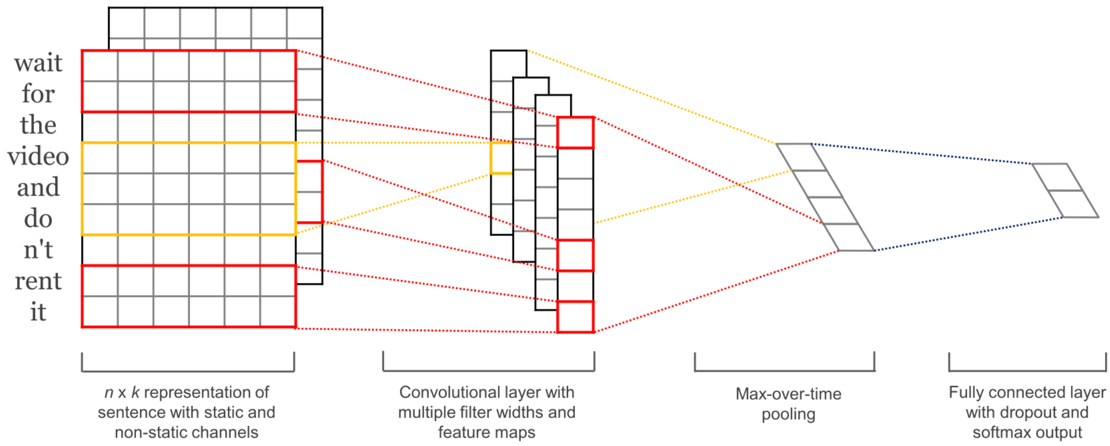


Figure 3.2: Proposed Convolutional Neural Network (CNN) model [5]

3.3 A Linear Support Vector Machine Based Model

The Feature Vectors generated by the CNN model in the penultimate layer is used as input for linear SVM and it is trained for different ***C values***. For creating this model, scikit-learn is used. *sklearn.svm.LinearSVC* is used as it uses liblinear library and can scale to large number of training samples.

3.4 A Bidirectional Recurrent Neural Network Based Model

The Feature Vectors generated by the CNN model in the penultimate layer is used as input for deep Bidirectional RNN. The Figure 3.3 shows the basic architecture of a deep Bidirectional RNN. The architecture of this proposed Bidirectional RNN model is sequence to sequence. Two multi-layer RNNs are stacked on top of each other and the inputs are passed at the bottom. The outputs at the top are computed based on the hidden state of both multi-layer RNNs. We have used LSTM cells in the above architecture as they are good at handling long term dependencies.

Each feature vector which is given as input is a fixed length representation of a sentence. The Bidirectional RNN is considered because it takes into account both the left and right context of a sentence(input feature vector) and thereby modelling the dependencies among sentences. The quality of the output depends on how good the fixed length representation of sentences are, i.e., how good the input feature vectors are.

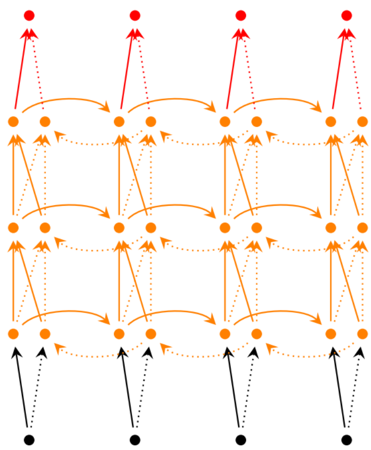


Figure 3.3: Bidiretional Recurrent Neural Network architecture. Source: WildML

Chapter 4

Results

4.1 CNN based Model

Refer to the section 3.2 for description of the CNN based model.

4.1.1 CNN dataset

Refer subsection 3.1.1 to know details about the CNN dataset. Refer to the three cases described in subsection 3.1.3

4.1.1.1 Case 1

Model Input Parameters:

- allow_soft_placement = True
- batch_size = 100
- checkpoint_every = 2000
- data_file = ./data/cnn_train.txt
- dev_sample_% = 0.1
- dropout_keep_prob = 0.5

- embedding_size = 128
- evaluate_every = 100
- filter_sizes = $3, 5, 7, 9$
- l2_reg_lambda = 0.05
- log_device_placement = False
- num_checpoints = 5
- num_epochs = 100
- num_filters = 64

Other parameters:

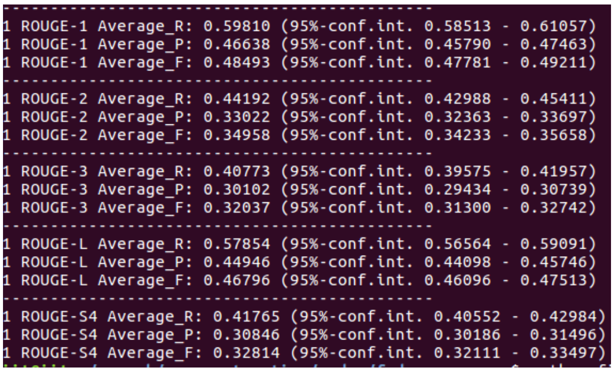
- max_sentence_length = 60
- learning_rate $\eta = 1e^{-3}$
- Train/Dev sample number = 2243200/249244

Evaluation on Test Data

- number of test samples = 33046
- **Accuracy** $= 0.685287 = 68.5287\%$

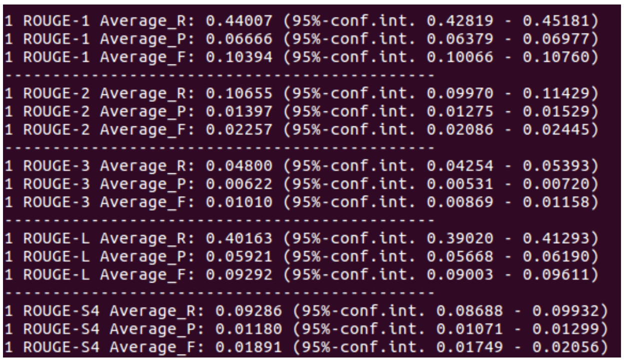
Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.1
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.2



```
------------------------------------------------
1 ROUGE-1 Average_R: 0.59810 (95%-conf.int. 0.58513 - 0.61057)
1 ROUGE-1 Average_P: 0.46638 (95%-conf.int. 0.45790 - 0.47463)
1 ROUGE-1 Average_F: 0.48493 (95%-conf.int. 0.47781 - 0.49211)
------------------------------------------------
1 ROUGE-2 Average_R: 0.44192 (95%-conf.int. 0.42988 - 0.45411)
1 ROUGE-2 Average_P: 0.33022 (95%-conf.int. 0.32363 - 0.33697)
1 ROUGE-2 Average_F: 0.34958 (95%-conf.int. 0.34233 - 0.35658)
------------------------------------------------
1 ROUGE-3 Average_R: 0.40773 (95%-conf.int. 0.39575 - 0.41957)
1 ROUGE-3 Average_P: 0.30102 (95%-conf.int. 0.29434 - 0.30739)
1 ROUGE-3 Average_F: 0.32037 (95%-conf.int. 0.31300 - 0.32742)
------------------------------------------------
1 ROUGE-L Average_R: 0.57854 (95%-conf.int. 0.56564 - 0.59091)
1 ROUGE-L Average_P: 0.44946 (95%-conf.int. 0.44098 - 0.45746)
1 ROUGE-L Average_F: 0.46796 (95%-conf.int. 0.46096 - 0.47513)
------------------------------------------------
1 ROUGE-S4 Average_R: 0.41765 (95%-conf.int. 0.40552 - 0.42984)
1 ROUGE-S4 Average_P: 0.30846 (95%-conf.int. 0.30186 - 0.31496)
1 ROUGE-S4 Average_F: 0.32814 (95%-conf.int. 0.32111 - 0.33497)
```

Figure 4.1: Reference: Original Extractive and Predicted: Predicted Extractive



```
1 ROUGE-1 Average_R: 0.44007 (95%-conf.int. 0.42819 - 0.45181)
1 ROUGE-1 Average_P: 0.06666 (95%-conf.int. 0.06379 - 0.06977)
1 ROUGE-1 Average_F: 0.10394 (95%-conf.int. 0.10066 - 0.10760)
------------------------------------------------
1 ROUGE-2 Average_R: 0.10655 (95%-conf.int. 0.09970 - 0.11429)
1 ROUGE-2 Average_P: 0.01397 (95%-conf.int. 0.01275 - 0.01529)
1 ROUGE-2 Average_F: 0.02257 (95%-conf.int. 0.02086 - 0.02445)
------------------------------------------------
1 ROUGE-3 Average_R: 0.04800 (95%-conf.int. 0.04254 - 0.05393)
1 ROUGE-3 Average_P: 0.00622 (95%-conf.int. 0.00531 - 0.00720)
1 ROUGE-3 Average_F: 0.01010 (95%-conf.int. 0.00869 - 0.01158)
------------------------------------------------
1 ROUGE-L Average_R: 0.40163 (95%-conf.int. 0.39020 - 0.41293)
1 ROUGE-L Average_P: 0.05921 (95%-conf.int. 0.05668 - 0.06190)
1 ROUGE-L Average_F: 0.09292 (95%-conf.int. 0.09003 - 0.09611)
------------------------------------------------
1 ROUGE-S4 Average_R: 0.09286 (95%-conf.int. 0.08688 - 0.09932)
1 ROUGE-S4 Average_P: 0.01180 (95%-conf.int. 0.01071 - 0.01299)
1 ROUGE-S4 Average_F: 0.01891 (95%-conf.int. 0.01749 - 0.02056)
```

Figure 4.2: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.1: Referring Figure 4.1

Table	Average_R
ROUGE-1	59.8 ± 2.5
ROUGE-L	57.9 ± 2.5

Table 4.2: Referring Figure 4.2

Table	Average_R
ROUGE-1	44.0 ± 2.4
ROUGE-L	40.2 ± 2.3

4.1.1.2 Case 2

Model Input Parameters:

- allow_soft_placement = True
- batch_size = 100
- checkpoint_every = 2000
- data_file = ./data/cnn_train.txt
- dev_sample_% = 0.1
- dropout_keep_prob = 0.5
- embedding_size = 128
- evaluate_every = 100
- filter_sizes = $3, 5, 7, 9$
- l2_reg_lambda = 0.05
- log_device_placement = False
- num_checpoints = 5
- num_epochs = 100
- num_filters = 64

Other parameters:

- max_sentence_length = 60
- learning_rate $\eta = 1e^{-3}$
- Train/Dev sample number = 2243200/249244

Evaluation on Test Data

- number of test samples = 33046
- **Accuracy** = 0.565273 = 56.5273%

Evaluating ROUGE Scores

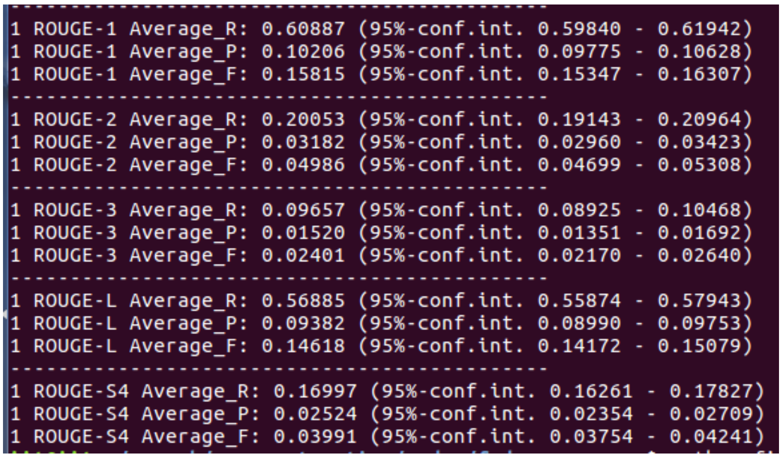
1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.3
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.4

```
1 ROUGE-1 Average_R: 0.75702 (95%-conf.int. 0.74563 - 0.76787)
1 ROUGE-1 Average_P: 0.62538 (95%-conf.int. 0.61679 - 0.63414)
1 ROUGE-1 Average_F: 0.65407 (95%-conf.int. 0.64666 - 0.66133)
---------------------------------------------
1 ROUGE-2 Average_R: 0.66395 (95%-conf.int. 0.65118 - 0.67596)
1 ROUGE-2 Average_P: 0.54183 (95%-conf.int. 0.53231 - 0.55144)
1 ROUGE-2 Average_F: 0.56924 (95%-conf.int. 0.56056 - 0.57820)
---------------------------------------------
1 ROUGE-3 Average_R: 0.63551 (95%-conf.int. 0.62273 - 0.64785)
1 ROUGE-3 Average_P: 0.51747 (95%-conf.int. 0.50780 - 0.52750)
1 ROUGE-3 Average_F: 0.54376 (95%-conf.int. 0.53486 - 0.55301)
---------------------------------------------
1 ROUGE-L Average_R: 0.74963 (95%-conf.int. 0.73809 - 0.76057)
1 ROUGE-L Average_P: 0.61847 (95%-conf.int. 0.60982 - 0.62732)
1 ROUGE-L Average_F: 0.64722 (95%-conf.int. 0.63979 - 0.65501)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.63317 (95%-conf.int. 0.62038 - 0.64525)
1 ROUGE-S4 Average_P: 0.51562 (95%-conf.int. 0.50606 - 0.52523)
1 ROUGE-S4 Average_F: 0.54141 (95%-conf.int. 0.53265 - 0.55051)
```

