EXPERIMENTS IN TEXT SUMMARIZATION USING DEEP LEARNING

by

Sai Amrit Bulusu

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

Charlotte

2018

Approved by:

Dr. Wlodek Zadrozny

Dr. Srinivas Akella

Dr. Minwoo Lee

ABSTRACT

SAI AMRIT BULUSU. Experiments in Text Summarization using Deep Learning.
(Under the direction of DR. WLODEK ZADROZNY)

Deep Learning has been the go-to tool for text summarization in the recent times. Traditional deep learning research focuses on performing abstractive text summarization without considering the user's interests to personalize the summaries.

This problem motivated us to develop a deep learning based text summarization system which can curate personalized summaries. In this work we propose an LSTM based Bi-Directional Recurrent Neural Network model to perform extractive text summarization. Our new deep learning approach focuses on personalizing the extractive summaries based on user's interests to make the summaries more intriguing to the user. We performed the experiments on CNN and Daily Mail news dataset. We also have experimented with a new set of semantic word vectors called Conceptnet Numberbatch. Out of domain evaluation was done on the Signal-Media one million news articles dataset. Experimental results on the two summarization datasets demonstrate that our models obtain results comparable to the state of the art. The personalization framework curates interesting summaries based on user's interests while retaining the important information from the source document.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor, Dr. Wlodek Zadrozny, for giving me the opportunity, guidance and support to complete my thesis in one of the most interesting research topics. He provided immense expertise at every step. He is one of the most versatile mentors who let me explore different ideas while steering me in the right direction.

I would also like to express my gratitude to my committee members, Dr. Minwoo Lee and Dr. Srinivas Akella for their valuable feedback at every milestone. They played a vital role in structuring my thesis by providing ideas that guided me towards this accomplishment. My sincere appreciation to the University of North Carolina at Charlotte for equipping me with necessary infrastructure and aiding me in the entire process.

This would not have been possible without consistent encouragement from my friends Lakshmi Lolla and Bhargav Nallani, and family. Especially my parents, sister Mahima Bulusu and brother-in-law Vishal Guntur who made this a smooth sailing journey. And of course a gigantic "thank you" to all my fellow researchers and well wishers.

DEDICATION

dedicated to my parents, Vijayalakshmi and Prakasam.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES ix

LIST OF ABBREVIATIONS x

CHAPTER 1: INTRODUCTION 1

1.1. Problem Statement 2

CHAPTER 2: BACKGROUND 3

2.1. Text Summarization 3

2.2. Approaches 4

2.2.1. Statistical and Machine Learning Based Approaches 4

2.2.2. Graph Based Approaches 7

2.3. Deep Learning 11

2.3.1. Recurrent Neural Networks 11

2.3.2. Long-Short Term Memory (LSTM) 14

2.3.3. Gate Recurrent Unit (GRU) 16

2.3.4. Encoder-Decoder Networks 17

2.3.5. Attention Mechanism 18

CHAPTER 3: RELATED WORK 20

3.1. Deep Learning in Text Summarization 20

3.2. Personalized Text Summarization 25

CHAPTER 4: PROPOSED MODEL 26

CHAPTER 5: EXPERIMENTAL SETUP 29

5.1. Corpora 29

5.2. Data Preprocessing 31

5.3. Word Embeddings 32

5.4. Data Preparation 33

5.5. Model Settings 34

5.6. Evaluation Metrics 35

CHAPTER 6: RESULTS 37

6.1. Quantitative Evaluation 38

6.2. Qualitative Evaluation 40

6.2.1. Generic Summaries 40

6.2.2. Personalized Summaries 43

CHAPTER 7: CONCLUSIONS 47

7.1. Future Scope 48

REFERENCES 49

APPENDIX A: EXAMPLES OF PREDICTED SUMMARIES 52

LIST OF FIGURES

FIGURE 2.1: Block diagram of automatic extractive text summarization using statistical techniques. 6

FIGURE 2.2: Degree centrality scores for LexRank. 10

FIGURE 2.3: Sequence to Vector learning RNN unrolled. 12

FIGURE 2.4: RNN and BRNN architectures. 13

FIGURE 2.5: LSTM Architecture. 15

FIGURE 2.6: GRU Architecture. 17

FIGURE 3.1: SummaRuNNer Model Architecture. 21

FIGURE 3.2: A recurrent convolutional document reader with a neural sentence extractor. 22

FIGURE 3.3: The illustration of the hierarchical LSTM encoder-decoder model. 23

FIGURE 3.4: Hierarchical Encoder-Decoder model architectures with statistical features and attention. 24

FIGURE 4.1: The illustration of the proposed BRNN with LSTM cells. 27

FIGURE 5.1: Example of Daily Mail dataset. 31

FIGURE 5.2: Conceptnet Numberbatch. 33

LIST OF TABLES

TABLE 6.1: Performance on exractive summarization on the extractive summaries. 38

TABLE 6.2: Performance comparison on exractive summarization using the gold summaries (F1). 39

TABLE 6.3: Performance of exractive summarization using the gold summaries (Recall). 40

TABLE 6.4: Performance of exractive summarization on the Signal Media Dataset (Recall). 40

LIST OF ABBREVIATIONS

AS Aggregate Similarity.

BP Bushy Path of sentence.

BRNN Bi-Directional Recurrent Neural Network.

CEN Centrality of Sentence.

CNN Convolutional Neural Network.

FFNN Feed Forward Neural Network.

GRU Gated Recurrent Unit.

HITS Hyperlink-Induced Topic Search.

LSTM Long Short Term Memory.

NER Named Entity Recognition.

PN Presence of Numbers in sentence.

PNE Presence of Name Entity in sentence.

PNN Probabilistic Neural Network.

R2T Resemblance of sentence to the Title.

RL Relative Length of sentence.

RNN Recurrent Neural Network.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

TF-IDF Term Frequency - Inverse Document Frequency.

CHAPTER 1: INTRODUCTION

Data is revolutionizing human life. In the last two years we've created more data than in the entire previous history of the human race. By 2020, about 1.7 megabytes of new information will be created every second for every human being on the planet [1]. Every second, we search thousands of queries on Google, upload hundreds of hours of video on YouTube, share thousands of pictures online, interact with many users on social media and upload immense amount of text data on the internet. All this data is extremely valuable, yet at the moment less than 0.5% of all data available is ever analyzed and used [2].

Data can be broadly classified into structured data and unstructured data. Structured data mostly comprised of well tabulated data forms a small part of content that is generated each day. Unstructured data is everything else consisting of videos, email messages, instant messages, text messages, text files, word documents, PDF files, books, letters, written documents, audio and CAT-scans(medical data), web pages, news articles, status updates and blogs. Data volume is set to grow 800% over the next 5 years and 80% of it will reside as unstructured data. Text data forms an important portion of unstructured data as a lot of value and insight can be drawn from it. For example, 269 billion emails are sent daily in 2017, and this is expected to grow by 4.4% yearly to 319.6 billion in 2021. 571 new websites are created every minute of the day, 30 Billion pieces of content shared on Facebook every month according to waterfordtechnologies. This data can be used in applications of Knowledge Management, Cybercrime Prevention, Text Summarization, Customer Care, Contextual Advertising, Content Enrichment, Spam Filtering, Social Media Analysis and these applications are endless. However, there comes a problem with the explosive

growth in unstructured data that has been compared to a popular metaphor of finding the needle in a haystack. Huge amounts of textual information has to be consumed quickly, in a condensed manner while preserving the essence of the data.

Text summarizations in particular are required in headlines, academic notes, previews, synopsis, reviews, abridgements, and chronologies as they save time while delivering the most important parts of a document in a concise manner. Processing and condensing data manually is extremely time consuming and cannot keep up with the rate at which data is growing each day. Automatic text summarization methods can process huge amounts of content quickly, can improve effectiveness of indexing, and create unbiased summaries compared to manual ones. The real challenge here was in making the summaries relevant and occupy less space.

1.1 Problem Statement

The automatic text summarization approaches, provide generic summaries to any user. The summaries are short, easier to read and present the most important information in the document. Personalized text summarization gives more interesting and intriguing summaries to the users by including relevant information.

Deep Learning has been the go-to tool for text summarization in the recent times. The state-of-the-art deep learning models in abstractive and extractive summarization develop generic summaries to the users irrespective of their interests. Very little emphasis has been given on personalizing the summaries while using deep learning as a tool for automatic summarization.

The goal of this research is to propose a new robust deep learning model which performs extractive text summarization on documents and additionally adds a personalization framework based on user's profile data. Extractive summarization techniques are less complex and generate grammatically correct sentences which can help us perform better personalization.

CHAPTER 2: BACKGROUND

2.1 Text Summarization

Automatic text summarization should consist of the most relevant information in a document and at the same time, it should occupy less space than the source document.

> "Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)." – - *Advances in Automatic Text Summarization, 1999.* (page 1)

There are essentially two types of text summarization - Extractive and Abstractive text summarization. Extractive text summarization is the process of extracting important sentences in the document without paraphrasing them and including in the summary. Abstractive text summarization on the other hand provides an abstract summary which includes words and phrases different from the ones occurring in the source document. Abstractive text summarization is comparatively complex and hard to implement as it requires extensive natural language understanding as an abstractive summary consists of ideas or concepts taken from the original document but are re-interpreted and shown in a different form. Over the last decade or so, extractive techniques have performed better than the abstractive approaches and people chose the former over the latter since the former is easier to implement [3].

Some of the key issues that we need to address while developing a text summarizer are redundancy in information, temporal dimension, co-reference resolution and sentence ordering. There are different types of text summarization, two important types of text summarization are based on number of documents, single and multi-document

summarization. Multi-document summarization is comparatively hard since redundancy is a huge problem. Multi-document summarization is fundamentally an extension of single document summaries. Query based summaries are the summaries which are tailored based on the user's query and generic summaries consist of the most generic and important information in the document. Using machine learning we can develop summaries either by using supervised or unsupervised methods. Indicative summaries are those which just give an overview of the document, whereas informative summaries give the whole information in an elaborated form.

Personalized text summaries are those which contain specific information that the user desires and its based on either consumer requirements or are generated based on their interests. In this research, we would like to work on extractive summarization for single documents by using machine learning/deep learning-based approaches. We would then extend the work by making the summaries personalized and incorporate deep learning to make it more robust.

2.2 Approaches

Extensive research has been done in the recent past in the field of text summarization and as a result many novel methods have been developed to perform extractive text summarization. Some of the most successful methods are divided into the following types of approaches, statistical based approaches, topic based approaches, graph based approaches, discourse based approaches and approaches based on machine learning and deep learning. This section describes in detail some of the most successful approaches in the respective areas. Machine learning approaches learn from the data and the task can be supervised or unsupervised or semi-supervised.

2.2.1 Statistical and Machine Learning Based Approaches

Statistical based approaches work on statistical features that are generated from the source document and use these features to summarize text in any language. This

approach can reduce the processor and memory capacity required to perform automatic text summarization. Some of the useful features according to the paper "A survey of text summarization extractive techniques" [4] are:

- Content word feature
- Title word feature
- Sentence Location feature
- Proper noun feature
- Upper-case word feature
- Cue-phrase feature
- Biased word feature
- Font based feature
- Pronouns
- Sentence-to-sentence cohesion
- Sentence-to-centroid cohesion
- Occurrence of non-essential information
- Discourse analysis
- Positive keyword (based on frequency count)
- Negative keyword (based on frequency count)
- Centrality of sentence
- Presence of numerical data or not

- Presence of proper noun in the sentence
- Node's sentence bushy path
- Summation of similarities for each node

Some other features that can discover important words are TF*IDF, information gain, mutual information and residual inverse document frequency. Features give weights to words and based on these weights scores are assigned to sentences and highly scored sentences are selected and included in the summary. Following is a basic taxonomy for using the statistical techniques for extractive text summarization.

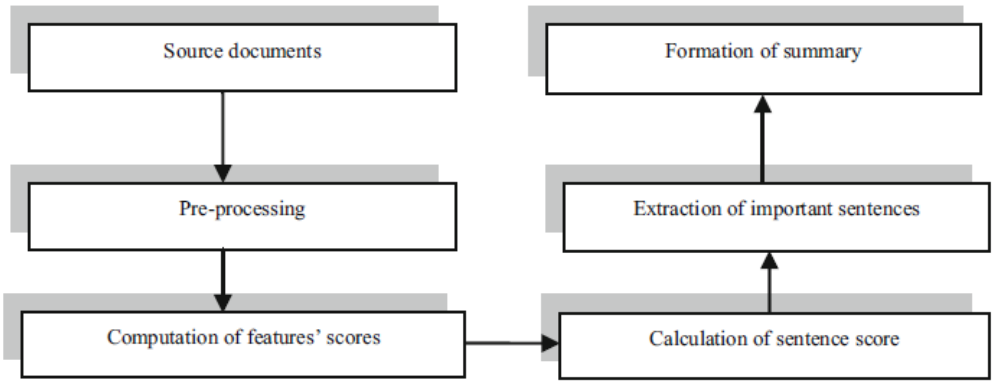


Figure 2.1: Block diagram of automatic extractive text summarization system by using statistical techniques. [4]

Fattah and Ren [5] proposed a method to improve selection of content in automatic summarization of text with the help of a few statistical features. Their model being a trainable summarizer develops statistical features which are used to generate summaries. These features are: Position of Sentence (Pos), +ve keyword, -ve keyword, Resemblance of sentence to the Title (R2T), Centrality of Sentence (Cen), Presence of Name Entity in sentence (PNE), Presence of Numbers in sentence (PN), Bushy Path of sentence (BP), Relative Length of sentence (RL), and Aggregate Similarity (AS). Genetic Algorithm and Regression models were used to generate weights for

these feature vectors. Neural networks (FFNN and PNN) are used for classification of sentences. Results show that the feature BP is the most important feature.

