Headline Generation for News Articles

Abhijeet Dubey (16305R006)
Abhishek Bagade (163059005)
Nithin S (16305R007)
Saket Bhojane (163050063)

Computer Science & Engineering
Indian Institute of Technology, Bombay
Mumbai, 400076
2018

Acknowledgements

We would like to thank our mentor **Ms. Vertika Srivastava** for always helping us and pointing us in the right direction whenever we were stuck. We would also like to thank **Prof. Sunita Sarawagi** for teaching the Advanced Machine Learning course and giving us the opportunity to pursue this project.

We would also like to thank **Prof. S. Sudarshan** for providing us with the dataset for running our experiments.

Abstract

Newspaper plays a significant role in our day to day life. In a news article, readers are attracted towards headline. Headline creation is very important while preparing news. It is a tedious work for journalists. Headline needs maximum text content in short length also it should preserve grammar. The proposed system is an automatic headline generation for news article. There are several methods to generate a headline for news. During the course of this project we experimented with neural models as well as SMT based models.

Neural sequence-to-sequence models have provided a viable new approach for abstractive text summarization (meaning they are not restricted to simply selecting and rearranging passages from the original text). However, these models have two shortcomings: they are liable to reproduce factual details inaccurately, and they tend to repeat themselves. We experimented with pointer generator networks which alleviate this problem and generate significantly better results.

Contents

List of Figures

Chapter 1

Introduction

Text summarization is the problem of creating a short, accurate, and fluent summary of a longer text document. Summarization is the task of condensing a piece of text to a shorter version that contains the main information from the original. There are two broad approaches to summarization: extractive and abstractive.

- **Extractive Summarization:** Extractive methods assemble summaries exclusively from passages (usually whole sentences) taken directly from the source text.
- **Abstractive Summarization:** Abstractive methods may generate novel words and phrases not featured in the source text – as a human-written abstract usually does.

Automatic text summarization methods are greatly needed to address the ever-growing amount of text data available online to both better help discover relevant information and to consume relevant information faster.

We aim to use statistical and deep learning based models for text summarization for generating headlines for news articles.

1.1 Roadmap

The structure of the rest of the report is as follows. Chapter 2 describes the related work in the field of text summarization. Chapter 3 explains various approaches and models we tried for this task. Chapter 4 presents the results obtained from various approaches and error analysis. Chapter 5 is the conclusion.

Chapter 2

Related Work

Neural abstractive summarization. [Rush et al., 2015] were the first to apply modern neural networks to abstractive text summarization, achieving state-of-the-art performance on DUC-2004[1] and Gigaword[2], two sentence-level summarization datasets. Their approach, which is centered on the attention mechanism, has been augmented with recurrent decoders.

However, large-scale datasets for summarization of longer text are relatively rare. [Nallapati et al., 2016b], which uses hierarchical RNNs to select sentences, and found that it significantly outperformed other approaches with respect to the ROUGE metric. Prior to modern neural methods, abstractive summarization received less attention than extractive summarization.

Internet giants like Google, Facebook, IBM etc., are also working on approaches for text summarization. Some of the state of the art models are:

- The Facebook AI Research model that uses the Encoder-Decoder model with a convolutional neural network encoder.
- The IBM Watson model that uses the Encoder-Decoder model with pointing and hierarchical attention.
- The Stanford / Google model that uses the Encoder-Decoder model with pointing and coverage.

[1] https://duc.nist.gov/data.html
[2] https://catalog.ldc.upenn.edu/ldc2003t05

Chapter 3

Approaches Tried

3.1 Headline Generation using Statistical Machine Translation

We modelled the task of generating headlines from news articles as a monolingual machine translation task. We used Moses [Koehn et al., 2007] for Statistical Machine Translation (SMT). Moses is a state of the art tool for statistical machine translation which provides phrase-based translation of short text chunks and handles words with multiple factored representations to enable the integration of linguistic and other information (e.g., surface form, lemma and morphology, part-of-speech, word class). It also provides support for large language models. Overview of SMT framework is illustrated in Figure 3.1

Figure 3.1: Phrase Based Statistical Machine Translation

3.2 The Facebook AI Research model - NAMAS

This approach was described by [Rush et al., 2015]. from Facebook AI Research (FAIR) in their 2015 paper "A Neural Attention Model for Abstractive Sentence Summarization"(NAMAS).