Figure 4.3: Reference: Original Extractive and Predicted: Predicted Extractive

Table 4.3: Referring Figure 4.3

Table	Average_R
ROUGE-1	75.7 ± 2.2
ROUGE-L	75.0 ± 2.2



```
---------------------------------------------
1 ROUGE-1 Average_R: 0.60887 (95%-conf.int. 0.59840 - 0.61942)
1 ROUGE-1 Average_P: 0.10206 (95%-conf.int. 0.09775 - 0.10628)
1 ROUGE-1 Average_F: 0.15815 (95%-conf.int. 0.15347 - 0.16307)
---------------------------------------------
1 ROUGE-2 Average_R: 0.20053 (95%-conf.int. 0.19143 - 0.20964)
1 ROUGE-2 Average_P: 0.03182 (95%-conf.int. 0.02960 - 0.03423)
1 ROUGE-2 Average_F: 0.04986 (95%-conf.int. 0.04699 - 0.05308)
---------------------------------------------
1 ROUGE-3 Average_R: 0.09657 (95%-conf.int. 0.08925 - 0.10468)
1 ROUGE-3 Average_P: 0.01520 (95%-conf.int. 0.01351 - 0.01692)
1 ROUGE-3 Average_F: 0.02401 (95%-conf.int. 0.02170 - 0.02640)
---------------------------------------------
1 ROUGE-L Average_R: 0.56885 (95%-conf.int. 0.55874 - 0.57943)
1 ROUGE-L Average_P: 0.09382 (95%-conf.int. 0.08990 - 0.09753)
1 ROUGE-L Average_F: 0.14618 (95%-conf.int. 0.14172 - 0.15079)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.16997 (95%-conf.int. 0.16261 - 0.17827)
1 ROUGE-S4 Average_P: 0.02524 (95%-conf.int. 0.02354 - 0.02709)
1 ROUGE-S4 Average_F: 0.03991 (95%-conf.int. 0.03754 - 0.04241)
```

Figure 4.4: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.4: Referring Figure 4.4

Table	Average_R
ROUGE-1	60.9 ± 2.1
ROUGE-L	56.9 ± 2.1

4.1.1.3 Case 3

Model Input Parameters:

- allow_soft_placement = True
- batch_size = 100
- checkpoint_every = 2000
- data_file = ./data/cnn_train.txt
- dev_sample_% = 0.1
- dropout_keep_prob = 0.5
- embedding_size = 128
- evaluate_every = 100

- filter_sizes = $3, 5, 7, 9$
- l2_reg_lambda = 0.05
- log_device_placement = False
- num_checpoints = 5
- num_epochs = 100
- num_filters = 64

Other parameters:

- max_sentence_length = 60
- learning_rate $\eta = 1e^{-3}$
- Train/Dev sample number = 1794572/199396

Evaluation on Test Data

- number of test samples = 26452
- **Accuracy** = 0.707432 = 70.7432%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.5
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.6

```
1 ROUGE-1 Average_R: 0.72218 (95%-conf.int. 0.71048 - 0.73455)
1 ROUGE-1 Average_P: 0.74807 (95%-conf.int. 0.73863 - 0.75752)
1 ROUGE-1 Average_F: 0.70405 (95%-conf.int. 0.69504 - 0.71224)
---------------------------------------------
1 ROUGE-2 Average_R: 0.65636 (95%-conf.int. 0.64395 - 0.66961)
1 ROUGE-2 Average_P: 0.67774 (95%-conf.int. 0.66647 - 0.68829)
1 ROUGE-2 Average_F: 0.63810 (95%-conf.int. 0.62752 - 0.64777)
---------------------------------------------
1 ROUGE-3 Average_R: 0.63492 (95%-conf.int. 0.62265 - 0.64823)
1 ROUGE-3 Average_P: 0.65513 (95%-conf.int. 0.64377 - 0.66596)
1 ROUGE-3 Average_F: 0.61662 (95%-conf.int. 0.60551 - 0.62661)
---------------------------------------------
1 ROUGE-L Average_R: 0.71721 (95%-conf.int. 0.70566 - 0.72953)
1 ROUGE-L Average_P: 0.74240 (95%-conf.int. 0.73266 - 0.75202)
1 ROUGE-L Average_F: 0.69897 (95%-conf.int. 0.68992 - 0.70736)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.63037 (95%-conf.int. 0.61795 - 0.64360)
1 ROUGE-S4 Average_P: 0.65059 (95%-conf.int. 0.63952 - 0.66142)
1 ROUGE-S4 Average_F: 0.61184 (95%-conf.int. 0.60093 - 0.62197)
```

Figure 4.5: Reference: Original Extractive and Predicted: Predicted Extractive

```
1 ROUGE-1 Average_R: 0.54752 (95%-conf.int. 0.53583 - 0.55902)
1 ROUGE-1 Average_P: 0.11120 (95%-conf.int. 0.10692 - 0.11558)
1 ROUGE-1 Average_F: 0.16752 (95%-conf.int. 0.16265 - 0.17244)
---------------------------------------------
1 ROUGE-2 Average_R: 0.17799 (95%-conf.int. 0.16862 - 0.18768)
1 ROUGE-2 Average_P: 0.03410 (95%-conf.int. 0.03153 - 0.03680)
1 ROUGE-2 Average_F: 0.05194 (95%-conf.int. 0.04850 - 0.05524)
---------------------------------------------
1 ROUGE-3 Average_R: 0.09049 (95%-conf.int. 0.08270 - 0.09841)
1 ROUGE-3 Average_P: 0.01723 (95%-conf.int. 0.01513 - 0.01934)
1 ROUGE-3 Average_F: 0.02644 (95%-conf.int. 0.02366 - 0.02916)
---------------------------------------------
1 ROUGE-L Average_R: 0.50716 (95%-conf.int. 0.49559 - 0.51857)
1 ROUGE-L Average_P: 0.10146 (95%-conf.int. 0.09770 - 0.10570)
1 ROUGE-L Average_F: 0.15351 (95%-conf.int. 0.14910 - 0.15814)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.15074 (95%-conf.int. 0.14304 - 0.15934)
1 ROUGE-S4 Average_P: 0.02716 (95%-conf.int. 0.02496 - 0.02945)
1 ROUGE-S4 Average_F: 0.04175 (95%-conf.int. 0.03902 - 0.04452)
```

Figure 4.6: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.5: Referring Figure 4.5

Table	Average_R
ROUGE-1	72.2 ± 2.4
ROUGE-L	71.7 ± 2.4

Table 4.6: Referring Figure 4.6

Table	Average_R
ROUGE-1	60.9 ± 2.1
ROUGE-L	56.9 ± 2.1

4.1.2 Dailymail dataset

Refer subsection 3.1.2 to know details about the Dailymail dataset. Refer to the three cases described in subsection 3.1.3

4.1.2.1 Case 1

Model Input Parameters:

- allow_soft_placement = True
- batch_size = 100
- checkpoint_every = 2000
- data_file = ./data/cnn_train.txt
- dev_sample_% = 0.1
- dropout_keep_prob = 0.5
- embedding_size = 128
- evaluate_every = 100
- filter_sizes = $3, 5, 7, 9$
- l2_reg_lambda = 0.05
- log_device_placement = False
- num_checpoints = 5
- num_epochs = 100
- num_filters = 64

Other parameters:

- max_sentence_length = 60

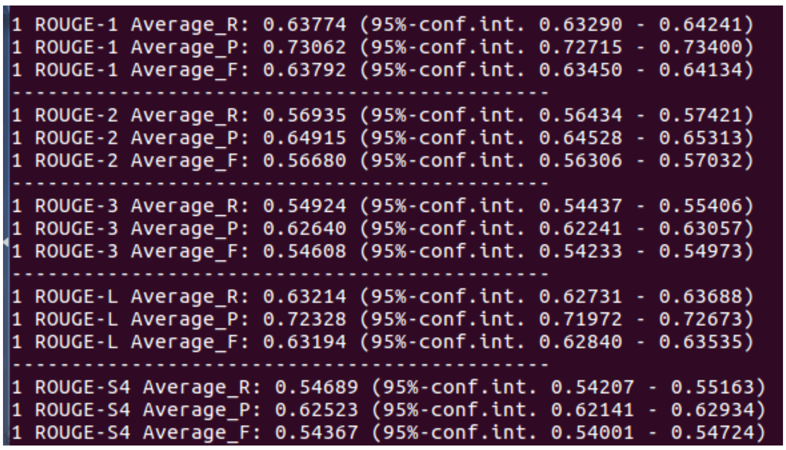
- learning_rate $\eta = 1e^{-3}$
- Train/Dev sample number = 4560099/506677

Evaluation on Test Data

- number of test samples = 264305
- **Accuracy** = 0.716199 = 71.6199%

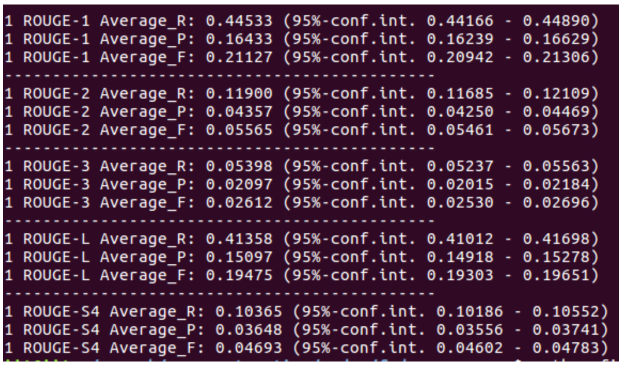
Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.7
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.8



```
1 ROUGE-1 Average_R: 0.63774 (95%-conf.int. 0.63290 - 0.64241)
1 ROUGE-1 Average_P: 0.73062 (95%-conf.int. 0.72715 - 0.73400)
1 ROUGE-1 Average_F: 0.63792 (95%-conf.int. 0.63450 - 0.64134)
---------------------------------------------
1 ROUGE-2 Average_R: 0.56935 (95%-conf.int. 0.56434 - 0.57421)
1 ROUGE-2 Average_P: 0.64915 (95%-conf.int. 0.64528 - 0.65313)
1 ROUGE-2 Average_F: 0.56680 (95%-conf.int. 0.56306 - 0.57032)
---------------------------------------------
1 ROUGE-3 Average_R: 0.54924 (95%-conf.int. 0.54437 - 0.55406)
1 ROUGE-3 Average_P: 0.62640 (95%-conf.int. 0.62241 - 0.63057)
1 ROUGE-3 Average_F: 0.54608 (95%-conf.int. 0.54233 - 0.54973)
---------------------------------------------
1 ROUGE-L Average_R: 0.63214 (95%-conf.int. 0.62731 - 0.63688)
1 ROUGE-L Average_P: 0.72328 (95%-conf.int. 0.71972 - 0.72673)
1 ROUGE-L Average_F: 0.63194 (95%-conf.int. 0.62840 - 0.63535)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.54689 (95%-conf.int. 0.54207 - 0.55163)
1 ROUGE-S4 Average_P: 0.62523 (95%-conf.int. 0.62141 - 0.62934)
1 ROUGE-S4 Average_F: 0.54367 (95%-conf.int. 0.54001 - 0.54724)
```