In another paper, Fattah et al. [6] proposed a multi-document summarization approach for enhancing content selection in text by making use of statistical features. They have used many new features for the hybrid machine learning model with Naive Bayes, Maximum Entropy and Support Vector Machine. Obtained results show that features like similarity of words, format of text, cue phrases and presence of unimportant information have showed good results.

Neural networks [7] have also been used to make the network learn the types of sentence that should be included in the summary. This is accomplished by training the network with sentences in several test paragraphs where each sentence is identified as to whether it should be included in the summary or not. It uses a three layered, feed forward network which has been proven to be a universal function approximator. After the network has learned the features, feature fusion is performed which eliminates the uncommon features and collapses the effects of common features [8]. The hidden layer activation values for each hidden layer activation values for each hidden layer neuron are clustered. Each cluster is identified by its centroid and frequency. The activation value of each hidden layer neuron is replaced by the centroid of the cluster, which the activation value belongs to.

2.2.2 Graph Based Approaches

Graph based approaches have shown promise in performing extractive text summarization. Fundamentally in graph-based approaches the sentences are treated as nodes and edges connect the related text elements together. Graph theoretic representation of passages provides a method of identification of these themes. After the common preprocessing steps, namely, stop word removal and stemming, sentences in the documents are represented as nodes in an undirected graph. Following are some of the most used graph-based approaches for extractive text summarization.

2.2.2.1 TextRank

The basic idea of implementing a graph-based ranking model is that of "voting" or "recommendation" [9]. TextRank algorithm is derived from Google's PageRank [10], but other graph-based algorithms such as HITS [10] or Positional Function [11] can be easily integrated into TextRank model. In PageRank algorithm the score of each vertex in a directed graph $G = (V, E)$ with V vertices and E edges is given as:

$$S(V_i) = (1 - d) + d * \sum_{j \epsilon I_n(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{2.1}$$

Where $I_n(V_i)$ be the set of vertices that point to it and let $Out(V_i)$ be the set of vertices that vertex V_i points to. d is a damping factor and it can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. In TextRank, unlike PageRank the graphs are built from natural language texts and may include multiple or partial links between the units. They have taken the PageRank algorithm and derived another equation with weighted links (W_{ij}) between the vertices (i and j) where the stronger the connection between two sentences the more the weight it has.

$$WS(V_i) = (1 - d) + d * \sum_{V_j \epsilon I_n(V_i)} \frac{w_{ij}}{\sum_{V_k \epsilon Out(V_j)} w_{jk}} WS(V_j) \tag{2.2}$$

One of the two uses for TextRank is sentence extraction which is used for generating extractive summaries for single documents. For the task of sentence extraction, the goal is to rank entire sentences and therefore a vertex is added to the graph for each sentence in the text. Similarity between sentences are calculated using either cosine, string kernels or any other similarity function. The similarity might be to determine the common tokens between the lexical representations of the two sentences. The resulting graph is highly connected with a weight associated with each edge, indicating the strength between sentences. By running the ranking algorithm, the top ranked

sentences are included in the summary.

2.2.2.2 LexRank

TextRank is used for single document summarization and does not perform well on multi-document extractive summaries. An updated approach LexRank [12] is used for computing sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. TextRank deals with assessing the centrality of each sentence in a cluster and extract the most important ones to include in the summary. The clusters are represented in undirected acyclic graphs where the vertices represent sentences and edges are defined in terms of the similarity between pairs of sentences.

Centrality of sentences depends upon the summation of centrality of words that are present in the sentence. The centroid of a cluster is a pseudo-document which consists of words that have tf-idf scores above a predefined threshold, where tf is the frequency of a word in the cluster, and idf values are typically computed over a much larger and similar genre data set. It works on something called centrality-based sentence salience. This approach is based on a concept called prestige in social networks. A cluster of documents can be viewed as a network of sentences that are related to each other.

To define centrality, Radev et al. [12] have defined a bag of words model to represent each sentence as an N-dimensional vector. The similarity between two words is calculated by using a modified version of cosine where $tf_{w,s}$ is the number of occurrences of the word in the sentence.

Cluster of documents may be represented by a cosine similarity matrix as shown below. D_xS_y represents the y^{th} sentence in x^{th} document. This matrix can also be represented as a weighted graph where each edge shows the cosine similarity between a pair of sentence. Since a lot of sentences can be similar, they chose only the ones which are significantly similar by defining a threshold so that the cluster can be viewed as

an undirected graph, where each sentence is a node and significantly similar sentences are connected to each other. They defined the degree of centrality as the degree of the corresponding node in the similarity graph. As seen in Figure 2.2, the choice of cosine threshold dramatically influences the interpretation of centrality. Too low thresholds may mistakenly take weak similarities into consideration while too high thresholds may lose many of the similarity relations in a cluster.

ID	Degree (0.1)	Degree (0.2)	Degree (0.3)
d1s1	5	4	2
d2s1	7	4	2
d2s2	2	1	1
d2s3	6	3	1
d3s1	5	2	1
d3s2	7	5	1
d3s3	2	2	1
d4s1	9	6	1
d5s1	5	4	2
d5s2	6	4	1
d5s3	5	2	2

Figure 2.2: Degree centrality scores for graphs using LexRank. [12]

In this approach each edge is treated as a vote to determine the overall centrality value of each node. Degree centrality may have a negative effect in the quality of the summaries in some cases where several unwanted sentences vote for each other and raise their centrality. This can be avoided by weighting each vote. A straightforward way of formulating this idea is to consider every node having a centrality value and distributing this centrality to its neighbors.

Radev et al. computed a modified version of Google's PageRank to solve the problem of random walker to escape from periodic or disconnected components, which make the graph irreducible and aperiodic. A markov chain's irreducible if any state is reachable from any other state and is aperiodic if for all the states, the greatest common divisor is equal to 1. By assigning a uniform probability for jumping to any node in the graph we get the modified version of PageRank.

Unlike the original PageRank method, the similarity graph for sentences is undi-

rected since cosine similarity is a symmetric relation. However, this does not make any difference in the computation of the stationary distribution. This algorithm is called LexRank or Lexical PageRank. The following is an equation for continuous LexRank:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \epsilon adj[u]} \frac{idf - modified - cosine(u,v)}{\sum_{z \epsilon adj[v]} idf - modified - cosine(z,v)} p(v) \tag{2.3}$$

$$idf - modified - cosing(x,y) = \frac{\sum_{w \epsilon x,y} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{x_i \epsilon x} (tf_{x_i,x} idf_{x_i})^2} * \sqrt{\sum_{y_i \epsilon y} (tf_{y_i,y} idf_{y_i})^2}} \tag{2.4}$$

2.3 Deep Learning

Deep Learning is a part of the machine learning family of algorithms which works on the core concept of replicating neural network architectures found in the human brain. Among different classes of neural networks, Recurrent Neural Networks have been the go to tool for text summarization tasks. The core concepts of Recurrent Neural Networks are presented in this section.

2.3.1 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network architecture which has connections pointing backward unlike the usual feed-forward neural networks. RNNs are a class of neural nets that can predict the future based on the present and past data. They are equipped to analyze various sequential data types such as time series data, text data, audio data, stock data, etc. RNNs are extremely useful for natural language processing systems such as text translation, speech-to-text, sentiment analysis or text summarization. Simple RNNs are a network of neurons where each neuron at a time-step t, receives the input vectors as well as its own output

from the previous timestep $t-1$. RNNs can take sequential inputs and produce sequential outputs, these are called sequence-to-sequence models. Alternatively, you could feed the network a sequence of inputs, and ignore all outputs except for the last one, which is called a sequence-to-vector model. Conversely, you could feed the network a single input and produce a sequential output, which is called as a vector-to-sequence model. Lastly, you could have a model that produces a vector from a sequence and then uses that vector to produce sequences as outputs. This hybrid network is called as an encoder-decoder framework, where the encoder is a sequence-to-vector model followed by the decoder which is a vector-to-sequence model. We used sequence-to-vector model for extractive text summarization, where the sequential inputs are sentences of documents and output vectors are labelled 0 or 1, thus making it a binary classification task.

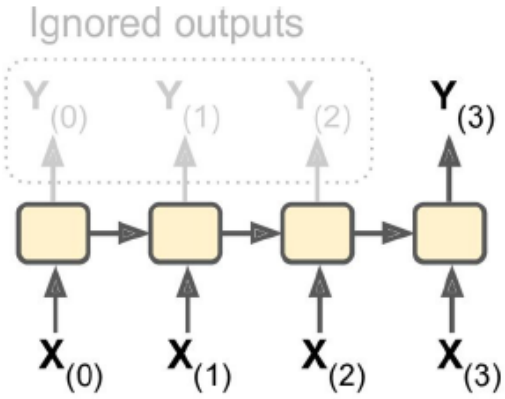


Figure 2.3: Sequence to Vector learning RNN unrolled. [13]

The output of each neuron for a timestep t in an RNN is a state y_t, so the input to each neuron is the previous output state $y_{(t-1)}$ and the standard input x_t. The operation performed in each neuron is:

$$y_{(t)} = \phi({x_{(t)}}^T.w_x + {y_{(t-1)}}^T.w_y + b) \tag{2.5}$$

Here, w_x is the weight assigned to input and w_y is the weight assigned to previous

output.

For extractive text summarization, we have worked with a variant of RNNs called the bi-directional RNN (BRNN). The concept of BRNNs is that to split the neurons in regular RNNs into two opposite directions. BRNNs do not require the input data to be fixed and their future input information can be reached from the current state. BRNNs connect two hidden layers of opposite directions to the same output. By this structure, the output layer can get information from the past and the future states. In BRNNs the neurons are split into two directions, one for forward states, and another for backward states.

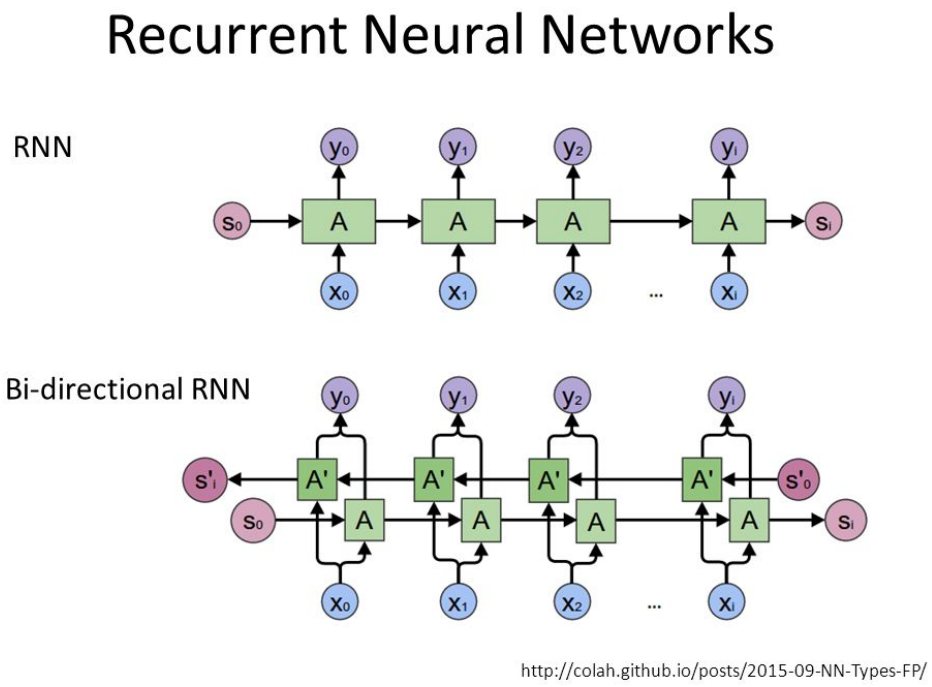


Figure 2.4: RNN and BRNN architectures. Credits: Colah's blog

Since the RNNs consider the outputs from previous time steps, we can say that the RNN has some sort of memory. Hence, each cell in RNN is called as a memory cell. There are multiple variants of the memory cell. One of the main disadvantages of using basic RNN cells is the problem of vanishing/exploding gradients. During backpropagation of deep RNNs, the gradients often get smaller and smaller as the

algorithm progresses down to the lower layers. As a result, the Gradient Descent update leaves the lower layer connection weights virtually unchanged. This problem is called vanishing gradients problem. Conversely, the gradients can get bigger and bigger, this problem is called exploding gradients problem. To overcome this problem, you can unroll the RNN only over a limited number of time steps during training. This is called truncated backpropagation through time. But the problem, of course, is that the model will not be able to learn long-term patterns. During long training of RNNs, the memory of the first inputs gradually starts fading away. After a while, the RNN's state contains no trace of the first inputs. To solve this problem, various types of cells with long-term memory have been introduced.