Figure 4.7: Reference: Original Extractive and Predicted: Predicted Extractive



```
1 ROUGE-1 Average_R: 0.44533 (95%-conf.int. 0.44166 - 0.44890)
1 ROUGE-1 Average_P: 0.16433 (95%-conf.int. 0.16239 - 0.16629)
1 ROUGE-1 Average_F: 0.21127 (95%-conf.int. 0.20942 - 0.21306)
---------------------------------------------
1 ROUGE-2 Average_R: 0.11900 (95%-conf.int. 0.11685 - 0.12109)
1 ROUGE-2 Average_P: 0.04357 (95%-conf.int. 0.04250 - 0.04469)
1 ROUGE-2 Average_F: 0.05565 (95%-conf.int. 0.05461 - 0.05673)
---------------------------------------------
1 ROUGE-3 Average_R: 0.05398 (95%-conf.int. 0.05237 - 0.05563)
1 ROUGE-3 Average_P: 0.02097 (95%-conf.int. 0.02015 - 0.02184)
1 ROUGE-3 Average_F: 0.02612 (95%-conf.int. 0.02530 - 0.02696)
---------------------------------------------
1 ROUGE-L Average_R: 0.41358 (95%-conf.int. 0.41012 - 0.41698)
1 ROUGE-L Average_P: 0.15097 (95%-conf.int. 0.14918 - 0.15278)
1 ROUGE-L Average_F: 0.19475 (95%-conf.int. 0.19303 - 0.19651)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.10365 (95%-conf.int. 0.10186 - 0.10552)
1 ROUGE-S4 Average_P: 0.03648 (95%-conf.int. 0.03556 - 0.03741)
1 ROUGE-S4 Average_F: 0.04693 (95%-conf.int. 0.04602 - 0.04783)
```

Figure 4.8: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.7: Referring Figure 4.7

Table	Average_R
ROUGE-1	63.8 ± 1.0
ROUGE-L	63.2 ± 1.0

Table 4.8: Referring Figure 4.8

Table	Average_R
ROUGE-1	44.5 ± 0.7
ROUGE-L	41.4 ± 0.7

4.1.2.2 Case 2

Model Input Parameters:

- allow_soft_placement = True
- batch_size = 100
- checkpoint_every = 2000
- data_file = ./data/cnn_train.txt
- dev_sample_% = 0.1
- dropout_keep_prob = 0.5

- embedding_size = 128
- evaluate_every = 100
- filter_sizes = $3, 5, 7, 9$
- l2_reg_lambda = 0.05
- log_device_placement = False
- num_checpoints = 5
- num_epochs = 100
- num_filters = 64

Other parameters:

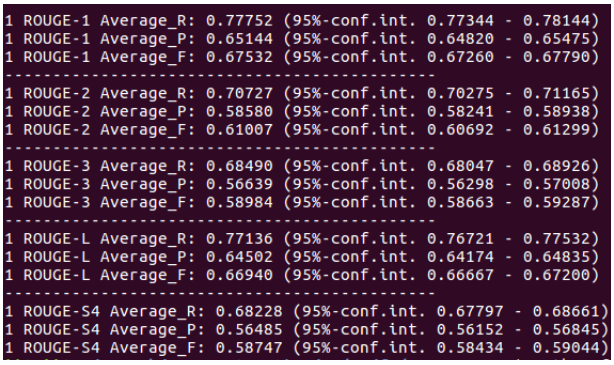
- max_sentence_length = 60
- learning_rate $\eta = 1e^{-3}$
- Train/Dev sample number = 4560099/506677

Evaluation on Test Data

- number of test samples = 264305
- **Accuracy** $= 0.60594 = 60.594\%$

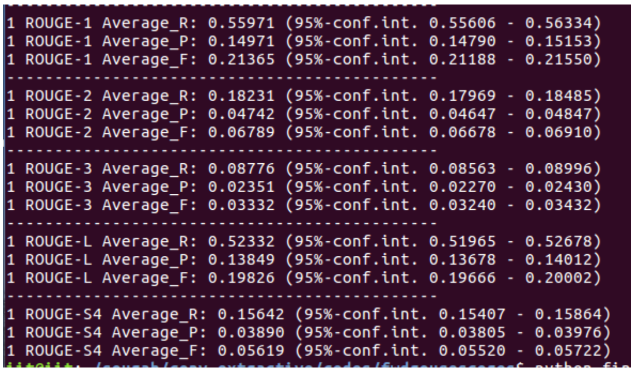
Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.9
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.10



```
1 ROUGE-1 Average_R: 0.77752 (95%-conf.int. 0.77344 - 0.78144)
1 ROUGE-1 Average_P: 0.65144 (95%-conf.int. 0.64820 - 0.65475)
1 ROUGE-1 Average_F: 0.67532 (95%-conf.int. 0.67260 - 0.67790)
---------------------------------------------
1 ROUGE-2 Average_R: 0.70727 (95%-conf.int. 0.70275 - 0.71165)
1 ROUGE-2 Average_P: 0.58580 (95%-conf.int. 0.58241 - 0.58938)
1 ROUGE-2 Average_F: 0.61007 (95%-conf.int. 0.60692 - 0.61299)
---------------------------------------------
1 ROUGE-3 Average_R: 0.68490 (95%-conf.int. 0.68047 - 0.68926)
1 ROUGE-3 Average_P: 0.56639 (95%-conf.int. 0.56298 - 0.57008)
1 ROUGE-3 Average_F: 0.58984 (95%-conf.int. 0.58663 - 0.59287)
---------------------------------------------
1 ROUGE-L Average_R: 0.77136 (95%-conf.int. 0.76721 - 0.77532)
1 ROUGE-L Average_P: 0.64502 (95%-conf.int. 0.64174 - 0.64835)
1 ROUGE-L Average_F: 0.66940 (95%-conf.int. 0.66667 - 0.67200)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.68228 (95%-conf.int. 0.67797 - 0.68661)
1 ROUGE-S4 Average_P: 0.56485 (95%-conf.int. 0.56152 - 0.56845)
1 ROUGE-S4 Average_F: 0.58747 (95%-conf.int. 0.58434 - 0.59044)
```

Figure 4.9: Reference: Original Extractive and Predicted: Predicted Extractive



```
---------------------------------------------
1 ROUGE-1 Average_R: 0.55971 (95%-conf.int. 0.55606 - 0.56334)
1 ROUGE-1 Average_P: 0.14971 (95%-conf.int. 0.14790 - 0.15153)
1 ROUGE-1 Average_F: 0.21365 (95%-conf.int. 0.21188 - 0.21550)
---------------------------------------------
1 ROUGE-2 Average_R: 0.18231 (95%-conf.int. 0.17969 - 0.18485)
1 ROUGE-2 Average_P: 0.04742 (95%-conf.int. 0.04647 - 0.04847)
1 ROUGE-2 Average_F: 0.06789 (95%-conf.int. 0.06678 - 0.06910)
---------------------------------------------
1 ROUGE-3 Average_R: 0.08776 (95%-conf.int. 0.08563 - 0.08996)
1 ROUGE-3 Average_P: 0.02351 (95%-conf.int. 0.02270 - 0.02430)
1 ROUGE-3 Average_F: 0.03332 (95%-conf.int. 0.03240 - 0.03432)
---------------------------------------------
1 ROUGE-L Average_R: 0.52332 (95%-conf.int. 0.51965 - 0.52678)
1 ROUGE-L Average_P: 0.13849 (95%-conf.int. 0.13678 - 0.14012)
1 ROUGE-L Average_F: 0.19826 (95%-conf.int. 0.19666 - 0.20002)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.15642 (95%-conf.int. 0.15407 - 0.15864)
1 ROUGE-S4 Average_P: 0.03890 (95%-conf.int. 0.03805 - 0.03976)
1 ROUGE-S4 Average_F: 0.05619 (95%-conf.int. 0.05520 - 0.05722)
```

Figure 4.10: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.9: Referring Figure 4.9

Table	Average_R
ROUGE-1	77.8 ± 0.8
ROUGE-L	77.1 ± 0.8

Table 4.10: Referring Figure 4.10

Table	Average_R
ROUGE-1	56.0 ± 0.7
ROUGE-L	52.3 ± 0.7

4.1.2.3 Case 3

Model Input Parameters:

- allow_soft_placement = True
- batch_size = 100
- checkpoint_every = 2000
- data_file = ./data/cnn_train.txt
- dev_sample_% = 0.1
- dropout_keep_prob = 0.5
- embedding_size = 128
- evaluate_every = 100
- filter_sizes = $3, 5, 7, 9$
- l2_reg_lambda = 0.05
- log_device_placement = False
- num_checpoints = 5
- num_epochs = 100
- num_filters = 64

Other parameters:

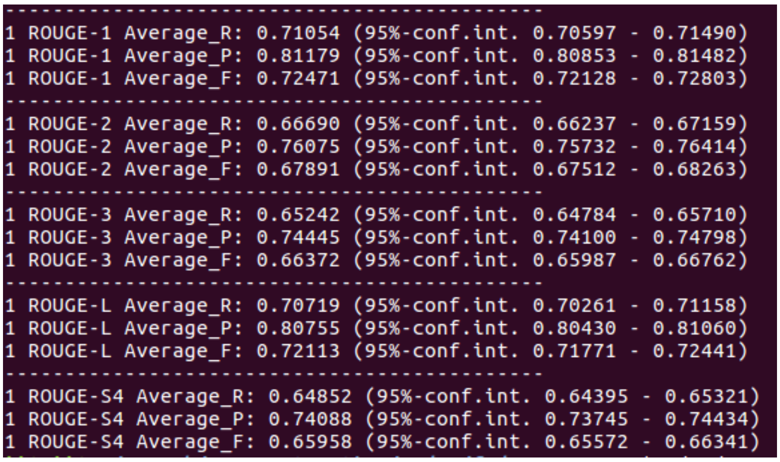
- max_sentence_length = 60
- learning_rate $\eta = 1e^{-3}$
- Train/Dev sample number = 3647781/405309

Evaluation on Test Data

- number of test samples = 211378
- **Accuracy** = 0.762473 = 76.2473%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.11
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.12



```
---------------------------------------------
1 ROUGE-1 Average_R: 0.71054 (95%-conf.int. 0.70597 - 0.71490)
1 ROUGE-1 Average_P: 0.81179 (95%-conf.int. 0.80853 - 0.81482)
1 ROUGE-1 Average_F: 0.72471 (95%-conf.int. 0.72128 - 0.72803)
---------------------------------------------
1 ROUGE-2 Average_R: 0.66690 (95%-conf.int. 0.66237 - 0.67159)
1 ROUGE-2 Average_P: 0.76075 (95%-conf.int. 0.75732 - 0.76414)
1 ROUGE-2 Average_F: 0.67891 (95%-conf.int. 0.67512 - 0.68263)
---------------------------------------------
1 ROUGE-3 Average_R: 0.65242 (95%-conf.int. 0.64784 - 0.65710)
1 ROUGE-3 Average_P: 0.74445 (95%-conf.int. 0.74100 - 0.74798)
1 ROUGE-3 Average_F: 0.66372 (95%-conf.int. 0.65987 - 0.66762)
---------------------------------------------
1 ROUGE-L Average_R: 0.70719 (95%-conf.int. 0.70261 - 0.71158)
1 ROUGE-L Average_P: 0.80755 (95%-conf.int. 0.80430 - 0.81060)
1 ROUGE-L Average_F: 0.72113 (95%-conf.int. 0.71771 - 0.72441)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.64852 (95%-conf.int. 0.64395 - 0.65321)
1 ROUGE-S4 Average_P: 0.74088 (95%-conf.int. 0.73745 - 0.74434)
1 ROUGE-S4 Average_F: 0.65958 (95%-conf.int. 0.65572 - 0.66341)
```

Figure 4.11: Reference: Original Extractive and Predicted: Predicted Extractive

Table 4.11: Referring Figure 4.11

Table	Average_R
ROUGE-1	71.7 ± 0.9
ROUGE-L	70.7 ± 0.9

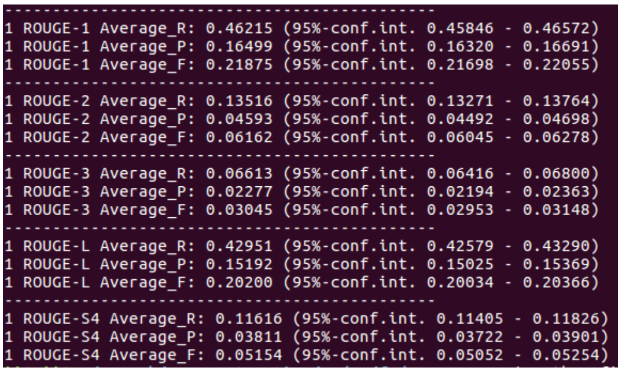

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.46215 (95%-conf.int. 0.45846 - 0.46572)
1 ROUGE-1 Average_P: 0.16499 (95%-conf.int. 0.16320 - 0.16691)
1 ROUGE-1 Average_F: 0.21875 (95%-conf.int. 0.21698 - 0.22055)
---------------------------------------------
1 ROUGE-2 Average_R: 0.13516 (95%-conf.int. 0.13271 - 0.13764)
1 ROUGE-2 Average_P: 0.04593 (95%-conf.int. 0.04492 - 0.04698)
1 ROUGE-2 Average_F: 0.06162 (95%-conf.int. 0.06045 - 0.06278)
---------------------------------------------
1 ROUGE-3 Average_R: 0.06613 (95%-conf.int. 0.06416 - 0.06800)
1 ROUGE-3 Average_P: 0.02277 (95%-conf.int. 0.02194 - 0.02363)
1 ROUGE-3 Average_F: 0.03045 (95%-conf.int. 0.02953 - 0.03148)
---------------------------------------------
1 ROUGE-L Average_R: 0.42951 (95%-conf.int. 0.42579 - 0.43290)
1 ROUGE-L Average_P: 0.15192 (95%-conf.int. 0.15025 - 0.15369)
1 ROUGE-L Average_F: 0.20200 (95%-conf.int. 0.20034 - 0.20366)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.11616 (95%-conf.int. 0.11405 - 0.11826)
1 ROUGE-S4 Average_P: 0.03811 (95%-conf.int. 0.03722 - 0.03901)
1 ROUGE-S4 Average_F: 0.05154 (95%-conf.int. 0.05052 - 0.05254)
```

Figure 4.12: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.12: Referring Figure 4.12

Table	Average_R
ROUGE-1	46.2 ± 0.7
ROUGE-L	43.0 ± 0.7

4.2 Linear SVM based Model

Refer to the section 3.3 for description of the Linear SVM based model.

4.2.1 CNN dataset

Refer subsection 3.1.1 to know details about the CNN dataset. Refer to the three cases described in subsection 3.1.3

4.2.1.1 Case 1

Model Input Parameters:

- Number of training samples = 1000000
- Models built with the following C values = $\{1, 10, 100, 1000, 10000, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$
- Input feature Vector Size = 256

Evaluation on Test Data

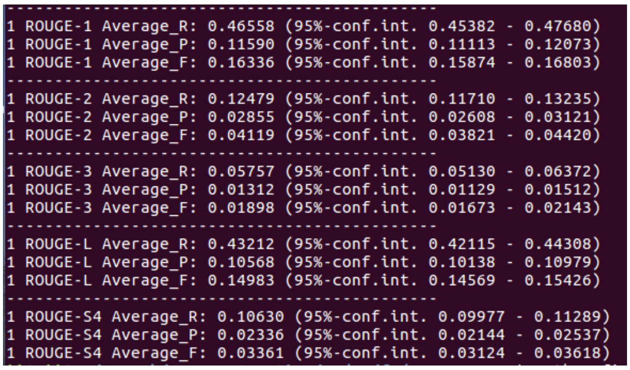
- number of test samples = 33046
- **Maximum Accuracy is given by the model with C value =** 0.0001
- **Accuracy** = 0.682382 = 68.2382%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.13
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.14

```
1 ROUGE-1 Average_R: 0.57972 (95%-conf.int. 0.56645 - 0.59253)
1 ROUGE-1 Average_P: 0.72759 (95%-conf.int. 0.71658 - 0.73806)
1 ROUGE-1 Average_F: 0.60558 (95%-conf.int. 0.59529 - 0.61529)
---------------------------------------------
1 ROUGE-2 Average_R: 0.49352 (95%-conf.int. 0.48146 - 0.50565)
1 ROUGE-2 Average_P: 0.62085 (95%-conf.int. 0.60798 - 0.63279)
1 ROUGE-2 Average_F: 0.51439 (95%-conf.int. 0.50369 - 0.52488)
---------------------------------------------
1 ROUGE-3 Average_R: 0.46828 (95%-conf.int. 0.45625 - 0.48033)
1 ROUGE-3 Average_P: 0.58923 (95%-conf.int. 0.57625 - 0.60137)
1 ROUGE-3 Average_F: 0.48746 (95%-conf.int. 0.47669 - 0.49806)
---------------------------------------------
1 ROUGE-L Average_R: 0.57349 (95%-conf.int. 0.56014 - 0.58624)
1 ROUGE-L Average_P: 0.71938 (95%-conf.int. 0.70809 - 0.72987)
1 ROUGE-L Average_F: 0.59885 (95%-conf.int. 0.58841 - 0.60882)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.46573 (95%-conf.int. 0.45388 - 0.47761)
1 ROUGE-S4 Average_P: 0.58651 (95%-conf.int. 0.57342 - 0.59832)
1 ROUGE-S4 Average_F: 0.48445 (95%-conf.int. 0.47370 - 0.49476)
```

Figure 4.13: Reference: Original Extractive and Predicted: Predicted Extractive



```
---------------------------------------------
1 ROUGE-1 Average_R: 0.46558 (95%-conf.int. 0.45382 - 0.47680)
1 ROUGE-1 Average_P: 0.11590 (95%-conf.int. 0.11113 - 0.12073)
1 ROUGE-1 Average_F: 0.16336 (95%-conf.int. 0.15874 - 0.16803)
---------------------------------------------
1 ROUGE-2 Average_R: 0.12479 (95%-conf.int. 0.11710 - 0.13235)
1 ROUGE-2 Average_P: 0.02855 (95%-conf.int. 0.02608 - 0.03121)
1 ROUGE-2 Average_F: 0.04119 (95%-conf.int. 0.03821 - 0.04420)
---------------------------------------------
1 ROUGE-3 Average_R: 0.05757 (95%-conf.int. 0.05130 - 0.06372)
1 ROUGE-3 Average_P: 0.01312 (95%-conf.int. 0.01129 - 0.01512)
1 ROUGE-3 Average_F: 0.01898 (95%-conf.int. 0.01673 - 0.02143)
---------------------------------------------
1 ROUGE-L Average_R: 0.43212 (95%-conf.int. 0.42115 - 0.44308)
1 ROUGE-L Average_P: 0.10568 (95%-conf.int. 0.10138 - 0.10979)
1 ROUGE-L Average_F: 0.14983 (95%-conf.int. 0.14569 - 0.15426)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.10630 (95%-conf.int. 0.09977 - 0.11289)
1 ROUGE-S4 Average_P: 0.02336 (95%-conf.int. 0.02144 - 0.02537)
1 ROUGE-S4 Average_F: 0.03361 (95%-conf.int. 0.03124 - 0.03618)
```

Figure 4.14: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.13: Referring Figure 4.13

Table	Average_R
ROUGE-1	58.0 ± 2.6
ROUGE-L	57.3 ± 2.6

Table 4.14: Referring Figure 4.14

Table	Average_R
ROUGE-1	46.6 ± 2.3
ROUGE-L	43.2 ± 2.2

4.2.1.2 Case 2

Model Input Parameters:

- Number of training samples = 1000000
- Models built with the following C values = $\{1, 10, 100, 1000, 10000, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$
- Input feature Vector Size = 256

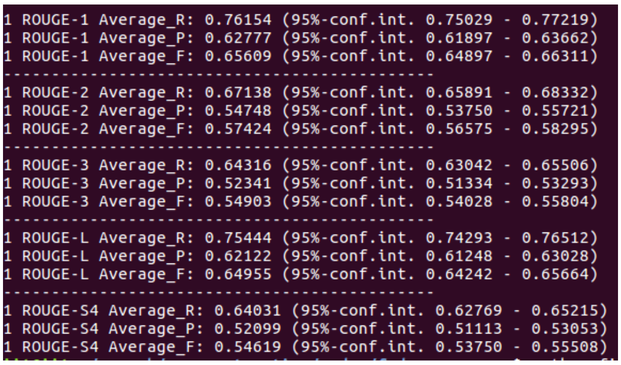
Evaluation on Test Data

- number of test samples = 33046

- **Maximum Accuracy is given by the model with C value =** 0.0001
- **Accuracy** = 0.567028 = 56.7028%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.15
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.16

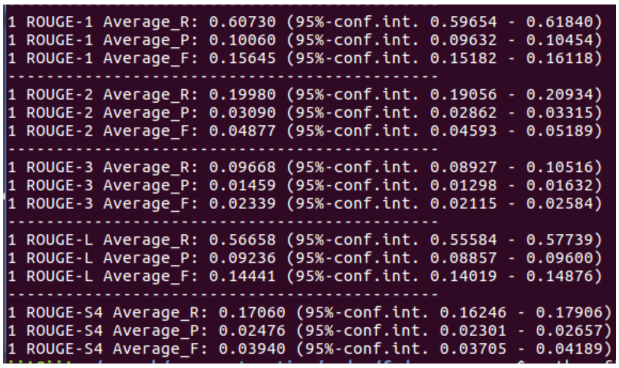


```
1 ROUGE-1 Average_R: 0.76154 (95%-conf.int. 0.75029 - 0.77219)
1 ROUGE-1 Average_P: 0.62777 (95%-conf.int. 0.61897 - 0.63662)
1 ROUGE-1 Average_F: 0.65609 (95%-conf.int. 0.64897 - 0.66311)
---------------------------------------------
1 ROUGE-2 Average_R: 0.67138 (95%-conf.int. 0.65891 - 0.68332)
1 ROUGE-2 Average_P: 0.54748 (95%-conf.int. 0.53750 - 0.55721)
1 ROUGE-2 Average_F: 0.57424 (95%-conf.int. 0.56575 - 0.58295)
---------------------------------------------
1 ROUGE-3 Average_R: 0.64316 (95%-conf.int. 0.63042 - 0.65506)
1 ROUGE-3 Average_P: 0.52341 (95%-conf.int. 0.51334 - 0.53293)
1 ROUGE-3 Average_F: 0.54903 (95%-conf.int. 0.54028 - 0.55804)
---------------------------------------------
1 ROUGE-L Average_R: 0.75444 (95%-conf.int. 0.74293 - 0.76512)
1 ROUGE-L Average_P: 0.62122 (95%-conf.int. 0.61248 - 0.63028)
1 ROUGE-L Average_F: 0.64955 (95%-conf.int. 0.64242 - 0.65664)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.64031 (95%-conf.int. 0.62769 - 0.65215)
1 ROUGE-S4 Average_P: 0.52099 (95%-conf.int. 0.51113 - 0.53053)
1 ROUGE-S4 Average_F: 0.54619 (95%-conf.int. 0.53750 - 0.55508)
```

Figure 4.15: Reference: Original Extractive and Predicted: Predicted Extractive

Table 4.15: Referring Figure 4.15

Table	Average_R
ROUGE-1	76.2 ± 2.2
ROUGE-L	75.4 ± 2.2



```
1 ROUGE-1 Average_R: 0.60730 (95%-conf.int. 0.59654 - 0.61840)
1 ROUGE-1 Average_P: 0.10060 (95%-conf.int. 0.09632 - 0.10454)
1 ROUGE-1 Average_F: 0.15645 (95%-conf.int. 0.15182 - 0.16118)
---------------------------------------------
1 ROUGE-2 Average_R: 0.19980 (95%-conf.int. 0.19056 - 0.20934)
1 ROUGE-2 Average_P: 0.03090 (95%-conf.int. 0.02862 - 0.03315)
1 ROUGE-2 Average_F: 0.04877 (95%-conf.int. 0.04593 - 0.05189)
---------------------------------------------
1 ROUGE-3 Average_R: 0.09668 (95%-conf.int. 0.08927 - 0.10516)
1 ROUGE-3 Average_P: 0.01459 (95%-conf.int. 0.01298 - 0.01632)
1 ROUGE-3 Average_F: 0.02339 (95%-conf.int. 0.02115 - 0.02584)
---------------------------------------------
1 ROUGE-L Average_R: 0.56658 (95%-conf.int. 0.55584 - 0.57739)
1 ROUGE-L Average_P: 0.09236 (95%-conf.int. 0.08857 - 0.09600)
1 ROUGE-L Average_F: 0.14441 (95%-conf.int. 0.14019 - 0.14876)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.17060 (95%-conf.int. 0.16246 - 0.17906)
1 ROUGE-S4 Average_P: 0.02476 (95%-conf.int. 0.02301 - 0.02657)
1 ROUGE-S4 Average_F: 0.03940 (95%-conf.int. 0.03705 - 0.04189)
```

Figure 4.16: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.16: Referring Figure 4.16

Table	Average_R
ROUGE-1	60.7 ± 2.2
ROUGE-L	56.7 ± 2.2

4.2.1.3 Case 3

Model Input Parameters:

- Number of training samples = 1000000
- Models built with the following C values = $\{1, 10, 100, 1000, 10000, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$
- Input feature Vector Size = 256

Evaluation on Test Data

- number of test samples = 26452
- **Maximum Accuracy is given by the model with C value** = 0.5
- **Accuracy** = 0.709927 = 70.9927%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.17

2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.18