2.3.2 Long-Short Term Memory (LSTM)

One such cells is called a Long-Short-Term Memory (LSTM) cell. The LSTM cell was proposed in 1997 by Sepp Hochreiter and Jurgen Schmidhuber [14]. LSTMs are designed to avoid the long-term dependency problem. The main difference with LSTM cells is that they have two state vectors instead of one and they are kept separate. The architecture of the LSTM cell is shown in the figure below. The state of the LSTM cells is split into two vectors h and c where we can think of h as a short-term state and c (c stands for "cell") as the long-term state. The main idea of having an LSTM cell is to make the model determine what to remember and what to forget.

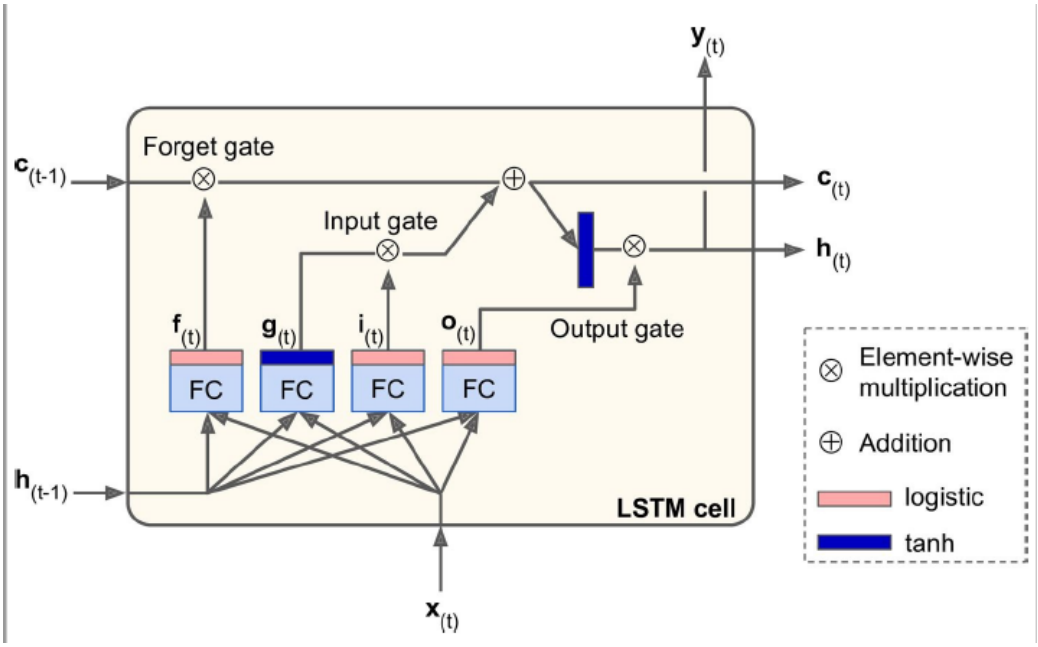


Figure 2.5: LSTM Architecture [13].

For a time-step t, the state cell state of previous time-step $c_{(t-1)}$, traverses the network from left to right and first goes into the forget gate, drops some memories, and then it adds some new memories via the addition operation. The resulting $c_{(t)}$ after the addition operation, is passed through the tanh function, and the result is filtered by the output gate. The main layer in the LSTM outputs $g_{(t)}$, has the role of analyzing the inputs $x_{(t)}$ and the previous short-term state $h_{(t-1)}$. The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is done by the forget gate (controlled by $f_{(t)}$). The next step is to decide what new information we're going to store in the cell state. The input gate (controlled by $i_{(t)}$) controls which parts of $g_{(t)}$ should be added to the long-term state. And finally, the output gate (controlled by $o_{(t)}$) controls which parts of the long-term state should be reads and produces as output $_{y(t)}$. LSTM cell has the capability of recognizing an important input, storing it in long-term state and preserving it for as long as it's needed, and learning to extract it whenever it is needed.

$$i_{(t)} = \sigma({W_{xi}}^T.x_{(t)} + {W_{hi}}^T.h_{(t-1)} + b_i) \tag{2.6}$$

$$f_{(t)} = \sigma({W_{xf}}^T.x_{(t)} + {W_{hf}}^T.h_{(t-1)} + b_f) \tag{2.7}$$

$$o_{(t)} = \sigma({W_{xo}}^T.x_{(t)} + {W_{ho}}^T.h_{(t-1)} + b_o) \tag{2.8}$$

$$g_{(t)} = tanh({W_{xg}}^T.x_{(t)} + {W_{hg}}^T.h_{(t-1)} + b_g) \tag{2.9}$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \tag{2.10}$$

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes tanh(c_{(t)}) \tag{2.11}$$

W_{xi}, W_{xf}, W_{xo}, W_{xg} are the weight matrices of each of the four layers for their conneciton to the input vector $x_{(t)}$. And W_{hi}, W_{hf}, W_{ho}, W_{hg} are the weight matrices of each of four layers for their connection to the previous short-term state $h_{(t-1)}$. And all the biases are present for respective layers.

2.3.3 Gate Recurrent Unit (GRU)

We have also run experiments on another variant of the LSTM cell called Gated Recurrent Unit (GRU). IT was first proposed by Kyunghyun Cho et al. [15], that also introduced the Encoder-Decoder network. GRU is a simplified version of LSTM, where the long-term state and short-term state are merged into a single vector. They lack an output gate and the full state vector is output at every time step. However, it consists of a gate controller which controls both the forget gate and the input gate. The gate is operated on binary outcomes. If the gate controller outputs 1, the input gate is open and the forget gate is closed and vice versa.

$$z_{(t)} = \sigma({W_{xz}}^T.x_{(t)} + {W_{hz}}^T.h_{(t-1)}) \tag{2.12}$$

$$r_{(t)} = \sigma({W_{xr}}^T.x_{(t)} + {W_{hr}}^T.h_{(t-1)}) \tag{2.13}$$

$$g_{(t)} = tanh({W_{xg}}^T.x_{(t)} + {W_{hg}}^T.(r_{(t)} \otimes h_{(t-1)})) \tag{2.14}$$

$$h_{(t)} = (1 - z_{(t)}) \otimes tanh({W_{xg}}^T.h_{(t-1)} + z_{(t)} \otimes g_{(t)}) \tag{2.15}$$

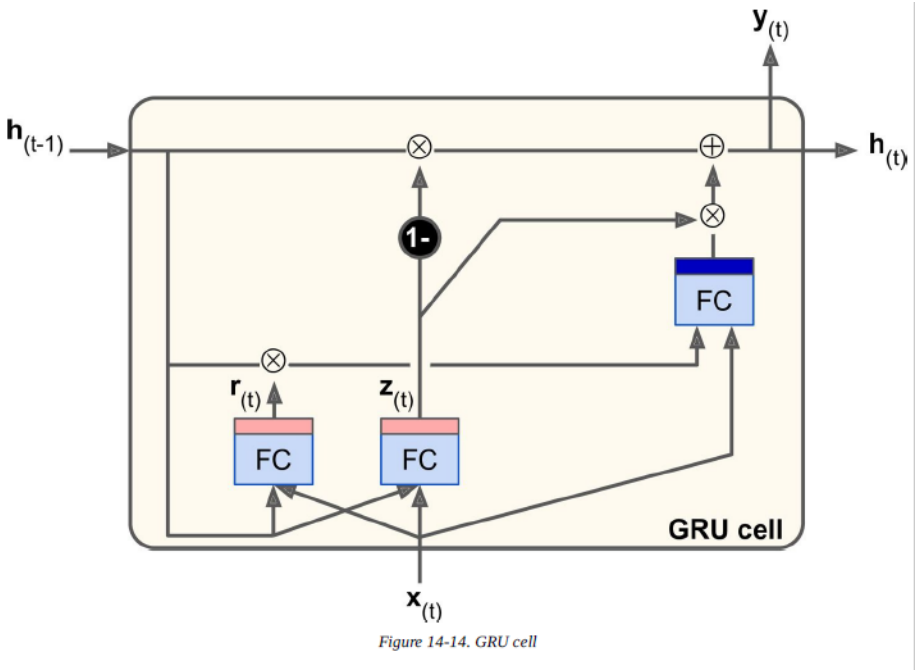


Figure 2.6: GRU Architecture [13].

2.3.4 Encoder-Decoder Networks

A sequence to sequence model is a two-part neural network architecture containing an encoder and a decoder. The encoder is used to encode the input data into a fixed-length vector while a decoder is to use this encoded representation to produce the output. The basic architecture of Encoder-Decoder framework is depicted below.

In an Encoder-Decoder framework, encoder reads the input sequences $x = (x_1, x_2, ..., x_{T_x})$ into a vector c_2. The most common approach is to use an RNN such that:

$$h_{(t)} = f(x_t, h_{(t-1)}) \tag{2.16}$$

and

$$c = q(h_1, ..., h_{T_x}), \tag{2.17}$$

where $h_t \epsilon \Re^n$ is a hidden state at time t, and c is a vector generated from the sequence of the hidden states. f and q are some nonlinear functions. Sutskever et al. [16] used an LSTM as f and q (h_1,...,h_T) = h_T , for instance.

The decoder is often trained to predict the next word yt, given the context vector c and all the previously predicted words y_1,..., $y_{(t-1)}$. In other words, the decoder defines a probability over the translation y by decomposing the joint probability into the ordered conditionals:

$$p(y) = \prod_{t=1}^{T} p(y_t | y_1, ..., y_{(t-1)}, c), \tag{2.18}$$

where y = (y_1,...,y_{t-1}). With an RNN, each conditional probability is modeled as

$$p(y_t | y_1, ..., y_{(t-1)}, c) = g(y_{t-1}, s_t, c), \tag{2.19}$$

where g is a nonlinear, potentially multi-layered, function that outputs the probability of y_t, and s_t is the hidden state of the RNN.

2.3.5 Attention Mechanism

For encoder-decoder neural networks, the use of attentional mechanism allows for the creation of a context-vector at each timestep, given the decoder's current hidden state and a subset of the encoder's hidden states [17]. The context vector c_i depends on a sequence of hidden states to which the encoder maps the input sentence. Each hidden state h_i contains information about the whole input sequence with a strong focus on the parts surrounding the i^{th} word of the input sentence. The context vector is computed as a weighted sum of these hidden states h_i.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.20}$$

The weight a_{ij} of each hidden state h_j is computed by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ij})} \tag{2.21}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.22}$$

Where e_{ij} is an alignment model which scores how well the inputs around position j and the output at position i match. Bahdanau et al. [17] parameterized the alignment model a as a feedforward neural network which is jointly trained with all the other components of the system. C_i is the context vector with the same dimensionality as the hidden states. h_i and c_i are used to compute the next hidden state in the decoder, $h_{(i+1)}$.

$$h_{i+1} = tanh(W[h_t : c_t] + b) \tag{2.23}$$

CHAPTER 3: RELATED WORK

A vast majority of past work in summarization has been extractive, which consists of extracting key phrases or sentences from the document, but most of them were statistical, graph based or natural language based algorithMrs. In applied deep learning, majority of the architectures deal with abstractive summarization, which is paraphrasing the important information present in the document. Although, research has been done in personalized text summarization, our model achieves it's novelty in applying deep learning to perform personalization of summaries. In this chapter, we present the related work in deep learning and personalized text summarization.

3.1 Deep Learning in Text Summarization

One of the most successful papers in extractive summarization is, "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents" [18] . They used a recurrent neural network to perform extractive summarization and proved that it performed better than the state-of-the-art. This work is closely related to ours, but they have created a hierarchical way of representing words, sentences and documents using multi-layered recurrent neural networks. Their main focus is on the sentential extractive summarization of single documents using neural networks as extractive methods are considered to be less complex and expensive and also grammatically and semantically generate correct summaries when compared to abstractive summaries.

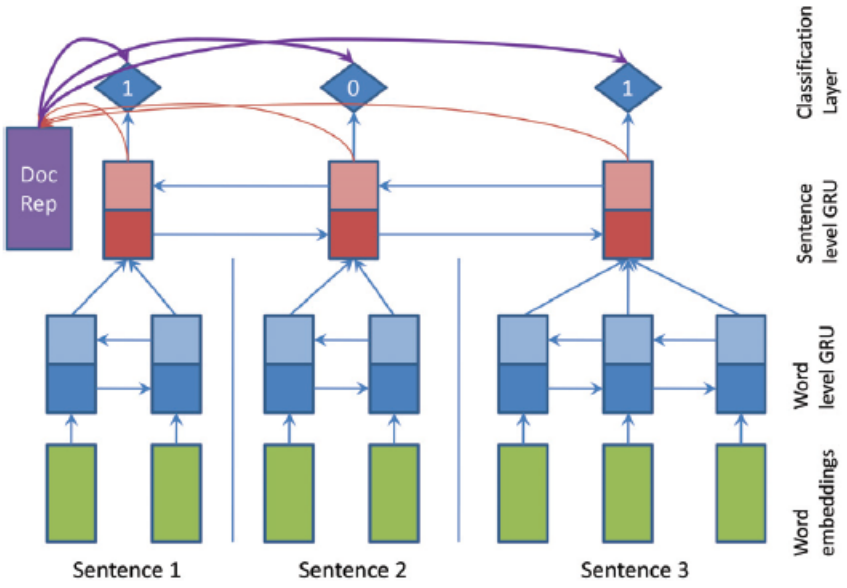


Figure 3.1: SummaRuNNer: A two-layer RNN based sequence classifier: the bottom layer operates at word level within each sentence, while the top layer runs over sentences. [18].