```
1 ROUGE-1 Average_R: 0.68335 (95%-conf.int. 0.67098 - 0.69597)
1 ROUGE-1 Average_P: 0.76826 (95%-conf.int. 0.75880 - 0.77852)
1 ROUGE-1 Average_F: 0.69199 (95%-conf.int. 0.68231 - 0.70136)
---------------------------------------------
1 ROUGE-2 Average_R: 0.61768 (95%-conf.int. 0.60472 - 0.63108)
1 ROUGE-2 Average_P: 0.69335 (95%-conf.int. 0.68172 - 0.70492)
1 ROUGE-2 Average_F: 0.62441 (95%-conf.int. 0.61357 - 0.63493)
---------------------------------------------
1 ROUGE-3 Average_R: 0.59634 (95%-conf.int. 0.58342 - 0.61002)
1 ROUGE-3 Average_P: 0.66929 (95%-conf.int. 0.65764 - 0.68118)
1 ROUGE-3 Average_F: 0.60238 (95%-conf.int. 0.59124 - 0.61329)
---------------------------------------------
1 ROUGE-L Average_R: 0.67861 (95%-conf.int. 0.66593 - 0.69130)
1 ROUGE-L Average_P: 0.76259 (95%-conf.int. 0.75294 - 0.77279)
1 ROUGE-L Average_F: 0.68710 (95%-conf.int. 0.67731 - 0.69667)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.59153 (95%-conf.int. 0.57876 - 0.60509)
1 ROUGE-S4 Average_P: 0.66423 (95%-conf.int. 0.65286 - 0.67621)
1 ROUGE-S4 Average_F: 0.59717 (95%-conf.int. 0.58624 - 0.60817)
```

Figure 4.17: Reference: Original Extractive and Predicted: Predicted Extractive

```
1 ROUGE-1 Average_R: 0.52057 (95%-conf.int. 0.50926 - 0.53203)
1 ROUGE-1 Average_P: 0.11412 (95%-conf.int. 0.10997 - 0.11861)
1 ROUGE-1 Average_F: 0.16958 (95%-conf.int. 0.16465 - 0.17479)
---------------------------------------------
1 ROUGE-2 Average_R: 0.16241 (95%-conf.int. 0.15349 - 0.17123)
1 ROUGE-2 Average_P: 0.03269 (95%-conf.int. 0.03016 - 0.03526)
1 ROUGE-2 Average_F: 0.04967 (95%-conf.int. 0.04647 - 0.05301)
---------------------------------------------
1 ROUGE-3 Average_R: 0.08229 (95%-conf.int. 0.07493 - 0.08979)
1 ROUGE-3 Average_P: 0.01613 (95%-conf.int. 0.01422 - 0.01810)
1 ROUGE-3 Average_F: 0.02482 (95%-conf.int. 0.02222 - 0.02743)
---------------------------------------------
1 ROUGE-L Average_R: 0.48222 (95%-conf.int. 0.47082 - 0.49316)
1 ROUGE-L Average_P: 0.10389 (95%-conf.int. 0.10024 - 0.10801)
1 ROUGE-L Average_F: 0.15523 (95%-conf.int. 0.15079 - 0.15984)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.13712 (95%-conf.int. 0.12971 - 0.14478)
1 ROUGE-S4 Average_P: 0.02612 (95%-conf.int. 0.02408 - 0.02815)
1 ROUGE-S4 Average_F: 0.03999 (95%-conf.int. 0.03734 - 0.04277)
```

Figure 4.18: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.17: Referring Figure 4.17

Table	Average_R
ROUGE-1	68.3 ± 2.5
ROUGE-L	67.9 ± 2.5

Table 4.18: Referring Figure 4.18

Table	Average_R
ROUGE-1	52.1 ± 2.3
ROUGE-L	48.2 ± 2.2

4.2.2 Dailymail dataset

Refer subsection 3.1.2 to know details about the Dailymail dataset. Refer to the three cases described in subsection 3.1.3

4.2.2.1 Case 1

Model Input Parameters:

- Number of training samples = 1000000
- Models built with the following C values = $\{1, 10, 100, 1000, 10000, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$
- Input feature Vector Size = 256

Evaluation on Test Data

- number of test samples = 264305
- **Maximum Accuracy is given by the model with C value** = 0.5
- **Accuracy** = 0.719805 = 71.9805%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.19

2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.20

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.60051 (95%-conf.int. 0.59545 - 0.60537)
1 ROUGE-1 Average_P: 0.74584 (95%-conf.int. 0.74230 - 0.74941)
1 ROUGE-1 Average_F: 0.62101 (95%-conf.int. 0.61725 - 0.62464)
---------------------------------------------
1 ROUGE-2 Average_R: 0.53515 (95%-conf.int. 0.53011 - 0.53998)
1 ROUGE-2 Average_P: 0.66253 (95%-conf.int. 0.65834 - 0.66674)
1 ROUGE-2 Average_F: 0.55105 (95%-conf.int. 0.54713 - 0.55497)
---------------------------------------------
1 ROUGE-3 Average_R: 0.51575 (95%-conf.int. 0.51069 - 0.52057)
1 ROUGE-3 Average_P: 0.63909 (95%-conf.int. 0.63478 - 0.64344)
1 ROUGE-3 Average_F: 0.53042 (95%-conf.int. 0.52642 - 0.53437)
---------------------------------------------
1 ROUGE-L Average_R: 0.59528 (95%-conf.int. 0.59019 - 0.60018)
1 ROUGE-L Average_P: 0.73843 (95%-conf.int. 0.73480 - 0.74202)
1 ROUGE-L Average_F: 0.61526 (95%-conf.int. 0.61150 - 0.61901)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.51313 (95%-conf.int. 0.50814 - 0.51795)
1 ROUGE-S4 Average_P: 0.63767 (95%-conf.int. 0.63351 - 0.64186)
1 ROUGE-S4 Average_F: 0.52766 (95%-conf.int. 0.52371 - 0.53149)
```

Figure 4.19: Reference: Original Extractive and Predicted: Predicted Extractive

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.41781 (95%-conf.int. 0.41412 - 0.42126)
1 ROUGE-1 Average_P: 0.16789 (95%-conf.int. 0.16585 - 0.16982)
1 ROUGE-1 Average_F: 0.20969 (95%-conf.int. 0.20781 - 0.21151)
---------------------------------------------
1 ROUGE-2 Average_R: 0.10636 (95%-conf.int. 0.10434 - 0.10835)
1 ROUGE-2 Average_P: 0.04273 (95%-conf.int. 0.04161 - 0.04384)
1 ROUGE-2 Average_F: 0.05291 (95%-conf.int. 0.05177 - 0.05400)
---------------------------------------------
1 ROUGE-3 Average_R: 0.04768 (95%-conf.int. 0.04614 - 0.04918)
1 ROUGE-3 Average_P: 0.02032 (95%-conf.int. 0.01942 - 0.02123)
1 ROUGE-3 Average_F: 0.02460 (95%-conf.int. 0.02371 - 0.02550)
---------------------------------------------
1 ROUGE-L Average_R: 0.38766 (95%-conf.int. 0.38425 - 0.39098)
1 ROUGE-L Average_P: 0.15417 (95%-conf.int. 0.15227 - 0.15603)
1 ROUGE-L Average_F: 0.19321 (95%-conf.int. 0.19142 - 0.19488)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.09283 (95%-conf.int. 0.09111 - 0.09457)
1 ROUGE-S4 Average_P: 0.03581 (95%-conf.int. 0.03483 - 0.03675)
1 ROUGE-S4 Average_F: 0.04471 (95%-conf.int. 0.04374 - 0.04562)
```

Figure 4.20: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.19: Referring Figure 4.19

Table	Average_R
ROUGE-1	60.1 ± 1.0
ROUGE-L	59.5 ± 1.0

Table 4.20: Referring Figure 4.20

Table	Average_R
ROUGE-1	41.8 ± 0.7
ROUGE-L	38.8 ± 0.7

4.2.2.2 Case 2

Model Input Parameters:

- Number of training samples = 1000000
- Models built with the following C values = $\{1, 10, 100, 1000, 10000, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$
- Input feature Vector Size = 256

Evaluation on Test Data

- number of test samples = 264305
- **Maximum Accuracy is given by the model with C value =** 0.5
- **Accuracy =** 0.608974 = 60.8974%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.21
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.22

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.75823 (95%-conf.int. 0.75394 - 0.76253)
1 ROUGE-1 Average_P: 0.66853 (95%-conf.int. 0.66536 - 0.67174)
1 ROUGE-1 Average_F: 0.67562 (95%-conf.int. 0.67269 - 0.67824)
---------------------------------------------
1 ROUGE-2 Average_R: 0.68713 (95%-conf.int. 0.68263 - 0.69172)
1 ROUGE-2 Average_P: 0.59955 (95%-conf.int. 0.59624 - 0.60305)
1 ROUGE-2 Average_F: 0.60826 (95%-conf.int. 0.60498 - 0.61134)
---------------------------------------------
1 ROUGE-3 Average_R: 0.66493 (95%-conf.int. 0.66041 - 0.66952)
1 ROUGE-3 Average_P: 0.57945 (95%-conf.int. 0.57611 - 0.58304)
1 ROUGE-3 Average_F: 0.58771 (95%-conf.int. 0.58438 - 0.59083)
---------------------------------------------
1 ROUGE-L Average_R: 0.75204 (95%-conf.int. 0.74769 - 0.75639)
1 ROUGE-L Average_P: 0.66188 (95%-conf.int. 0.65874 - 0.66508)
1 ROUGE-L Average_F: 0.66957 (95%-conf.int. 0.66646 - 0.67217)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.66250 (95%-conf.int. 0.65799 - 0.66693)
1 ROUGE-S4 Average_P: 0.57812 (95%-conf.int. 0.57483 - 0.58169)
1 ROUGE-S4 Average_F: 0.58548 (95%-conf.int. 0.58226 - 0.58854)
```

Figure 4.21: Reference: Original Extractive and Predicted: Predicted Extractive

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.54251 (95%-conf.int. 0.53897 - 0.54610)
1 ROUGE-1 Average_P: 0.15311 (95%-conf.int. 0.15129 - 0.15504)
1 ROUGE-1 Average_F: 0.21513 (95%-conf.int. 0.21335 - 0.21708)
---------------------------------------------
1 ROUGE-2 Average_R: 0.17142 (95%-conf.int. 0.16881 - 0.17381)
1 ROUGE-2 Average_P: 0.04722 (95%-conf.int. 0.04624 - 0.04828)
1 ROUGE-2 Average_F: 0.06647 (95%-conf.int. 0.06537 - 0.06762)
---------------------------------------------
1 ROUGE-3 Average_R: 0.08164 (95%-conf.int. 0.07946 - 0.08380)
1 ROUGE-3 Average_P: 0.02331 (95%-conf.int. 0.02251 - 0.02413)
1 ROUGE-3 Average_F: 0.03239 (95%-conf.int. 0.03149 - 0.03336)
---------------------------------------------
1 ROUGE-L Average_R: 0.50664 (95%-conf.int. 0.50308 - 0.51021)
1 ROUGE-L Average_P: 0.14143 (95%-conf.int. 0.13973 - 0.14320)
1 ROUGE-L Average_F: 0.19939 (95%-conf.int. 0.19768 - 0.20117)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.14730 (95%-conf.int. 0.14500 - 0.14961)
1 ROUGE-S4 Average_P: 0.03882 (95%-conf.int. 0.03798 - 0.03970)
1 ROUGE-S4 Average_F: 0.05513 (95%-conf.int. 0.05416 - 0.05611)
```

Figure 4.22: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.21: Referring Figure 4.21

Table	Average_R
ROUGE-1	75.8 ± 0.9
ROUGE-L	75.2 ± 0.9

Table 4.22: Referring Figure 4.22

Table	Average_R
ROUGE-1	54.3 ± 0.7
ROUGE-L	50.7 ± 0.7

4.2.2.3 Case 3

Model Input Parameters:

- Number of training samples = 1000000
- Models built with the following C values = $\{1, 10, 100, 1000, 10000, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001\}$
- Input feature Vector Size = 256

Evaluation on Test Data

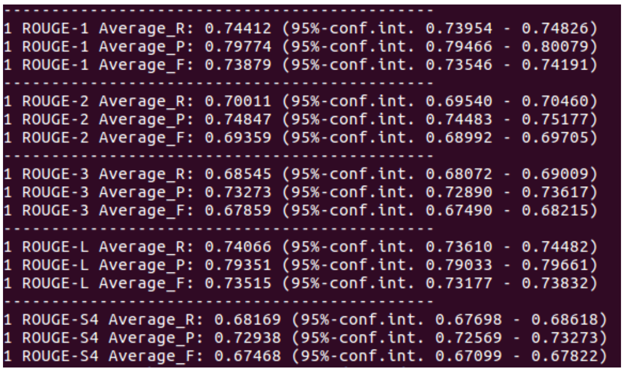
- number of test samples = 211378
- **Maximum Accuracy is given by the model with C value** = 0.0001
- **Accuracy** = 0.762771 = 76.2771%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.23
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.24

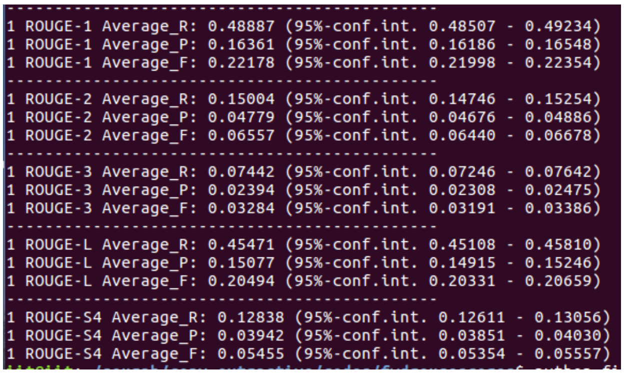
Table 4.23: Referring Figure 4.23

Table	Average_R
ROUGE-1	74.4 ± 0.9
ROUGE-L	74.1 ± 0.9