SummaRuNNer model comprises of two layers which use bidirectional GRU RNN, where the first layer runs at the word level and the second layer runs at the sentence level. The abstractive summaries are converted into extractive labels using unsupervised approach. In Abstractive training, the RNN decoder implementation eliminates the use/conversion of extractive labels. The results were calculated on various ROUGE metrics with respect to the gold summaries. SummaRuNNer is considered to be a state-of-art performer and also can be interpretable. Therefore, an interpretable neural sequence model is built which can be used for extractive document summarization. An abstractive approach has also been used to eliminate the extractive labels during training.

Another interesting paper in extractive summarization is, "Neural Summarization by Extracting Sentences and Words" [19]. It focuses on a single document summarization which can extract sentences or words using hierarchical document encoder and an attention-based extractor. They discuss a data driven approach which in-

cludes neural networks and continuous sentence features. Attention is used to select the input words in this approach unlike the intermediary step performed in the previous paper. Transformation and scoring algorithms have been used to match the highlights in a document. The model performs summarization on multiple sentences instead of individual sentences and the decoder selects the desired output from the document we are interested in rather than the entire vocabulary.

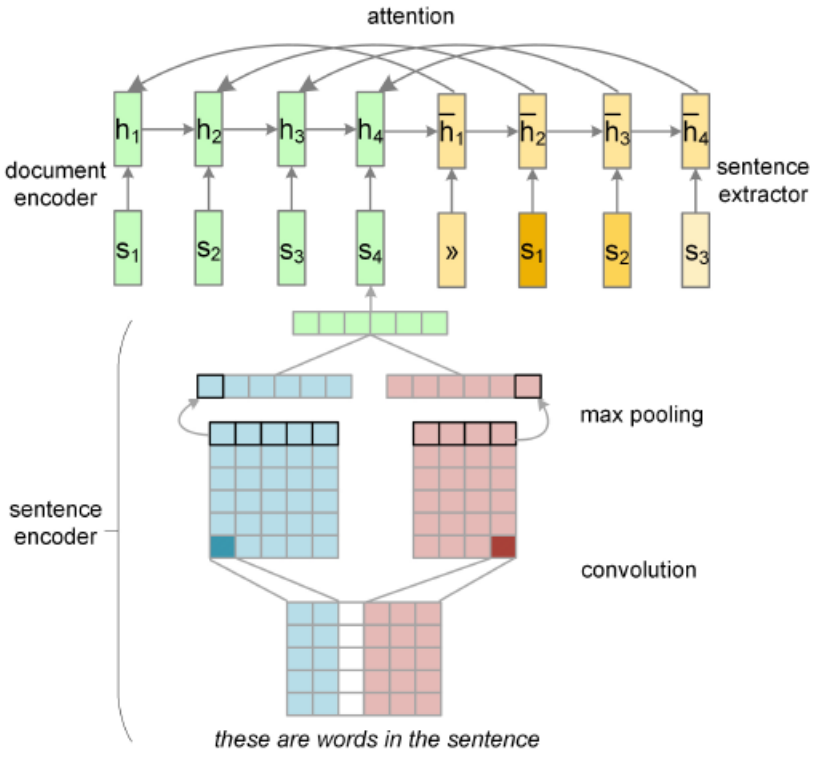


Figure 3.2: A recurrent convolutional document reader with a neural sentence extractor [19].

A document reader in the Neural Summarization Model is used to derive meaningful representations from the given sentences. Convolutional sentence encoder has been used to effectively train single layer CNNs and for Sentiment Analysis. Recurrent Document Encoder has been used for achieving minimum compression.

"A Hierarchical Model for Text Auto-summarization" [20] discussed Summariza-

tions which can be both abstractive and extractive methods, where abstractive methods are identical to summaries generated by humans and are generated from source files, on the other hand, extractive methods may not be generated from source files and they calculate the word frequency to regulate the importance of a word in a given sentence. They used an encoder-decoder model to summarize news article into its headline. A hierarchical LSTM is used as an encoder and a normal LSTM is used as a decoder. They performed experiments on the Signal Media News Dataset [21].

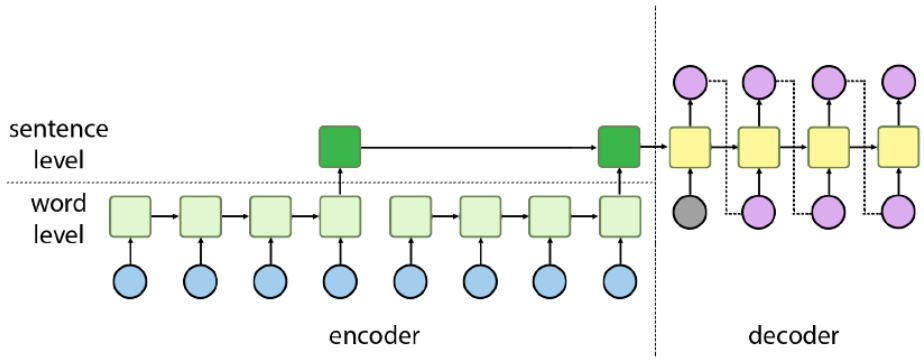
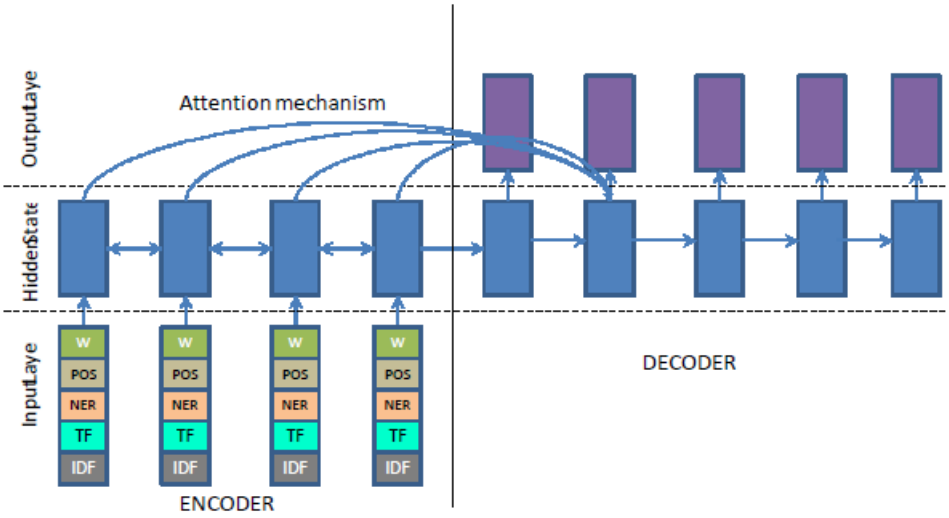
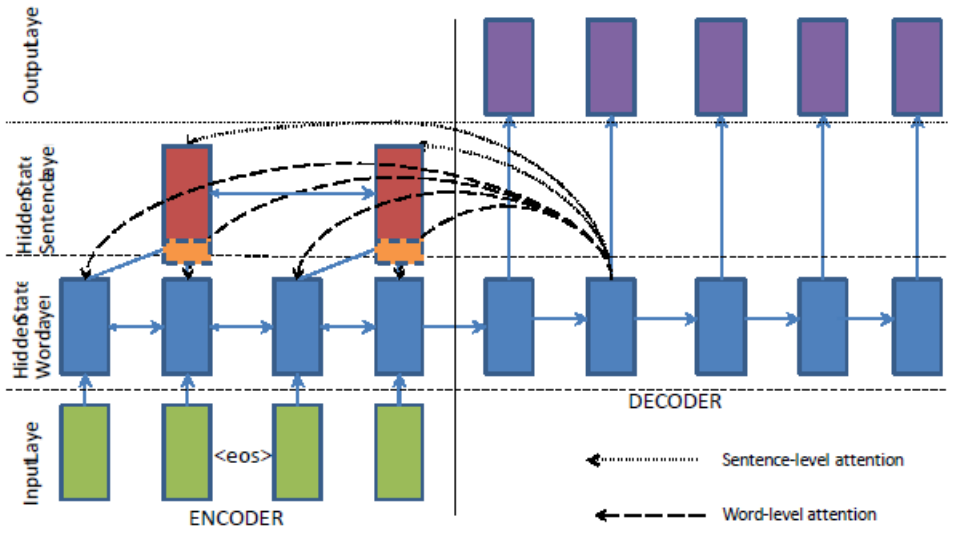


Figure 3.3: The illustration of the hierarchical LSTM encoder-decoder model [20]. We can observe how the hierarchical encoder is designed at both word and sentence level. The sentence level encoded input is then fed into the decoder.

In the paper "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond" [22], they present a state-of-the-art Attentional Encoder-Decoder Recurrent Neural Network architecure. Novel models for summarization are proposed to show an additional improvement in the performance instead of using a machine translated based model. The model uses a bidirectional GRU-RNN encoder and a unidirectional GRU-RNN decoder along with an attention mechanism and a softmax layer. Parts of Speech, named-entity tags and TF-IDF statistics of the words are captured to identify the key concepts and key entities.



(a) Feature-rich-encoder: Using one embedding vector each for POS, NER tags and discretized TF and IDF values, which are concatenated together with word-based embeddings as input to the encoder.



(b) Hierarchical encoder with hierarchical attention: the attention weights at the word level, represented by the dashed arrows are re-scaled by the corresponding sentencelevel attention weights, represented by the dotted arrows. The dashed boxes at the bottom of the top layer RNN represent sentence-level positional embeddings concatenated to the corresponding hidden states.

Figure 3.4: Hierarchical Encoder-Decoder model architectures with statistical features and attention. [22]

Models are trained on the Gigaword corpus instead of tuning them on the DUC validation set [23]. Model performance on the test set is compared using ABS and ABS+ models. A new dataset was proposed for this purpose which was released in two different versions.

3.2 Personalized Text Summarization

Personalized text summaries were mostly performed using a topic modeling, user modeling or using other popular graph based, statistical based algorithMrs. In "Personalized Text Summarization Based on Important Terms Identification" [24], Robert et al. used Latent Semantic Analysis as the main model. Annotations from the user's are used to determine personalized content. Considering the differences in reader's characteristics helped them generate better personalized summaries. They have experimentally evaluated the proposed method in the domain of learning/education. The model was capable of extracting important concepts explained in the document when considering the relevant domain terms in the process of summarization.

Work published in "User-model based personalized summarization" [25], deals with performing user modeling on the user profiles. Diaz et al. propose a single document summarization model. The user model is capable of storing long and short term interests using for reference systems: sections, categories, keywords and feedback terMrs. Important sentences are selected to be included in the summary, based on the user model.

In "Aspect-Based Personalized Text Summarization" [26], Berkovsky et al. investigate the user attitude towards personalized summaries generated from a coarse-grained user model based on document aspects. Their results show that the better the fit between real user model and the user model on which the summary is based, the higher the user's rating for the summary. Evaluating the perceived faithfulness of a summary to the original document did not show a significant difference between personalized and general summaries.

CHAPTER 4: PROPOSED MODEL

The proposed model is an extractive text summarization model using deep learning, followed by a personalization framework to generate personalized user summaries.

For extractive text summarization, we have worked with a variant of RNNs called the bi-directional RNN (BRNN). The concept of BRNNs is that to split the neurons in regular RNNs into two opposite directions. BRNNs do not require the input data to be fixed and their future input information can be reached from the current state. BRNNs connect two hidden layers of opposite directions to the same output. By this structure, the output layer can get information from the past and the future states. LSTM cells are the popular choice of memory cells in BRNNs. BRNNs have shown promise when applied to the task of speech recognition.

Our model consists of a three-layered BRNN with LSTM cells followed by a soft-max layer at the end. Inputs to the model are sentences from each document and outputs will be probabilities of class labels 0 or 1. The sentences are represented using the word embeddings matrices. The first layer of the RNN runs at the word level, and computed hidden states sequentially for each word position, based on the current word embeddings and the previous hidden state. Another RNN runs backwards from the last word to the first, and thus this mechanism is called as a Bi-Directional RNN.