```
--------------------------------------------
1 ROUGE-1 Average_R: 0.74412 (95%-conf.int. 0.73954 - 0.74826)
1 ROUGE-1 Average_P: 0.79774 (95%-conf.int. 0.79466 - 0.80079)
1 ROUGE-1 Average_F: 0.73879 (95%-conf.int. 0.73546 - 0.74191)
--------------------------------------------
1 ROUGE-2 Average_R: 0.70011 (95%-conf.int. 0.69540 - 0.70460)
1 ROUGE-2 Average_P: 0.74847 (95%-conf.int. 0.74483 - 0.75177)
1 ROUGE-2 Average_F: 0.69359 (95%-conf.int. 0.68992 - 0.69705)
--------------------------------------------
1 ROUGE-3 Average_R: 0.68545 (95%-conf.int. 0.68072 - 0.69009)
1 ROUGE-3 Average_P: 0.73273 (95%-conf.int. 0.72890 - 0.73617)
1 ROUGE-3 Average_F: 0.67859 (95%-conf.int. 0.67490 - 0.68215)
--------------------------------------------
1 ROUGE-L Average_R: 0.74066 (95%-conf.int. 0.73610 - 0.74482)
1 ROUGE-L Average_P: 0.79351 (95%-conf.int. 0.79033 - 0.79661)
1 ROUGE-L Average_F: 0.73515 (95%-conf.int. 0.73177 - 0.73832)
--------------------------------------------
1 ROUGE-S4 Average_R: 0.68169 (95%-conf.int. 0.67698 - 0.68618)
1 ROUGE-S4 Average_P: 0.72938 (95%-conf.int. 0.72569 - 0.73273)
1 ROUGE-S4 Average_F: 0.67468 (95%-conf.int. 0.67099 - 0.67822)
```

Figure 4.23: Reference: Original Extractive and Predicted: Predicted Extractive



```
--------------------------------------------
1 ROUGE-1 Average_R: 0.48887 (95%-conf.int. 0.48507 - 0.49234)
1 ROUGE-1 Average_P: 0.16361 (95%-conf.int. 0.16186 - 0.16548)
1 ROUGE-1 Average_F: 0.22178 (95%-conf.int. 0.21998 - 0.22354)
--------------------------------------------
1 ROUGE-2 Average_R: 0.15004 (95%-conf.int. 0.14746 - 0.15254)
1 ROUGE-2 Average_P: 0.04779 (95%-conf.int. 0.04676 - 0.04886)
1 ROUGE-2 Average_F: 0.06557 (95%-conf.int. 0.06440 - 0.06678)
--------------------------------------------
1 ROUGE-3 Average_R: 0.07442 (95%-conf.int. 0.07246 - 0.07642)
1 ROUGE-3 Average_P: 0.02394 (95%-conf.int. 0.02308 - 0.02475)
1 ROUGE-3 Average_F: 0.03284 (95%-conf.int. 0.03191 - 0.03386)
--------------------------------------------
1 ROUGE-L Average_R: 0.45471 (95%-conf.int. 0.45108 - 0.45810)
1 ROUGE-L Average_P: 0.15077 (95%-conf.int. 0.14915 - 0.15246)
1 ROUGE-L Average_F: 0.20494 (95%-conf.int. 0.20331 - 0.20659)
--------------------------------------------
1 ROUGE-S4 Average_R: 0.12838 (95%-conf.int. 0.12611 - 0.13056)
1 ROUGE-S4 Average_P: 0.03942 (95%-conf.int. 0.03851 - 0.04030)
1 ROUGE-S4 Average_F: 0.05455 (95%-conf.int. 0.05354 - 0.05557)
```

Figure 4.24: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.24: Referring Figure 4.24

Table	Average_R
ROUGE-1	48.9 ± 0.7
ROUGE-L	45.5 ± 0.7

4.3 Bidirectional RNN based Model

Refer to the section 3.4 for description of the Bidirectional RNN based model.

4.3.1 CNN dataset

Refer subsection 3.1.1 to know details about the CNN dataset. Refer to the three cases described in subsection 3.1.3

4.3.1.1 Case 1

Model Input Parameters:

- Input Feature Vector Size = 256
- batch_size = 100
- checkpoint_every = 2000
- dropout_keep_prob = 0.5
- evaluate_every = 100
- num_checkpoints = 5
- num_epochs = 100
- num_layers = 2
- truncated_backprop_length = 40

Some other parameters:

- learning rate $\eta = 0.3$
- optimizer = AdagradOptimizer
- Number of training samples = 2492444
- state size = input feature vector size

Evaluation on Test Data

- Number of test samples = 33000

- **Accuracy** = 0.704 =70.4%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.25

2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.26

```
1 ROUGE-1 Average_R: 0.53884 (95%-conf.int. 0.52673 - 0.55144)
1 ROUGE-1 Average_P: 0.75569 (95%-conf.int. 0.74457 - 0.76552)
1 ROUGE-1 Average_F: 0.59979 (95%-conf.int. 0.58932 - 0.61051)
---------------------------------------------
1 ROUGE-2 Average_R: 0.45601 (95%-conf.int. 0.44350 - 0.46854)
1 ROUGE-2 Average_P: 0.64071 (95%-conf.int. 0.62710 - 0.65270)
1 ROUGE-2 Average_F: 0.50643 (95%-conf.int. 0.49455 - 0.51746)
---------------------------------------------
1 ROUGE-3 Average_R: 0.43278 (95%-conf.int. 0.42018 - 0.44518)
1 ROUGE-3 Average_P: 0.60914 (95%-conf.int. 0.59515 - 0.62143)
1 ROUGE-3 Average_F: 0.48013 (95%-conf.int. 0.46836 - 0.49155)
---------------------------------------------
1 ROUGE-L Average_R: 0.53295 (95%-conf.int. 0.52067 - 0.54540)
1 ROUGE-L Average_P: 0.74698 (95%-conf.int. 0.73556 - 0.75713)
1 ROUGE-L Average_F: 0.59304 (95%-conf.int. 0.58239 - 0.60370)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.43015 (95%-conf.int. 0.41780 - 0.44293)
1 ROUGE-S4 Average_P: 0.60727 (95%-conf.int. 0.59383 - 0.61910)
1 ROUGE-S4 Average_F: 0.47715 (95%-conf.int. 0.46533 - 0.48842)
```

Figure 4.25: Reference: Original Extractive and Predicted: Predicted Extractive

```
1 ROUGE-1 Average_R: 0.43528 (95%-conf.int. 0.42441 - 0.44608)
1 ROUGE-1 Average_P: 0.11784 (95%-conf.int. 0.11277 - 0.12322)
1 ROUGE-1 Average_F: 0.16475 (95%-conf.int. 0.15963 - 0.16944)
---------------------------------------------
1 ROUGE-2 Average_R: 0.10600 (95%-conf.int. 0.09897 - 0.11308)
1 ROUGE-2 Average_P: 0.02758 (95%-conf.int. 0.02502 - 0.03042)
1 ROUGE-2 Average_F: 0.03904 (95%-conf.int. 0.03596 - 0.04218)
---------------------------------------------
1 ROUGE-3 Average_R: 0.04683 (95%-conf.int. 0.04139 - 0.05268)
1 ROUGE-3 Average_P: 0.01277 (95%-conf.int. 0.01067 - 0.01509)
1 ROUGE-3 Average_F: 0.01771 (95%-conf.int. 0.01538 - 0.02021)
---------------------------------------------
1 ROUGE-L Average_R: 0.40201 (95%-conf.int. 0.39111 - 0.41226)
1 ROUGE-L Average_P: 0.10741 (95%-conf.int. 0.10278 - 0.11238)
1 ROUGE-L Average_F: 0.15076 (95%-conf.int. 0.14611 - 0.15516)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.09014 (95%-conf.int. 0.08427 - 0.09602)
1 ROUGE-S4 Average_P: 0.02254 (95%-conf.int. 0.02039 - 0.02516)
1 ROUGE-S4 Average_F: 0.03168 (95%-conf.int. 0.02930 - 0.03424)
```

Figure 4.26: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.25: Referring Figure 4.25

Table	Average_R
ROUGE-1	53.9 ± 2.5
ROUGE-L	53.3 ± 2.5

Table 4.26: Referring Figure 4.26

Table	Average_R
ROUGE-1	43.5 ± 2.2
ROUGE-L	40.2 ± 2.1

4.3.1.2 Case 2

Model Input Parameters:

- Input Feature Vector Size = 256
- batch_size = 100
- checkpoint_every = 2000
- dropout_keep_prob = 0.5
- evaluate_every = 100
- num_checkpoints = 5
- num_epochs = 100
- num_layers = 2
- truncated_backprop_length = 40

Some other parameters:

- learning rate $\eta = 0.3$
- optimizer = AdagradOptimizer

- Number of training samples = 2492444
- state size = input feature vector size

Evaluation on Test Data

- Number of test samples = 33000
- **Accuracy** = 0.587 =58.7%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.27
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.28

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.73769 (95%-conf.int. 0.72558 - 0.75018)
1 ROUGE-1 Average_P: 0.66381 (95%-conf.int. 0.65579 - 0.67211)
1 ROUGE-1 Average_F: 0.67549 (95%-conf.int. 0.66752 - 0.68379)
---------------------------------------------
1 ROUGE-2 Average_R: 0.64300 (95%-conf.int. 0.63049 - 0.65633)
1 ROUGE-2 Average_P: 0.57483 (95%-conf.int. 0.56539 - 0.58419)
1 ROUGE-2 Average_F: 0.58607 (95%-conf.int. 0.57668 - 0.59537)
---------------------------------------------
1 ROUGE-3 Average_R: 0.61562 (95%-conf.int. 0.60276 - 0.62928)
1 ROUGE-3 Average_P: 0.54958 (95%-conf.int. 0.53979 - 0.55918)
1 ROUGE-3 Average_F: 0.56034 (95%-conf.int. 0.55052 - 0.57006)
---------------------------------------------
1 ROUGE-L Average_R: 0.73059 (95%-conf.int. 0.71839 - 0.74296)
1 ROUGE-L Average_P: 0.65695 (95%-conf.int. 0.64894 - 0.66539)
1 ROUGE-L Average_F: 0.66873 (95%-conf.int. 0.66058 - 0.67716)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.61399 (95%-conf.int. 0.60124 - 0.62747)
1 ROUGE-S4 Average_P: 0.54804 (95%-conf.int. 0.53871 - 0.55712)
1 ROUGE-S4 Average_F: 0.55854 (95%-conf.int. 0.54885 - 0.56807)
```