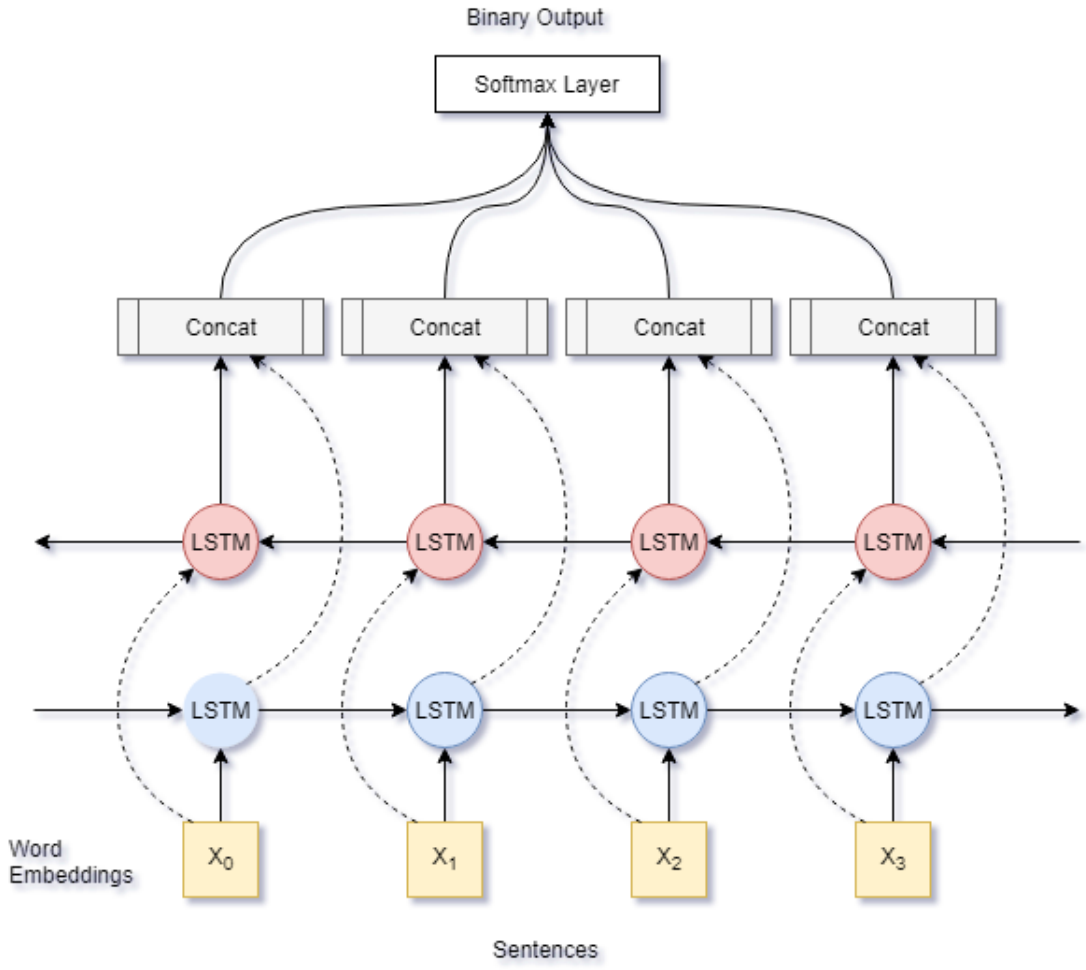


Figure 4.1: The illustration of the proposed BRNN with LSTM cells. The sentences are represented using word embeddings in the embedding layer. Outputs from the BRNN LSTMs are concatenated to obtain a combined representation of each sentence in both the directions before sending it to the softmax layer.

We minimize the negative log-likelihood of the observed labels at training time. Here X is the document representation using word-embeddings and Y is the vector containing summarizing classifier labels.

$$L(y, y^p) = -y * logy^p + (1 - y) * log(1 - y^p) \tag{4.1}$$

In equation 4.1, y is the actual class of the sample and y^p is the predicted probability of the class for the sample. Here we use the natural log.

Because the input sentences are variable in length, we standardized all the inputs

to represent the maximum length in each batch. This is done by adding PAD tokens at the end of each sentence to match the maximum length. The padding token was considered for both scoring and loss functions, making sure that the model doesn't update parameters that should not be updated.

The model predicts on an average of 8-10 sentences as "summary" class out of the average 18 sentences in the document. Thus, the model prediction has a high recall rate of extracting key sentences in the document. During summary generation, four sentences with highest probabilities in the "summary" class are selected to be included in the summary.

For personalized text summarization, we first extract the user's interests from the user profile using Named Entity Extraction (NER). NER Classifies named entities into categories such as persons, organizations, events, locations, expressions, etc. To pick the personalized summaries, we select the top 4 sentences which contain the user entities.

CHAPTER 5: EXPERIMENTAL SETUP

The experiments conducted in this research are presented in this chapter. The experiments are performed to better understand the behavior of the proposed extractive text summarization techniques along with the personalization of summaries based on user's interests. The motive of the experimental setup is to evaluate the proposed techniques. It analyzes how accurate the generated extractive summaries are and how our personalization algorithm increases the readability of a summary by the user.

The experiments for Extractive Text Summarization are performed on the CNN and Daily Mail datasets using the BRNN sequence classifier model with LSTM and GRU variants. We have not used the popular DUC datasets for our experiments as the corpora is not large enough to train deep learning models.

5.1 Corpora

For our experiments, we have used the CNN and Daily Mail corpus originally constructed by Hermann et al. [27] for the task of passage-based question answering, and re-purposed for the task of document summarization as proposed in Cheng and Lapata 2016 [19] for extractive summarization and abstractive summarization [28]. The joint CNN and Daily Mail corpus contains 216,475 newspaper articles with abstractive summaries and extractive labels for each sentence in the document. The dataset is divided into training (193,982), validation (12,147) and test (10,346) documents. On average, there are about 28 sentences per document in the training set, and an average of 3-4 sentences in the reference summaries. The average word count per document in the training set is 802.

Another dataset we used to perform out-of-domain evaluation is The Signal Media

One-Million News Articles Dataset [21]. The dataset is released by the Signal Media to facilitate conducting research on newspaper articles. It is an open source dataset and can be obtained easily by sending in a request. As the name suggests, it contains one million newspaper articles which are mostly in English. The dataset did not contain human generated reference summaries, but we considered the headlines as reference summaries. Though this dataset has hardly been experimented with, we hope the results we found will help other researchers gain interest in it. The number of individual unique sources are over 93k. The dataset contains 265,512 Blog articles and 734,488 News articles. The average length of an article is 405 words.

For the user profile information, we used a dataset called CASIA-crossOSN [29]. It is a Cross-Network User Dataset from Chinese Academy of Sciences. The dataset contains rich user metadata and historical behaviors in YouTube and Twitter, including basic user profiles in YouTube and Twitter, their social relations and tweeting data in Twitter, their three kinds of video behaviors in YouTube as well as rich video metadata for all the collected videos. The cross-network activities together record people's integral online footprints and reflect their demographics as well as interests from different perspectives.

5.2 Data Preprocessing

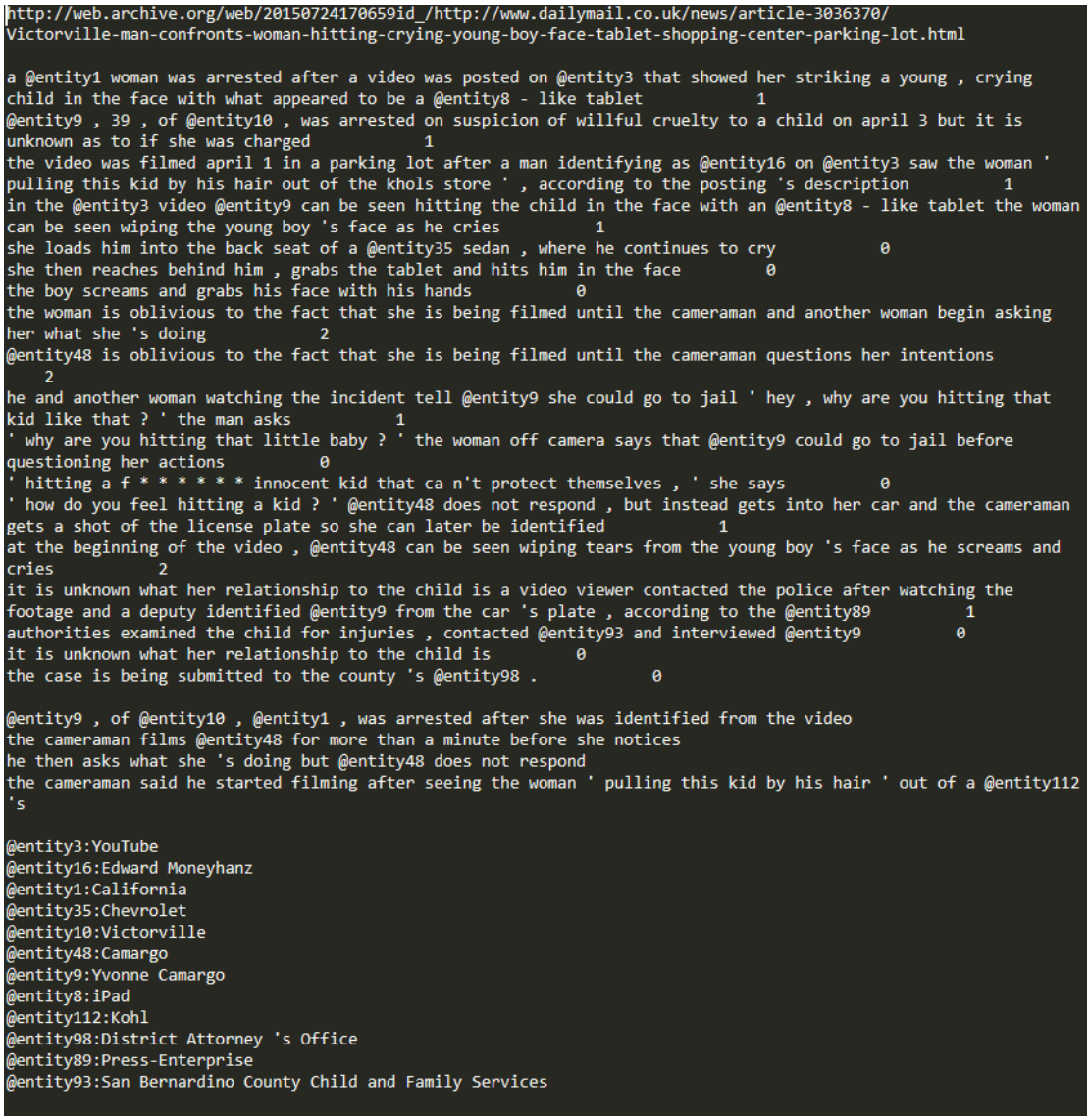


Figure 5.1: Example raw document from Daily Mail dataset. We can see that the document is divided into url and headline, followed by main article divided into sentences with their respective class labels, followed by reference gold summary and entities with their values.

Preprocessing of the text was one of the most important steps before we performed text summarization tasks, as we can see from Figure 5.1, the terms in the documents have many structural variants. Here, the label 2 means "you may include this sentence in the summary". To simplify the process, we have converted all the label 2 sentences

to label 0 sentences. The first task was to replace the referenced entities in each document with their actual words. Then, the terms in the documents are tokenized. The tokenized words are then checked for contractions and are replaced by contracted words i.e. "can't" is replaced by "cannot". Stopwords are removed from the corpus as it will help reduce the dimensionality of the problem space. For extractive text summarization, since we generate just the classifier labels for sentences and are not worried about generating words for summaries, we performed stemming of words to reduce them to their root forMrs. In linguistic morphology and information retrieval, stemming is the process of reducing infected words to their word stem, base or root form. We used the stemming porter2 algorithm from python's NLTK library. The vocabulary size was 416,253 words after the preprocessing was done.

5.3 Word Embeddings

Semantic vectors (also known as word embeddings) let you compare word meanings numerically. For this research, we have used a set of emerging semantic vectors called the ConceptNet Numberbatch [30]. ConceptNet Numberbatch consists of state-of-the-art semantic vectors (also known as word embeddings) that can be used directly as a representation of word meanings or as a starting point for further machine learning. ConceptNet Numberbatch is part of the ConceptNet open data project. ConceptNet provides lots of ways to compute with word meanings, one of which is word embeddings. ConceptNet Numberbatch is a snapshot of just the word embeddings.

It is built using an ensemble that combines data from ConceptNet, word2vec [31], GloVe [32], and OpenSubtitles 2016, using a variation on retrofitting. According to Luminoso Technologies, Inc., it is an ensemble of the word embeddings from word2vec and GloVe and they claim it's better than its parts. The English language vocabulary for ConceptNet is currently at 484,557 words with 300-dimensional floating-point vectors. The following is the evaluation done by Luminoso Technologies, as part of

SemEval 2017 task 2 and the comparisons show promise that the ConceptNet word embeddings significantly outperform other traditionally used word embeddings.

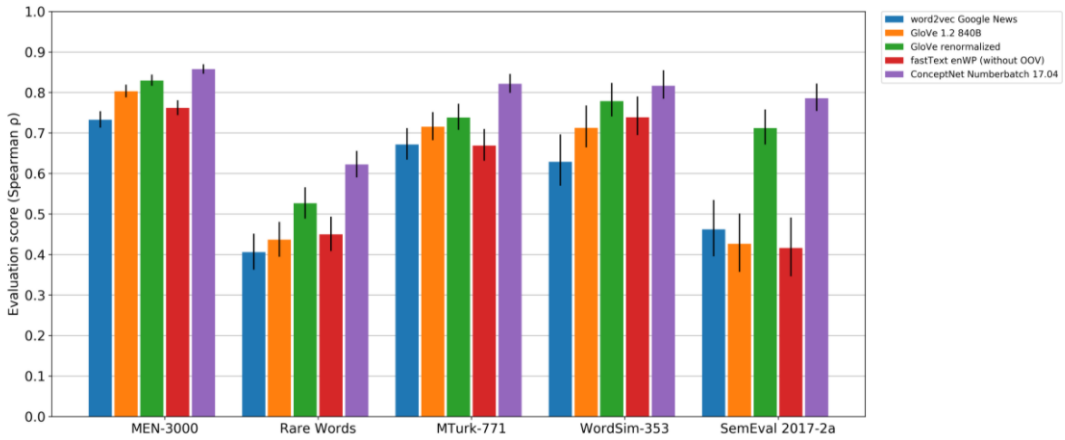


Figure 5.2: Evaluation of Conceptnet Numberbatch word embeddings [30]. In SemEval 2017 task 2 [33], Conceptnet outperformed the traditional word2vec [31] and GloVe [32] models on different corpora.

We created the word-embedding matrix on our vocabulary using ConceptNet vectors. Since 400k words can be too much for the model to work on, we reduced the vocabulary size by creating a count threshold of at least 15 occurrences. We found that some words are missing from ConceptNet or are below the count threshold, so we created randomly initialized vectors for these words. Ultimately, that left us with 88037 unique words useful for the model, which is around 21% of the vocabulary. We then created the word embedding matrix of shape (88037, 300).

5.4 Data Preparation

Using the word embedding matrix, each sentence of words is converted into a sequence of 300-dimensional vectors. The starting of the sentence is represented using the token "<GO>", the unknown words are represented using the "<UNK>" token, and end of the sentence is represented using the "<EOS>" token. We found that 999119 words out of the total 76674995 words were unknown words. That is merely 1.3% of the total. The training documents were split into sentences with

their respective classification labels. The 99th percentile of length of sentences was recorded at 56 words. The sentences were divided into batches of 128 sentences each. The sentences have been padded with "<PAD>" tokens to meet the longest length in their batch. For a subset of data (50,000) sentences, we added "<PAD>" tokens randomly to ensure uniform learning in all neurons.

The target values have been encoded using the one-hot encoder to make sure they're in the shape of mx2 shape, where m is the number of samples in the dataset. The summaries are generated using the labels predicted by the model.

5.5 Model Settings

The baseline of the model is a 2-layered BRNN with each 300 neurons followed by a soft-max classification layer in the end. Variants of the RNN cells and the model architectures were tried to achieve best results. Bi-directional RNNs with LSTM cells and GRU cells were some of the main variants used in the experiments. All the RNN layers in the models were regularized by using dropout regularization [34]. Dropout is a simple regularization approach, where at every training step, every neuron has a probability p of being temporarily 'dropped out', meaning it will be entirely ignored during this training step, but it may be active during the next step. This brutal technique is really useful in practice as it makes the model adapt to different conditions and avoids relying on any single neuron for prediction. Another way to understand the power of dropout is to realize that a unique neural network is generated at each training step. After training, neurons don't get dropped anymore.

In the case of single directional RNNs, the final state of the RNNs were considered for the soft-max classification layer to predict the labels. In the case of BRNNs, the final state of concatenated forward and backward states was considered for the final layer. The model is trained using the Adam Optimizer [35]. Adam Optimizer, which stands for "adaptive moment estimation", is a hybrid of multiple optimizers like Momentum Optimization and RMSProp. It is one of the most popular choices

for optimization of neural networks as it's an adaptive learning rate algorithm and it requires less hyperparameter tuning. To avoid, exploding/vanishing gradients, gradient clipping [36] was performed. Gradient clipping is a technique used to clip the gradients, during backpropagation to prevent them exceeding some threshold.

The model was trained with a starting learning rate of 0.005, batch size of 128 and for maximum of 100 epochs. The sigmoid cross-entropy loss is checked after every 3 epochs and if the loss doesn't decrease after three update checks, early stopping of training is done. Learning rate decay is performed after each epoch at 0.95 times the current learning rate and the minimum learning rate is maintained at 0.0005.

5.6 Evaluation Metrics

In our experiments, we evaluate the performance of our extractive summarizer model using different variants of the ROUGE metric [37] computed with respect to the reference summaries. The proposed models are independently evaluated and compared with the state-of-the art summarization techniques. ROUGE stands for Recall-Oriented Understudy of Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts. It works by comparing an automatically produced summary with a human-produced reference summary. Below is the list of ROUGE metrics used to evaluate automatic summaries:

1. ROUGE-N measures the N-gram units common between a particular summary and a collection of reference summaries where N determines the N-gram's length [38]. E.g., ROUGE-1 for unigrams and ROUGE-2 for bi-graMrs.

2. ROUGE-L computes longest common subsequences (LCS) metric [39]. LCS is the maximum size of common subsequences for two given sequences X and Y. ROUGE-L calculates ratio between size of two summaries' LCS and size of reference summary.

3. ROUGE-1: Overlap of unigram between the automatic and reference sum-

maries.

4. ROUGE-2: Overlap of bigrams between the automatic and reference summaries.

For intrinsic evaluation of the summaries, other popular metrics used were precision, recall and F-measure. They are required to predict coverage between human-made summary and automatically generated machine-made summaries. These metrics are explained below:

1. Precision: Precision is the fraction of the sentences chosen by the humans and selected by the system are correct.

$$Precision = \frac{|RelevantSentences \cap RetrievedSentences|}{RetrievedSentences} \tag{5.1}$$

2. Recall: Recall is the proportion of the sentences chosen by humans are even recognized by the machine.

$$Recall = \frac{|RelevantSentences \cap RetrievedSentences|}{RelevantSentences} \tag{5.2}$$

3. F-measure: F-measure is a hybrid of both precision and recall. It is calculated using the formula below:

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.3}$$

CHAPTER 6: RESULTS

In this chapter, we present and discuss the results of experiments, which are independently (or combinedly) performed for the extractive text summarization along with personalization.

For the extractive text summarization task, we use the combined CNN and Daily Mail news datasets. The base model is a Bi-Directional Recurrent Neural Network with a classification layer at the end. The task essentially is a variable sequence classification task where the summary labels for sentences are used to create the summaries. We explored a variety of models with different layers and memory cells. We started with a 2-layered BRNN model with LSTM cells and kept adding layers to achieve the best results. We used hidden sizes of 300, 400, 512 and batch sizes of 64, 128 and 256. We found the best parameters were 300 hidden unit size and batch size of 128. At test time, we pick top four sentences with class label 1 (we should include the sentence in summary) which have highest predicted probabilities, to be included in the summary.

The training was done on Telsa K80 GPUs and occasionally on Telsa K100 GPUs. For the extractive summarization model on the CNN, Daily Mail combined datasets, the training for each epoch took around 4 hours. As a result, due to the computational limitations, the training was done only for maximum 3-4 epochs for each model. The results are evaluated with extractive summaries as well as the gold reference summaries.

6.1 Quantitative Evaluation

We report the performance of our extractive summarization models on the joint CNN/Daily Mail corpus. Nallapati et al.[19] and SummaRuNNer [18] are the only ones that report performance on this dataset. Nallapati et al. is an abstractive encoder-decoder based model, in which they use full-length F1 as metric and the same is followed by SummaRuNNer which is a GRU based RNN model for extractive summarization. In order to do a fair comparison with their work, we use the same metrics as them. On this dataset, our proposed models are outperformed by SummaRuNNer's extractive and abstractive summarization models. The high recall scores show that the proposed models work well in extracting the relevant sentences as we use the sentence-level extractive labels to train our model.

Table 6.1: Performance of extractive summarization of 3-layered BRNN with LSTM cells on the extractive summaries of entire CNN, Daily Mail test set using Rouge-1, Rouge-2 and Rouge-L.

	Precision	Recall	F1
Rouge-1	0.747	0.925	0.824
Rouge-2	0.688	0.885	0.771
Rouge-L	0.695	0.887	0.749

According to Table 6.1, the baseline model 3-layered BRNN with LSTM cells has performed well on the extractive summaries generated using the class labels. Although, the precision is relatively less (0.688) for the Rouge-2 metric, the model seems to perform well on extracting relevant documents resulting in high recall. The BRNN with LSTM cells has constantly outperformed BRNN with GRU cells. We think this is largely because of advantages of LSTM cells that they deal with complicated data in a better way by memorizing and forgetting cells.

Table 6.2 shows the performance comparison of our extractive summarization models with the state-of-the-art model of Cheng et al.[19] and Nallapati et al. [28] and other baselines. While our model performs poorly when compared to the state-of-the-art, we believe one of the causes is the computational limitation of not able to train the model until it converged, as each epoch of training took around 4 hours and we had to stop early after 3-4 epochs. The number at the end of the name "Proposed 3-BRNN LSTM - 4", determines the top number of sentences selected at the summary generation phase. Since, top 4 sentences gave us the best results we

In Table 3 we present the recall metrics of Rouge-1, Rouge-2 and Rouge-L. This shows that even if the precision of extracting gold summary of our models is low, the high recall scores indicate that the model is better at picking the best sentences for summarization. The results are a demonstration of the difficulty of using F1 metric as an evaluation metric as our proposed models deal with extracting key sentences resulting in high recall at the expense of precision.

Table 6.2: Performance comparison of extractive summarization models on the gold summaries of entire CNN, Daily Mail test set using full length F1 variants of Rouge-1, Rouge-2 and Rouge-L.

	Rouge-1	Rouge-2	Rouge-L
Lead-3	39.2	15.7	35.5
SummaRuNNer-abs	37.5	14.5	33.4
SummaRuNNer	39.6	16.2	35.3
(Nallapati et al. 2016)	35.4	13.3	32.6
Proposed 3-BRNN LSTM - 4	32.3	12.2	30.1
Proposed 3-BRNN GRU - 5	29.3	11.2	26.7
Proposed 3-BRNN LSTM - 3	31.4	10.3	29.4

Table 6.3: Performance of extractive summarization models on the gold summaries of entire CNN, Daily Mail test set using full length Recall variants of Rouge-1, Rouge-2 and Rouge-L.

	Rouge-1	Rouge-2	Rouge-L
Proposed 3-BRNN LSTM - 4	81.2	43.6	58.7
Proposed 3-BRNN GRU	62.4	26.3	44.4
Proposed 3-BRNN LSTM	87.3	44.5	60.2

To perform out of the domain evaluation, we have evaluated the trained models on signal-media one million news dataset. The task is now converted to generate headlines by selecting top 1 sentence in the "summary" class. The evaluation is done by dividing 120,000 samples into the test set and remaining into train set. We compared our work with Hujia et al [40].

Table 6.4: Performance comparison of extractive summarization models on the headlines of Signal-Media One Million News test set using F1 variants of Rouge-1 and Rouge-2.

	Rouge-1	Rouge-2
A-4-RNN-LSTM	20.52	5.48
A-3-RNN-LSTM	18.77	5.17
Proposed 3-BRNN LSTM - 1	13.5	4.43

6.2 Qualitative Evaluation

6.2.1 Generic Summaries

Document:

["a Walmart employee has been applauded for his honesty after finding $ 4,400 worth of cash in the parking lot and handing it over to authorities , who returned

it to its relieved owner Cassidy was picking up trash outside the store in Bangor , Maine on thursday when he discovered the wet stack of money beneath a large piece of paper in the parking lot he immediately contacted store security to tell them what he had found and , realizing it was a sizable sum , they then contacted the police department , sergeant Cotton said officer Dustin Dow went to the store to take the report and learned that a man had visited the security office last winter to claim he had lost about $ 4,000 in the parking lot great work : Cassidy , right , is pictured being presented with a Bangor Police Department Challenge coin by an officer after he found $ 4,400 cash in a Walmart parking lot and alerted security the man , who was later identified as Chen , said at the time that he had put the money in an envelope in his pocket after leaving work at a nearby restaurant he had intended to send the money home to his family , Cotton said but after clearing snow off his car and driving home , he realized that his pocket was empty Chen went to the security office to ask if the money had been turned in or if surveillance footage had revealed what could have happened to it , but nothing emerged when Dow heard of the story , he went to the nearby restaurant , Kobe Steakhouse , and asked for the man , and Chen confirmed he had misplaced the cash after counting the money at the station , the officer returned to the restaurant to give it back to him - and to take a picture showing him gladly accepting it returned : the money belonged to Chen , who lost the $ 4,000 as he cleared snow off his car last winter after leaving his job at a nearby restaurant he told Walmart about it at the time but they could not find it stash : Cassidy found the money , pictured , while picking up trash near the Walmart parking lot last thursday Cotton applauded Cassidy for his honesty , particularly because the Walmart employee found the money while working at a job that does not pay much 'Cassidy is not getting rich doing this and has dealt with other issues over the past year , ' Cotton explained in a lengthy Facebook post 'Cassidy , like many people , has even had to sleep in his car for a time when things were rough ' in recognition

of his honesty , Cassidy was presented with a Bangor police department challenge coin' for being honest and having great integrity in doing his job', the department said ' Cassidy personifies what we hope is in all of us, 'Cotton wrote 'Thanks Brian, the men and women of the Bangor Police Department salute you ' scene : Cassidy , who works at this Walmart in Bangor , Maine , was thanked for his honesty and integrity"]

Reference Summary:

["Cassidy was picking up trash from the parking lot of the Bangor , Maine store last thursday when he found the wet stack of cash he immediately contacted security and police were called they discovered that a man who worked at a nearby restaurant , Chen , had reported losing the money while cleaning snow off his car last winter police returned the cash to Chen and thanked Cassidy for his integrity by awarding him a police deparment coin"]

Predicted Generic Summary:

["a Walmart employee has been applauded for his honesty after finding $ 4,400 worth of cash in the parking lot and handing it over to authorities , who returned it to its relieved owner. Cassidy was picking up trash outside the store in Bangor , Maine on thursday when he discovered the wet stack of money beneath a large piece of paper in the parking lot. officer Dustin Dow went to the store to take the report and learned that a man had visited the security office last winter to claim he had lost about $ 4,000 in the parking lot. after counting the money at the station , the officer returned to the restaurant to give it back to him - and to take a picture showing him gladly accepting it"]

Discussion:

The reference summary gives us a good understanding of the main document by stating important facts. But the predicted summary gives a good overview of who "cassidy" is and how he found the money and handed it over to the authorities. This

is a good example where the predicted summary is more coherent and easy to read and understand than the human generated summary.