Figure 4.27: Reference: Original Extractive and Predicted: Predicted Extractive

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.56887 (95%-conf.int. 0.55790 - 0.57894)
1 ROUGE-1 Average_P: 0.09979 (95%-conf.int. 0.09566 - 0.10417)
1 ROUGE-1 Average_F: 0.15641 (95%-conf.int. 0.15163 - 0.16152)
---------------------------------------------
1 ROUGE-2 Average_R: 0.17060 (95%-conf.int. 0.16156 - 0.17927)
1 ROUGE-2 Average_P: 0.02873 (95%-conf.int. 0.02662 - 0.03108)
1 ROUGE-2 Average_F: 0.04539 (95%-conf.int. 0.04240 - 0.04849)
---------------------------------------------
1 ROUGE-3 Average_R: 0.07804 (95%-conf.int. 0.07075 - 0.08567)
1 ROUGE-3 Average_P: 0.01346 (95%-conf.int. 0.01182 - 0.01530)
1 ROUGE-3 Average_F: 0.02124 (95%-conf.int. 0.01885 - 0.02372)
---------------------------------------------
1 ROUGE-L Average_R: 0.53002 (95%-conf.int. 0.51898 - 0.54013)
1 ROUGE-L Average_P: 0.09169 (95%-conf.int. 0.08806 - 0.09542)
1 ROUGE-L Average_F: 0.14426 (95%-conf.int. 0.13997 - 0.14898)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.14475 (95%-conf.int. 0.13770 - 0.15210)
1 ROUGE-S4 Average_P: 0.02312 (95%-conf.int. 0.02148 - 0.02505)
1 ROUGE-S4 Average_F: 0.03667 (95%-conf.int. 0.03440 - 0.03924)
```

Figure 4.28: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.27: Referring Figure 4.27

Table	Average_R
ROUGE-1	73.8 ± 2.5
ROUGE-L	73.1 ± 2.5

Table 4.28: Referring Figure 4.28

Table	Average_R
ROUGE-1	56.9 ± 2.1
ROUGE-L	53.0 ± 2.1

4.3.1.3 Case 3

Model Input Parameters:

- Input Feature Vector Size = 256
- batch_size = 100
- checkpoint_every = 2000
- dropout_keep_prob = 0.5
- evaluate_every = 100
- num_checkpoints = 5

- num_epochs = 100
- num_layers = 2
- truncated_backprop_length = 40

Some other parameters:

- learning rate $\eta = 0.3$
- optimizer = AdagradOptimizer
- Number of training samples = 1993968
- state size = input feature vector size

Evaluation on Test Data

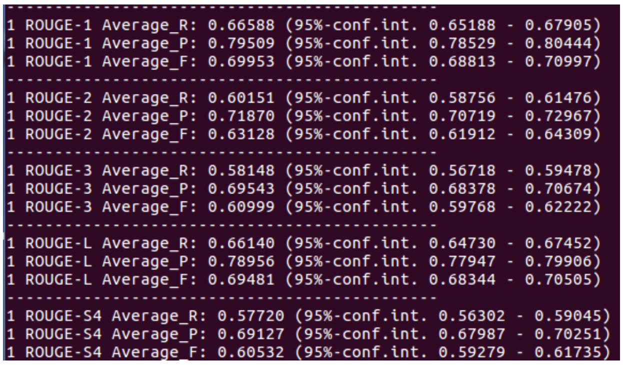
- Number of test samples = 24000
- **Accuracy** = 0.730333 =73.033%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.29
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.30

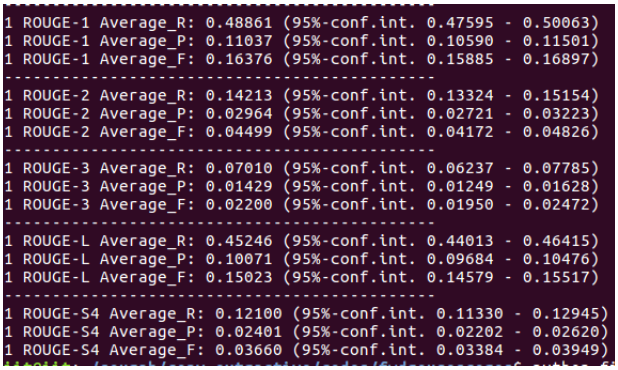
Table 4.29: Referring Figure 4.29

Table	Average_R
ROUGE-1	66.6 ± 2.7
ROUGE-L	66.1 ± 2.7



```
1 ROUGE-1 Average_R: 0.66588 (95%-conf.int. 0.65188 - 0.67905)
1 ROUGE-1 Average_P: 0.79509 (95%-conf.int. 0.78529 - 0.80444)
1 ROUGE-1 Average_F: 0.69953 (95%-conf.int. 0.68813 - 0.70997)
---------------------------------------------
1 ROUGE-2 Average_R: 0.60151 (95%-conf.int. 0.58756 - 0.61476)
1 ROUGE-2 Average_P: 0.71870 (95%-conf.int. 0.70719 - 0.72967)
1 ROUGE-2 Average_F: 0.63128 (95%-conf.int. 0.61912 - 0.64309)
---------------------------------------------
1 ROUGE-3 Average_R: 0.58148 (95%-conf.int. 0.56718 - 0.59478)
1 ROUGE-3 Average_P: 0.69543 (95%-conf.int. 0.68378 - 0.70674)
1 ROUGE-3 Average_F: 0.60999 (95%-conf.int. 0.59768 - 0.62222)
---------------------------------------------
1 ROUGE-L Average_R: 0.66140 (95%-conf.int. 0.64730 - 0.67452)
1 ROUGE-L Average_P: 0.78956 (95%-conf.int. 0.77947 - 0.79906)
1 ROUGE-L Average_F: 0.69481 (95%-conf.int. 0.68344 - 0.70505)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.57720 (95%-conf.int. 0.56302 - 0.59045)
1 ROUGE-S4 Average_P: 0.69127 (95%-conf.int. 0.67987 - 0.70251)
1 ROUGE-S4 Average_F: 0.60532 (95%-conf.int. 0.59279 - 0.61735)
```

Figure 4.29: Reference: Original Extractive and Predicted: Predicted Extractive



```
1 ROUGE-1 Average_R: 0.48861 (95%-conf.int. 0.47595 - 0.50063)
1 ROUGE-1 Average_P: 0.11037 (95%-conf.int. 0.10590 - 0.11501)
1 ROUGE-1 Average_F: 0.16376 (95%-conf.int. 0.15885 - 0.16897)
---------------------------------------------
1 ROUGE-2 Average_R: 0.14213 (95%-conf.int. 0.13324 - 0.15154)
1 ROUGE-2 Average_P: 0.02964 (95%-conf.int. 0.02721 - 0.03223)
1 ROUGE-2 Average_F: 0.04499 (95%-conf.int. 0.04172 - 0.04826)
---------------------------------------------
1 ROUGE-3 Average_R: 0.07010 (95%-conf.int. 0.06237 - 0.07785)
1 ROUGE-3 Average_P: 0.01429 (95%-conf.int. 0.01249 - 0.01628)
1 ROUGE-3 Average_F: 0.02200 (95%-conf.int. 0.01950 - 0.02472)
---------------------------------------------
1 ROUGE-L Average_R: 0.45246 (95%-conf.int. 0.44013 - 0.46415)
1 ROUGE-L Average_P: 0.10071 (95%-conf.int. 0.09684 - 0.10476)
1 ROUGE-L Average_F: 0.15023 (95%-conf.int. 0.14579 - 0.15517)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.12100 (95%-conf.int. 0.11330 - 0.12945)
1 ROUGE-S4 Average_P: 0.02401 (95%-conf.int. 0.02202 - 0.02620)
1 ROUGE-S4 Average_F: 0.03660 (95%-conf.int. 0.03384 - 0.03949)
```

Figure 4.30: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.30: Referring Figure 4.30

Table	Average_R
ROUGE-1	48.9 ± 2.5
ROUGE-L	45.2 ± 2.4

4.3.2 Dailymail dataset

Refer subsection 3.1.2 to know details about the Dailymail dataset. Refer to the three cases described in subsection 3.1.3

4.3.2.1 Case 1

Model Input Parameters:

- Input Feature Vector Size = 256
- batch_size = 100
- checkpoint_every = 2000
- dropout_keep_prob = 0.5
- evaluate_every = 100
- num_checkpoints = 5
- num_epochs = 100
- num_layers = 2
- truncated_backprop_length = 40

Some other parameters:

- learning rate $\eta = 0.3$
- optimizer = AdagradOptimizer
- Number of training samples = 5066775
- state size = input feature vector size

Evaluation on Test Data

- Number of test samples = 264000
- **Accuracy** = 0.734280 =73.4280%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.31

2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.32

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.60764 (95%-conf.int. 0.60335 - 0.61198)
1 ROUGE-1 Average_P: 0.77394 (95%-conf.int. 0.77057 - 0.77725)
1 ROUGE-1 Average_F: 0.65160 (95%-conf.int. 0.64819 - 0.65487)
---------------------------------------------
1 ROUGE-2 Average_R: 0.53765 (95%-conf.int. 0.53320 - 0.54191)
1 ROUGE-2 Average_P: 0.68426 (95%-conf.int. 0.68026 - 0.68840)
1 ROUGE-2 Average_F: 0.57523 (95%-conf.int. 0.57152 - 0.57883)
---------------------------------------------
1 ROUGE-3 Average_R: 0.51788 (95%-conf.int. 0.51343 - 0.52219)
1 ROUGE-3 Average_P: 0.65987 (95%-conf.int. 0.65585 - 0.66415)
1 ROUGE-3 Average_F: 0.55375 (95%-conf.int. 0.55001 - 0.55743)
---------------------------------------------
1 ROUGE-L Average_R: 0.60194 (95%-conf.int. 0.59763 - 0.60628)
1 ROUGE-L Average_P: 0.76598 (95%-conf.int. 0.76254 - 0.76936)
1 ROUGE-L Average_F: 0.64525 (95%-conf.int. 0.64180 - 0.64853)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.51586 (95%-conf.int. 0.51144 - 0.52008)
1 ROUGE-S4 Average_P: 0.65910 (95%-conf.int. 0.65520 - 0.66326)
1 ROUGE-S4 Average_F: 0.55167 (95%-conf.int. 0.54801 - 0.55522)
```

Figure 4.31: Reference: Original Extractive and Predicted: Predicted Extractive

```
1 ROUGE-1 Average_R: 0.41759 (95%-conf.int. 0.41406 - 0.42092)
1 ROUGE-1 Average_P: 0.16821 (95%-conf.int. 0.16628 - 0.17010)
1 ROUGE-1 Average_F: 0.21491 (95%-conf.int. 0.21308 - 0.21669)
---------------------------------------------
1 ROUGE-2 Average_R: 0.10159 (95%-conf.int. 0.09956 - 0.10350)
1 ROUGE-2 Average_P: 0.04163 (95%-conf.int. 0.04056 - 0.04274)
1 ROUGE-2 Average_F: 0.05253 (95%-conf.int. 0.05145 - 0.05365)
---------------------------------------------
1 ROUGE-3 Average_R: 0.04415 (95%-conf.int. 0.04260 - 0.04561)
1 ROUGE-3 Average_P: 0.01925 (95%-conf.int. 0.01840 - 0.02011)
1 ROUGE-3 Average_F: 0.02369 (95%-conf.int. 0.02284 - 0.02459)
---------------------------------------------
1 ROUGE-L Average_R: 0.38625 (95%-conf.int. 0.38303 - 0.38936)
1 ROUGE-L Average_P: 0.15431 (95%-conf.int. 0.15252 - 0.15610)
1 ROUGE-L Average_F: 0.19773 (95%-conf.int. 0.19607 - 0.19935)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.08846 (95%-conf.int. 0.08670 - 0.09009)
1 ROUGE-S4 Average_P: 0.03501 (95%-conf.int. 0.03405 - 0.03595)
1 ROUGE-S4 Average_F: 0.04442 (95%-conf.int. 0.04350 - 0.04540)
```

Figure 4.32: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.31: Referring Figure 4.31

Table	Average_R
ROUGE-1	60.8 ± 0.9
ROUGE-L	60.2 ± 0.9

Table 4.32: Referring Figure 4.32

Table	Average_R
ROUGE-1	41.8 ± 0.7
ROUGE-L	38.6 ± 0.6

4.3.2.2 Case 2

Model Input Parameters:

- Input Feature Vector Size = 256
- batch_size = 100
- checkpoint_every = 2000
- dropout_keep_prob = 0.5
- evaluate_every = 100
- num_checkpoints = 5
- num_epochs = 100
- num_layers = 2
- truncated_backprop_length = 40

Some other parameters:

- learning rate $\eta = 0.3$
- optimizer = AdagradOptimizer

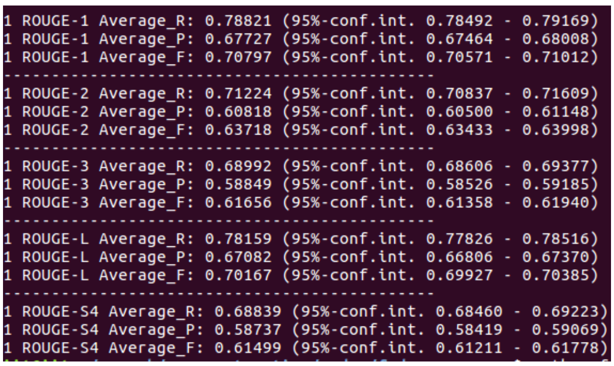
- Number of training samples = 5066775
- state size = input feature vector size

Evaluation on Test Data

- Number of test samples = 264000
- **Accuracy** = 0.6243 =62.43%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.33
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.34



```
1 ROUGE-1 Average_R: 0.78821 (95%-conf.int. 0.78492 - 0.79169)
1 ROUGE-1 Average_P: 0.67727 (95%-conf.int. 0.67464 - 0.68008)
1 ROUGE-1 Average_F: 0.70797 (95%-conf.int. 0.70571 - 0.71012)
---------------------------------------------
1 ROUGE-2 Average_R: 0.71224 (95%-conf.int. 0.70837 - 0.71609)
1 ROUGE-2 Average_P: 0.60818 (95%-conf.int. 0.60500 - 0.61148)
1 ROUGE-2 Average_F: 0.63718 (95%-conf.int. 0.63433 - 0.63998)
---------------------------------------------
1 ROUGE-3 Average_R: 0.68992 (95%-conf.int. 0.68606 - 0.69377)
1 ROUGE-3 Average_P: 0.58849 (95%-conf.int. 0.58526 - 0.59185)
1 ROUGE-3 Average_F: 0.61656 (95%-conf.int. 0.61358 - 0.61940)
---------------------------------------------
1 ROUGE-L Average_R: 0.78159 (95%-conf.int. 0.77826 - 0.78516)
1 ROUGE-L Average_P: 0.67082 (95%-conf.int. 0.66806 - 0.67370)
1 ROUGE-L Average_F: 0.70167 (95%-conf.int. 0.69927 - 0.70385)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.68839 (95%-conf.int. 0.68460 - 0.69223)
1 ROUGE-S4 Average_P: 0.58737 (95%-conf.int. 0.58419 - 0.59069)
1 ROUGE-S4 Average_F: 0.61499 (95%-conf.int. 0.61211 - 0.61778)
```

Figure 4.33: Reference: Original Extractive and Predicted: Predicted Extractive

```
--------------------------------------------
1 ROUGE-1 Average_R: 0.54854 (95%-conf.int. 0.54505 - 0.55202)
1 ROUGE-1 Average_P: 0.14855 (95%-conf.int. 0.14695 - 0.15017)
1 ROUGE-1 Average_F: 0.21535 (95%-conf.int. 0.21363 - 0.21711)
--------------------------------------------
1 ROUGE-2 Average_R: 0.16830 (95%-conf.int. 0.16573 - 0.17097)
1 ROUGE-2 Average_P: 0.04547 (95%-conf.int. 0.04447 - 0.04641)
1 ROUGE-2 Average_F: 0.06572 (95%-conf.int. 0.06459 - 0.06690)
--------------------------------------------
1 ROUGE-3 Average_R: 0.07800 (95%-conf.int. 0.07603 - 0.08012)
1 ROUGE-3 Average_P: 0.02193 (95%-conf.int. 0.02117 - 0.02269)
1 ROUGE-3 Average_F: 0.03133 (95%-conf.int. 0.03045 - 0.03229)
--------------------------------------------
1 ROUGE-L Average_R: 0.51137 (95%-conf.int. 0.50809 - 0.51477)
1 ROUGE-L Average_P: 0.13739 (95%-conf.int. 0.13589 - 0.13888)
1 ROUGE-L Average_F: 0.19964 (95%-conf.int. 0.19800 - 0.20124)
--------------------------------------------
1 ROUGE-S4 Average_R: 0.14401 (95%-conf.int. 0.14181 - 0.14622)
1 ROUGE-S4 Average_P: 0.03728 (95%-conf.int. 0.03644 - 0.03809)
1 ROUGE-S4 Average_F: 0.05434 (95%-conf.int. 0.05333 - 0.05535)
```

Figure 4.34: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.33: Referring Figure 4.33

Table	Average_R
ROUGE-1	78.8 ± 0.7
ROUGE-L	78.1 ± 0.7

Table 4.34: Referring Figure 4.34

Table	Average_R
ROUGE-1	54.9 ± 0.7
ROUGE-L	51.1 ± 0.7

4.3.2.3 Case 3

Model Input Parameters:

- Input Feature Vector Size = 256
- batch_size = 100
- checkpoint_every = 2000
- dropout_keep_prob = 0.5
- evaluate_every = 100
- num_checkpoints = 5

- num_epochs = 100
- num_layers = 2
- truncated_backprop_length = 50

Some other parameters:

- learning rate $\eta = 0.3$
- optimizer = AdagradOptimizer
- Number of training samples = 402309
- state size = input feature vector size

Evaluation on Test Data

- Number of test samples = 208000
- **Accuracy** = 0.7806714 =78.06714%

Evaluating ROUGE Scores

1. Reference Summary folder path: Original Extractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.35
2. Reference Summary folder path: Original Abstractive Summary files and Predicted Summary folder path: Predicted Extractive Summary files. Refer Figure 4.36

Table 4.35: Referring Figure 4.35

Table	Average_R
ROUGE-1	75.6 ± 0.8
ROUGE-L	75.2 ± 0.8

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.75606 (95%-conf.int. 0.75201 - 0.76004)
1 ROUGE-1 Average_P: 0.81941 (95%-conf.int. 0.81647 - 0.82220)
1 ROUGE-1 Average_F: 0.76553 (95%-conf.int. 0.76242 - 0.76844)
---------------------------------------------
1 ROUGE-2 Average_R: 0.71019 (95%-conf.int. 0.70579 - 0.71453)
1 ROUGE-2 Average_P: 0.76878 (95%-conf.int. 0.76528 - 0.77221)
1 ROUGE-2 Average_F: 0.71821 (95%-conf.int. 0.71447 - 0.72151)
---------------------------------------------
1 ROUGE-3 Average_R: 0.69584 (95%-conf.int. 0.69128 - 0.70022)
1 ROUGE-3 Average_P: 0.75353 (95%-conf.int. 0.75000 - 0.75703)
1 ROUGE-3 Average_F: 0.70346 (95%-conf.int. 0.69958 - 0.70685)
---------------------------------------------
1 ROUGE-L Average_R: 0.75246 (95%-conf.int. 0.74840 - 0.75648)
1 ROUGE-L Average_P: 0.81514 (95%-conf.int. 0.81216 - 0.81799)
1 ROUGE-L Average_F: 0.76176 (95%-conf.int. 0.75845 - 0.76469)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.69272 (95%-conf.int. 0.68820 - 0.69702)
1 ROUGE-S4 Average_P: 0.75082 (95%-conf.int. 0.74731 - 0.75432)
1 ROUGE-S4 Average_F: 0.70024 (95%-conf.int. 0.69639 - 0.70352)
```

Figure 4.35: Reference: Original Extractive and Predicted: Predicted Extractive

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.47811 (95%-conf.int. 0.47458 - 0.48148)
1 ROUGE-1 Average_P: 0.15986 (95%-conf.int. 0.15811 - 0.16173)
1 ROUGE-1 Average_F: 0.21992 (95%-conf.int. 0.21817 - 0.22167)
---------------------------------------------
1 ROUGE-2 Average_R: 0.13682 (95%-conf.int. 0.13441 - 0.13921)
1 ROUGE-2 Average_P: 0.04464 (95%-conf.int. 0.04366 - 0.04572)
1 ROUGE-2 Average_F: 0.06172 (95%-conf.int. 0.06057 - 0.06290)
---------------------------------------------
1 ROUGE-3 Average_R: 0.06496 (95%-conf.int. 0.06304 - 0.06692)
1 ROUGE-3 Average_P: 0.02177 (95%-conf.int. 0.02101 - 0.02262)
1 ROUGE-3 Average_F: 0.02982 (95%-conf.int. 0.02887 - 0.03085)
---------------------------------------------
1 ROUGE-L Average_R: 0.44356 (95%-conf.int. 0.44018 - 0.44692)
1 ROUGE-L Average_P: 0.14720 (95%-conf.int. 0.14562 - 0.14892)
1 ROUGE-L Average_F: 0.20300 (95%-conf.int. 0.20136 - 0.20469)
---------------------------------------------
1 ROUGE-S4 Average_R: 0.11694 (95%-conf.int. 0.11479 - 0.11894)
1 ROUGE-S4 Average_P: 0.03693 (95%-conf.int. 0.03608 - 0.03782)
1 ROUGE-S4 Average_F: 0.05136 (95%-conf.int. 0.05036 - 0.05239)
```

Figure 4.36: Reference: Original Abstractive and Predicted: Predicted Extractive

Table 4.36: Referring Figure 4.36

Table	Average_R
ROUGE-1	47.8 ± 0.7
ROUGE-L	44.4 ± 0.7

Chapter 5

Conclusion and Future work

In this report, we covered the basic concepts of text summarization, neural networks and ROUGE [4] measure. We saw some of the recent works in the field of extractive text summarization. Three models for extractive text summarization were proposed in this report. The first model was Convolutional Neural Network based model. The second model was Linear Support Vector Machine based model. The third model was Deep Bidirectional Recurrent Neural Network based model. We used the CNN and Dailymail datasets for training and evaluating the proposed models.

As one can see from the results, we got really good values for ROUGE-1 and ROUGE-L in all cases. We have seen that all the three proposed model perform more or less the same. We have got the highest ROUGE scores for the **CASE 2** for both the datasets and for all the proposed model. We see that ROUGE scores are higher when references are original extractive summaries rather than original abstract summaries.

We plan to work on a sequence to sequence attention model for extractive text summarization similar to the one proposed in [15] but it was proposed for abstractive text summarization. The main challenge is each article in the CNN and Dailymail datasets, on an average has 800 words which is very very large when compared to what is typical fed as input to a sequence to sequence attention model with input sequence being 120-150 words. We can improve the performance of our Bidirectional RNN model by giving it better quality fixed length representations for sentences, i.e., if we are able to get feature vectors which represent sentences in a better way. We need to model sentence embeddings for sentences which will be similar to having word embeddings for words and then feed those sentence embeddings to the Bidirectional RNN.

References

[1] N.Moratanch ,S.Chitrakala , "A Survey on Extractive Text Summarization"

[2] Ramesh Nallapati, Feifei Zhai, Bowen Zhou, “SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents”

[3] L. SUanmali, M. S. Binwahlan, and N. Salim, "Sentence features fusion for text summarization using fuzzy logic," in Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on, vol. 1. IEEE, 2009, pp. 142-146.

[4] Chin-Yew Lin, "ROUGE : A Package for Automatic Evaluation of Summaries"

[5] Yoon Kim, "Convolutional Neural Networks for Sentence Classification"

[6] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, Bing Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond"

[7] Y. Sankarasubramaniam, K. Ramanathan, and S. Ghosh, "Text summarization using wikipedia," Information Processing & Management, vol. 50, no. 3, pp. 443-461, 2014.

[8] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," Journal of Artificial Intelligence Research, pp. 457-479, 2004.

[9] F. Kyoomarsi, H. Khosravi, E. Eslami, P. K. Dehkordy, and A. Tajoddin, "Optimizing text summarization based on fuzzy logic," in Seventh IEEElACIS International Conference on Computer and Information Science. IEEE, 2008, pp. 347-352.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space"

[11] I. Mashechkin, M. Petrovskiy, D. Popov, and D. V. Tsarev, "Automatic text summarization using latent semantic analysis," Programming and Computer Software, vol. 37, no. 6, pp. 299-305, 2011.

[12] J. L. Neto, A. A. Freitas, and C. A. Kaestner, "Automatic text summarization using a machine learning approach," in Advances in Artificial Intelligence. Springer, 2002, pp. 205-215.

[13] M. G. Ozsoy, F. N. Alpaslan, and 1. Cicekli, "Text summarization using latent semantic analysis," Journal of Information Science, vol. 37, no. 4,pp. 405-417, 2011

[14] S. M. R .. W. T. L., Brin, "The pagerank citation ranking: Bringing order to the web," Technical report, Stanford University, Stanford, CA., Tech. Rep., (1998).

[15] Xin Pan, Peter Liu, "Sequence-to-Sequence with Attention Model for Text Summarization."