6.2.2 Personalized Summaries

Document:

["Google launched a new US wireless service today that switches between Wi-Fi network and cellular networks Google is already the world 's most popular phone software provider , and a pay - tv operator - and now it wants to be your mobile network carrier the company has unveiled a US wireless service that switches between Wi-Fi network and cellular networks to curb data use and keep phone bills low the service , called ' Project Fi , ' debuted today , about two months after Google revealed its plans to expand its ever - growing empire into providing wireless connections for smartphones Google is selling the basic phone service for $ 20 a month and will only charge customers for the amount of cellular data that they use each month , instead of a flat rate each gigabyte of data will cost $ 10 a month that means a customer could sign up for a plan offering three gigabytes of data and get $ 20 back if only one gigabyte was used in a month most wireless phone carriers allow their customers to roll over unused data into another month of service without refunding any money Project Wi-Fi initially will only be sold to a narrow US audience that owns the Nexus 6 6 , a smartphone that Motorola Mobility made with Google 's help Google 's pricing setup makes Project Fi less expensive than most of the comparable plans offering by the four biggest wireless phone carriers - Verizon , AT&T , T-Mobile and Sprint Corp the monthly prices for a single line of smartphone service with up to one gigabyte of cellular data at those carriers range from $ 45 to $ 50 compared to $ 30 from Google the major carriers , though , offer a variety of family plans that could still be better deals than Project Fi those bundled plans allow several phone lines to share a pool of cellular data rather than building its own network , Google is leasing space on cellular towers built by Sprint Corp and T-Mobile , which are hoping the deals will boost their

profits without costing them too many customers tempted to defect to Project Fi to use the service , Nexus 6 6 owners must sign - up to request an invitation , must have a Gmail address , and must live in a US zip code within the coverage area in this map , the dark green areas are covered by the service 's 4G LT , the lime green is covered by 3G and the pale green by 2G Verizon LTE coverage is pictured for comparison Project Fi will be hosted through Sprint Corp and T-Mobile 's networks the service will work only on the company 's Nexus 6 6 phones and only in the US Project Fi will be hosted through Sprint Corp and T-Mobile 's networks overall , it costs $ 20 for basic service , which includes unlimited domestic talk , unlimited texting , tethering , and access in 120 countries customers pay $ 10 per gb of data data is paid for in advance , and the cost of unused data gets refunded rather than rolled over or lost phone numbers will live in the cloud so that consumers can talk and text on any connected tablet calls can be made via Hangouts on Android , iOS , and through Gmail on desktops , via the Hangouts widget texts can be made and received in the same way there is no annual service contract required when you sign up to use the service , Nexus 6 6 owners must sign - up to request an invitation , must have a Gmail address , and must live in a US zip code within the coverage area Google is promising Project Fi will automatically switch over to an available Wi-Fi network if that is running at a higher speed than the cellular alternatives T-Mobile ceo Legere , whose company already has been cutting its prices and rolling out new options , said it was a ' no - brainer ' to work with Google on Project Fi ' anything that shakes up the industry status quo is a good thing - for both US wireless customers and T-Mobile , ' Legere wrote in a blog post Google has an incentive to promote cheaper and faster wireless service as a mobile virutal network operator this is because it operates some of the world 's most popular online services , including its search engine , maps , Gmail and YouTube video site the Mountain View , California , company believes most people will visit those services more frequently if they are enticed to stay online

for longer periods , giving Google more opportunities to show the digital ads that generate most of its revenue similar motives prompted Google to begin building high - speed , hard - wired networks capable of navigating the internet at speeds up to 100 times faster than existing broadband services although Google is only selling its broadband service in a handful of US cities so far , AT&T and Comcast are now offering options with comparable speeds in a few communities price in dollars : this graphic shows how much Google 's new Project Fi will cost for various plans for Talk and Text only , the Google carrier will cost $ 20 a month as compared to $ 35 for new partner T-Mobile however the savings really start when 5GB of LTE is consider - with the internet search giant 's plan coming in at $ 70 and telecom giant Verizon 's costing $ 110 a month Project Fi initially will only be sold to a narrow US audience that owns the Nexus 6 6 , a smartphone that Motorola Mobility made with Google 's help"]

User profile:

["I am your atypical car and technology nut, Recovering Twitter Addict. I enjoy autos travel tech and photography I also enjoy reading fiction and learning new technology. I am a Microsoft, Apple, Google fan. Did I say I like cars Estudiante de Espa ol Soy latino honarario"]

Extracted user entities:

{'Addict', 'Apple', 'Estudiante de Espa', 'Google', 'Microsoft', 'autos', 'car', 'fan', 'honarario', 'nut', 'photography', 'reading fiction', 'technology'}

Reference Summary:

["Project Fi will be hosted through Sprint Corp and T-Mobile 's networks it costs $ 20 for basic service and unused data is paid back to customer the invitation - only service will work only on Nexus 6 phones in the US numbers will live in the cloud so users can talk on any connected tablet"]

Predicted Personalized Summary:

["**Google** launched a new US wireless service today that switches between Wi-Fi network and cellular networks. The service , called ' Project Fi , ' debuted today , about two months after **Google** revealed its plans to expand its ever - growing empire into providing wireless connections for smartphones. **Google** is selling the basic phone service for $ 20 a month and will only charge customers for the amount of cellular data that they use each month , instead of a flat rate. T-Mobile CEO Legere , whose company already has been cutting its prices and rolling out new options , said it was a ' no - brainer ' to work with **Google** on Project Fi"]

Discussion:

Here, we present an example of a personalized summary. We first extract the user interests/entities from the user profile information. Then we perform text summarization on the entire document to capture the "summary" class classified sentences. Among these sentences, we include the top-4 sentences which contain user entities.

The document is about a new wireless service released by Google. The reference summary does a good job summarizing the key statements in the document. However, the reference never mentions the word Google. The user mentioned that he is a Google fan. The model's predicted personalized summary creates a summary around the theme "Google" by selecting sentences which contain the entity "Google" from the "summary" class sentences.

The personalized summary is definitely more intriguing to the user than the generic reference summary. It also tries to cover most of the important information in just four sentences. However, as we can see, the entities created from the user's profile include words like "addict", "nut", etc. which necessarily do not portray user's interests. To overcome such issues, our framework can be improved to be a more robust rating based personalization framework.

CHAPTER 7: CONCLUSIONS

In this research, we have built a simple but robust deep learning model to perform extractive text summarization which uses a relatively new set of semantic word vectors. We have also proposed a framework to create personalized summaries based on user profile data to make the summaries more interesting and intriguing to the user.

We have proposed a new model for extractive text summarization, which performs the summarization on documents by classifying sentences into summary and non-summary classes. The model then selects top-4 sentences in the summary class to be included in the predicted summary. We have performed experiments on the CNN Daily Mail News and The Signal Media One Million News articles datasets. We compared the results with the state-of-the-art and the proposed model gives us reasonably good Rouge metrics, given the fact that the models were trained only for 6 epochs maximum due to the computational limitations. We have also observed that the models give us relatively high recall scores.

We also proposed a new personalization framework which works on personalizing the sentences predicted as "summary" class based on user's interests. We used Name Entity Recognition to attain the user's interests from their profile data. We then used these entities/interests to pick the sentences which the user might be interested in reading. The model then selects top-4 interesting sentences to include in the summary. Thus, the proposed approach will add a new value to the summarization model and engage the users. Though our experiments were conducted on newspaper articles, we believe that the application to such personalized summarization techniques are endless.

7.1 Future Scope

The deep learning models are computationally expensive and take immense GPU and time resources to train fully on such large datasets. Our models were trained on high-performance GPUs for a maximum of 8 hours, but the models never really converged. In future, the models can be trained further to make sure they converge.

Building the personalization framework was a challenge and the proposed framework gives us good results. We have performed quality and Rouge evaluation on the personalized summaries. There is scope to make the personalization framework more scalable and robust by integrating user feedback as one of the features to aid the personalization.

REFERENCES

[1] "Idc digital universe study: Big data, bigger digital shadows and biggest growth in the far east." Sponsored by EMC.

[2] A. Regalado, "The data made me do it." https://www.technologyreview.com/s/514346/the-data-made-me-do-it/, 2013.

[3] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[4] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of emerging technologies in web intelligence*, vol. 2, no. 3, pp. 258–268, 2010.

[5] M. A. Fattah and F. Ren, "Ga, mr, ffnn, pnn and gmm based models for automatic text summarization," *Computer Speech & Language*, vol. 23, no. 1, pp. 126–144, 2009.

[6] M. A. Fattah, "A hybrid machine learning model for multi-document summarization," *Applied intelligence*, vol. 40, no. 4, pp. 592–600, 2014.

[7] K. Kaikhah, "Automatic text summarization with neural networks," in *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, vol. 1, pp. 40–44, IEEE, 2004.

[8] K. Kaikhah, "Text summarization using neural networks," 2004. Department of Faculty Publications- Computer Science, Texas State University, eCommons.

[9] M. R, "Textrank: bringing order into texts," 2004. In: Conference on empirical methods in natural language processing. pp 404-411.

[10] J. Kleinberg, "Authoritative sources in a hyperlinked environment," 1999. Journal of the ACM, 46(5):604-632.

[11] P. Herings, "Measuring the power of nodes in digraphs," 2001. Technical report, Tinbergen Institute.

[12] R. D, "Lexrank: graph-based lexical centrality as salience in text summarization," 2004. J Artif Intell Res 22:457-479.

[13] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2017.

[14] S. Hochreiter, "Long short-term memory," 1997. Neural computation.

[15] K. C. et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014. arXiv:1406.1078v3.

[16] O. V. Ilya Sutskever and Q. V. Le, "Sequence to sequence learning with neural networks," 2014. Advances in neural information processing systems.

[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[18] F. Z. Nallapati, Ramesh and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," 2017. AAAI.

[19] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words.," 2016. arXiv preprint arXiv:1603.07252.

[20] Z. Zhou, "A hierarchical model for text autosummarization."

[21] D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?," 2016. Proceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016. Pages 42-47.

[22] e. a. Nallapati, Ramesh, "Abstractive text summarization using sequence-to-sequence rnns and beyond.," 2016. arXiv preprint arXiv:1602.06023.

[23] "Duc dataset." https://duc.nist.gov/duc2004/.

[24] R. Moro, "Personalized text summarization based on important terms identification.," 2012. Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on, IEEE.

[25] A. Diaz and P. Gerva, "User-model based personalized summarization.," 2007. Information Processing & Management 43.6.

[26] S. Berkovsky, T. Baldwin, and I. Zukerman, "Aspect-based personalized text summarization," in *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 267–270, Springer, 2008.

[27] K. M. H. et al., "Advances in neural information processing systems.," 2015. pages 1684-1692. Curran Associates, Inc.

[28] R. Nallapati, B. Xiang, and B. Zhou, "Sequence-to-sequence rnns for text summarization," 2016.

[29] J. S. Ming Yan and C. Xu, "Mining cross-network association for youtube video promotion," 2014. ACM Multimedia, Orlando, Florida, USA.

[30] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," 2017.

[31] e. a. Mikolov, Tomas, "Efficient estimation of word representations in vector space.," 2013. arXiv preprint arXiv:1301.3781.

[32] R. S. Pennington, Jeffrey and C. Manning, "Glove: Global vectors for word representation.," 2014. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

[33] J. Camacho-Collados, M. T. Pilehvar, N. Collier, and R. Navigli, "Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (Vancouver, Canada), pp. 15–26, Association for Computational Linguistics, August 2017.

[34] e. a. Srivastava, Nitish, "Dropout: A simple way to prevent neural networks from overfitting.," 2014. The Journal of Machine Learning Research 15.1.

[35] D. P. Kingma and J. Ba., "Adam: A method for stochastic optimization.," 2014. arXiv preprint arXiv:1412.6980.

[36] T. M. Pascanu, Razvan and Y. Bengio, "On the difficulty of training recurrent neural networks.," 2013. International Conference on Machine Learning.

[37] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," 2004.

[38] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," 2003. in Proceedings of the 2003 Conference of the North Amer- ican Chapter of the Association for Computational Linguistics on Human Lan- guage Technology-Volume 1, pp. 7178, Association for Computational Linguis- tics.

[39] C.-Y. Lin and F. Och, "Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics," 2004. in Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 605, Association for Computational Linguistics,.

[40] H. Yu, C. Yue, and C. Wang, "News article summarization with attention-based deep recurrent neural networks,"

APPENDIX A: EXAMPLES OF PREDICTED SUMMARIES

A.1 Example 1

Document: ["A teenage girl who suffered horrific injuries after being dragged behind a car and left for dead in a gutter has broken her silence. on September 7, 2014, Giufre, 19, spent months in hospital after being towed behind a vehicle along a suburban street in Casula, south-west of Sydney, before she was abandoned. so far one man has been charged in relation to the violent incident , but police are searching for others they believe were involved. 'I'm blind in one eye now, I have limited vision in the other eye, I have hearing loss in one of my ears, one side of my nose I can't smell out of,' Mrs. Giufre revealed. The young woman's parents have been by her side since the horrible day, and say her life has been consumed by her recovery for the last six months. 'She's continually at the doctor's, having surgeries, her whole life is just consumed with what's happened to her,' Mrs. Giufre's mother Karen said. Mrs. Giufre told 9News her life has been 'destroyed' by the violent attack which has left her with permanent disabilities her parents Karen (left) and Frank (right) said their daughter's life is now' consumed' by what happened one of Mrs. Giufre's alleged attackers, 18-year-old Hawchar, was supposed to front court on Wednesday but instead sent legal representation 'I look at Sam and I think that she's been held prisoner, and someone's thrown the key away, and we're just trying to find that key for her, 'her father Frank revealed. On Wednesday Mrs. Guifre fronted court with the hopes of confronting one of her alleged attackers, 18-year-old Hawchar. However he instead sent legal representation, dealing the victim of the senseless crime another blow. On his behalf, Hawchar's lawyer plead not guilty to four charges, including dangerous driving occasioning grievous bodily harm. 'He needs to know how it feels, you know, how I feel, 'Mrs. Guifre told 9News. The young woman has urged anyone else who was in the car to come forward last October when Mrs. Giufre was still in

hospital, her parents spoke about their grave concerns that their daughter would have to learn to walk, talk and eat again. Mrs. Giufre was rushed to Liverpool Hospital in September where she spent time in intensive care after suffering multiple fractures to her skull and face, along with bleeding on her brain. 'We don't know whether she'll talk , whether she'll see. We don't know how she will end up, knowing that someone has taken that away from her. She's a good kid, she didn't deserve this,' her mother Karen said at the time. 'The image of Giufre when we saw of her in the hospital...I just want to see her like she was originally. She doesn't look like that any more. 'Doctors believed early on that Giufre may have sustained permanent brain damage and could be blind in one eye following the incident. Footage of the incident, captured by a property's CCTV camera, shows a silver sedan driving down the street as Mrs. Giufre is dragged near the rear door of the vehicle her father Frank said the family was celebrating Father's Day when Giufre decided to go meet up with friends at Casula. Doctors believed early on that Giufre had sustained permanent brain damage and may be left blind in one eye following the incident her cousin gave her a lift about 5 pm and by 6.30pm the family was told Giufre was in the emergency department. 'She had traumatic injuries and god knows what else was happening in her head, 'He said following the horrific incident. 'We've been by her side 24 hours. I've given up work to be by her side. 'The Giufre family say their daughter was a 'good girl' who had just finished a couple of courses through TAFE. 'She didn't have enemies, 'Mr. Giufre said. Her parents pleaded for anyone who witnessed the incident to come forward as police try to track down the driver. CCTV Footage of the incident shows a silver sedan driving down the street as Mrs. Giufre is dragged near the rear door of the vehicle. Just seconds after two people crossed the road with their dogs, the car entered a roundabout where the teenager is believed to have been found. Police have described the incident as a 'cowardly and callous act' and are not treating it as an abduction attempt. Her parents, Frank and Karen Giufre, pleaded for anyone

who witnessed the incident to come forward as police try to track down the driver Mrs. Giufre said last year her daughter would have to learn to walk, talk and eat again 'Giufre was leaning in the rear passenger window...(and) subsequently fell out of the vehicle, 'Liverpool Crime manager Dean Johnston said. 'We're dealing with real victims here...this incident has left a young woman with serious injuries'. 'Due to Giuffre's injuries, investigators have been unable to talk to her. She will have to learn to walk , talk and eat again.' Detectives from Liverpool Local Area Command formed Strike Force O'Loughlin to investigate the incident"]

Reference Summary: ["Giufre, 19, endured the horrific attack last September was dragged behind a car and left in a gutter in Casula, south-west Sydney Mrs. Giufre spent months in hospital and now has permanent injuries has lost vision in one eye, hearing in one ear, and smell in one *nostril* one man has been charged over the violent incident police are still searching for others believed to be involved"]

Predicted Summary: ["We don't know how she will end up, knowing that someone has taken that away from her. She doesn't look like that any more. Footage of the incident, captured by a property's cctv camera, shows a silver sedan driving down the street as Mrs. Giufre is dragged near the rear door of the vehicle her father Frank said the family were celebrating Father's Day when Giufre decided to go meet up with friends at Casula. Doctors believed early on that Giufre had sustained permanent brain damage and may be left blind in one eye following the incident her cousin gave her a lift about 5pm and by 6.30pm the family were told Giufre was in the emergency department"]

A.2 Example 2

Document: ["An undercover police officer who was captured on video ranting at an Uber driver is assigned to the Joint Terrorism Task Force, it has emerged. The NYPD's Internal Affairs Bureau is investigating the incident after the footage was posted online involving detective Cherry. One of the passengers in the car captured

the exchange after the incident in West Village. Detective Cherry was trying to park his Hyundai without indicating when the Uber driver known only as Humayun honked his horn. The angered detective pulled over the Uber cab and launched a tirade, questioning the driver's command of the English language as well as his car control skills. Detective Cherry is now under investigation by the Civilian Complaint Review Board following a complaint of 'discourtesy'. President of the Detectives Endowment Association said detective Cherry had been visiting his colleague detective Harry Hill who had recently had a heart attack while undergoing surgery and was very stressed. he said: ' the past five days have been emotionally draining for the members of the Joint Terrorism Task Force dealing with their fellow detective's health' despite what some people think, cops have feelings, too. 'During the three-minute video, detective Cherry shouts: ' I don't know where you're coming from, where you think you're appropriate in doing that; that's not the way it works. How long have you been in this country ?' the two passengers in the backseat of the car can be heard assuring the driver that he has done nothing wrong, with one of them saying; ' it's not your fault; this guy 's just a d***.' Moments later the cop returns with a ticket, this after he has pounded on the car and, according to one of the passengers, thrown things in the vehicle. As the driver agrees with everything the cop says, he manages to cut off the man, which sends him into another tirade as he screams;' I don't know what f****** planet you're on right now !' The officer also mocks the driver's accent throughout and his pronunciation of certain words. Passengers in the cab claimed that detective Cherry failed to indicate before attempting to park his Hyundai according to the New York Daily News, the NYPD has seen the video and said in a statement;'We are aware of the incident and video and it is under review with the department's Internal Affairs Bureau.' 'Our Uber driver, Humayun, was abused by a police officer today in New York, ' Sanjay Seth, who took the video, wrote on his Facebook page on Monday. 'The unending rage, door slamming, throwing items into the car, threatening arrest

without cause was bad enough-but the officer's remarks at the end really took it to another level. 'Uber said in a statement;' the behavior in the video is wrong and unacceptable and we appreciate the NYPD investigating the incident. We are in touch with our driver - partner who was subjected to this terrible experience and will continue to provide any support he needs"]

Referece Summary: ["The NYPD detective has been *accused* of shouting abuse at an Uber driver Cherry of the Joint Terrorism Task Force is now under investigation detective Cherry was on his way back from visiting a colleague in *hospital* the Uber driver 'honked' Det Cherry as he reversed into a parking space"]

Predicted Summary: ["President of the Detectives Endowment Association said detective Cherry had been visiting his colleague detective Harry Hill who had recently had a heart attack while undergoing surgery and was very stressed. 'During the three-minute video, detective Cherry shouts:'I don't know where you're coming from, where you think you're appropriate in doing that; that's not the way it works. How long have you been in this country?' the two passengers in the backseat of the car can be heard assuring the driver that he has done nothing wrong, with one of them saying;'It's not your fault; this guy's just a d***. 'Moments later the cop returns with a ticket, this after he has pounded on the car and, according to one of the passengers, thrown things in the vehicle"]

A.3 Example 3

Document: ["An ex-wife of a North Carolina man serving life in prison for the murder of his third wife has opened up about the abuse she faced when she was married to the man. Casey, also of North Carolina, was married to Michael Wilkie for four years and had a daughter with him before the couple divorced. He went on to marry his third wife, Shelby Wilkie. Michael Wilkie was found guilty of first-degree murder in January for the 2012 killing of Shelby Wilkie and is serving a life sentence without parole. Casey has opened up about the abuse she faced at the hands of

Michael Wilkie, who in January was found guilty of the 2012 murder of his third wife 'He said if I ever tried to take his daughter away from him that he would kill me, 'Casey told ABC's 20/20. Casey and Shelby Wilkie had met Michael Wilkie through an online dating site. Casey said they dated for a year-and-a-half before getting married. 'He was very friendly, very charming, easy to talk to, very soft spoken, and he had a good job and seemed to be pretty good,' Casey said. A couple months after marrying in 2004, Michael Wilkie began controlling aspects of Casey's life and alienating himself from Casey's daughter from a previous marriage, Casey said. 'If I planned to do something with one of my friends, he would manipulate the situation, and there would be something that came up that would interfere or get in the way,' she said. And then he began to get physically abusive and attacked her when she was pregnant with their daughter. He grabbed me by my throat and threw me around our bedroom and on the bed. My shoulder went through and made that hole in the sheetrock in the bedroom,' she said. Michael Wilkie (left) was sentenced to life in prison with no parole for killing his third wife, Shelby Wilkie (right). Shelby Wilkie and Michael Wilkie, both of North Carolina, had met on an online dating site but Casey never reported the incident and her friends and family were not aware of Michael Wilkie's abusive side because 'he was so good at masking'. 'It was like Jekyll and Hyde: two personalities and you didn't know which one you would get,' Casey said. 'You didn't know which one. You would meet when you got home.' She said she 'had thoughts' that Michael Wilkie would kill her, 'mainly because he told me he would kill me'. But Casey didn't leave Michael Wilkie for quite some time. 'I am the type of person that I will stay in a situation, whether it's a job or a marriage...longer than I should because I don't give up hope easily,' she told 20/20. 'And I am always thinking about, 'What could I do to make it better?' after an argument about pictures taken of their daughters together in 2006, Casey left Michael Wilkie. She took her older daughter but left the couple's three-year-old behind. The couple later divorced

in 2008. Casey eventually remarried and gained joint-custody of her and Michael Wilkie's child, and met her former husband's new wife, Shelby Wilkie, at a school event for their daughter. Shelby Wilkie was murdered in 2012 and her remains were found after a long search. Her and Michael Wilkie's child, Sydney (left), is in the process of being adopted by Shelby Wilkie's brother, Bill Sprowls, Jr, against Michael Wilkie's wishes she did not, however, warn Shelby Wilkie about the abuse she faced when she was married to Michael Wilkie. 'I had hoped that things had changed and that it was me and not, you know, him. And that way, hey, he could be happy. She could be happy, and it could be a nice household environment, ' Casey said about the couple. Just before Shelby Wilkie went missing the pair did have a short conversation.' She said, 'I just want to ask you some things about Michael Wilkie, is that ok?' and I said,' sure,'Casey recalled.' And I said, 'Shelby Wilkie if there is anybody that knows what you are going through, it's me. 'Casey told Shelby Wilkie she had to go shortly after and asked her to call her back. she never got a callback, and instead saw Michael Wilkie pleading for Shelby Wilkie to come home. Casey (right) married Michael Wilkie in 2004. she said a few months after the wedding be become controlling and eventually physically abusive. At times, she said she feared he would kill her at first, Casey thought that Shelby Wilkie had run from her husband, but Michael Wilkie was then arrested, charged and found guilty of his wife 's murder. Shelby Wilkie had filed two domestic violence charges against her husband before her death, but both were voluntarily dismissed, according to WSOC. blood and her ashes, along with a charred bracelet her mother had given her, were later found and Michael Wilkie was arrested, according to ABC. And it wasn't until his arrest that Casey finally felt safe, she said. 'It's made me grow as a person,' she said.' And it has made me stronger as a human being"]

Reference Summary: ["Michael Wilkie was found guilty in january of first - degree murder in january for the 2012 killing of his third wife , Shelby Wilkie his second wife

, Casey , has opened up about the abuse she faced before divorcing Casey said that he controlled aspects of her life and was physically abusive , particularly when she was pregnant she even said she feared that Michael Wilkie would kill her she said she never warned Shelby Wilkie , but told her she was there if she needed someone to talk to shortly before she *disappeared*"]

Predicted Summary: ["An ex-wife of a North Carolina man serving life in prison for the murder of his third wife has opened up about the abuse she faced when she was married to the man. 'He was very friendly, very charming, easy to talk to, very soft spoken, and he had a good job and seemed to be pretty good,' Casey said. 'You would meet when you got home.' she said, 'I just want to ask you some things about Michael Wilkie, is that ok?' and I said, 'sure,' Casey recalled"]

VITA

Sai Amrit Bulusu was born and raised in Hyderabad, India. Before attending the University of North Carolina at Charlotte, he attended GITAM University, Hyderabad, India, where he earned a Bachelor of Technology in Computer Science, in 2016.

While at the University of North Carolina at Charlotte, Amrit worked on many interesting Machine Learning, Natural Language Processing and Data Science related projects. He received Master of Science in Computer Science and a Graduate Certificate in Advanced Databases and Knowledge Discovery from the University of North Carolina at Charlotte in May 2018.