The model was developed for sentence summarization, specifically, to take x as input of length M and output a shortened sentence y of length N <M. The words in the summary also come from the same vocabulary.
The approach follows the general approach used for neural machine translation with an encoder and a decoder. Three different decoders are explored:

- **Bag-of-Words Encoder**. The input sentence is encoded using a bag-of-words model, discarding word order information.
- **Convolutional Encoder**. A word embedding representation is used followed by time-delay convolutional layers across words and pooling layers.
- **Attention-Based Encoder**. A word embedding representation is used with a simple attention mechanism over a context vector, providing a type of soft alignment between input sentence and output summary.

A network diagram for NAMAS is shown in 3.2.

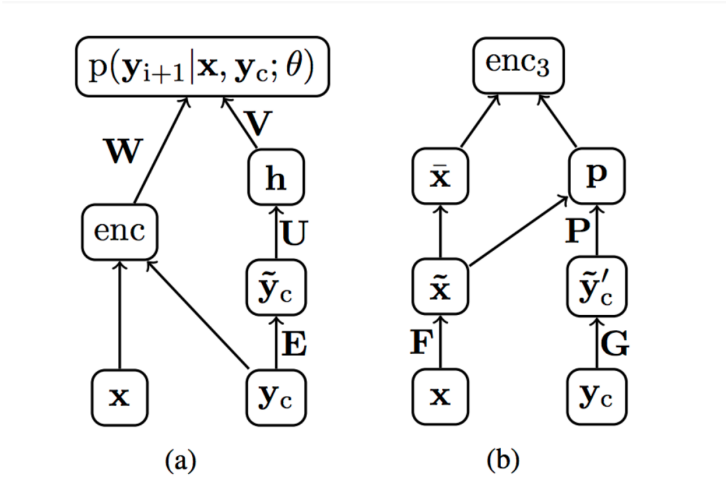

Figure 3.2: Network Diagram of Encoder and Decoder Elements

3.3 Pointer generator network

Pointer generator network[See et al., 2017] handles the following two problems, let's look at them:

- **Problem 1: Sometimes the summary could produce incorrect factual results (e.g. Germany beat Argentina 3-2)**:

  This is because 2-0 is out of vocabulary term. Author speculates that, this probably is caused because of poor word embeddings, which in turn might have been caused because of less frequent appearance of word in the training data[1]. Also, note that it might so happen that the network replaces proper nouns with another
  eg. Abhijeet $\longrightarrow$ Abhishek

[1]http://www.abigailsee.com/2017/04/16/taming-rnns-for-better-summarization.html

Solution: As a solution to this problem, pointer generator network, instead of doing pure abstraction, goes by a hybrid model i.e it has the capacity to copy words directly from the source text (like pointing)

- **Problem 2: The summaries sometimes repeat themselves (e.g. Germany beat Germany beat Germany beat. . .)**:

  According to author's speculation, this may be caused by the decoder's over-reliance on the decoder input (i.e. previous summary word), rather than storing longer-term information in the decoder state.[1]

  Solution: this problem is tackled by using a concept called as coverage, i.e to keep track of what has been covered so far using attention distribution and then penalizing the network for covering the same part again.

To sum up, out of vocabulary words or words with poor embeddings (less frequent words) are handled by the "extractive" nature of the network and the problem of repetition of abstractive nature is handled by coverage phenomenon. This is apparent in the results we got in simple encoder decoder model and pointer generator network discussed in later chapter.

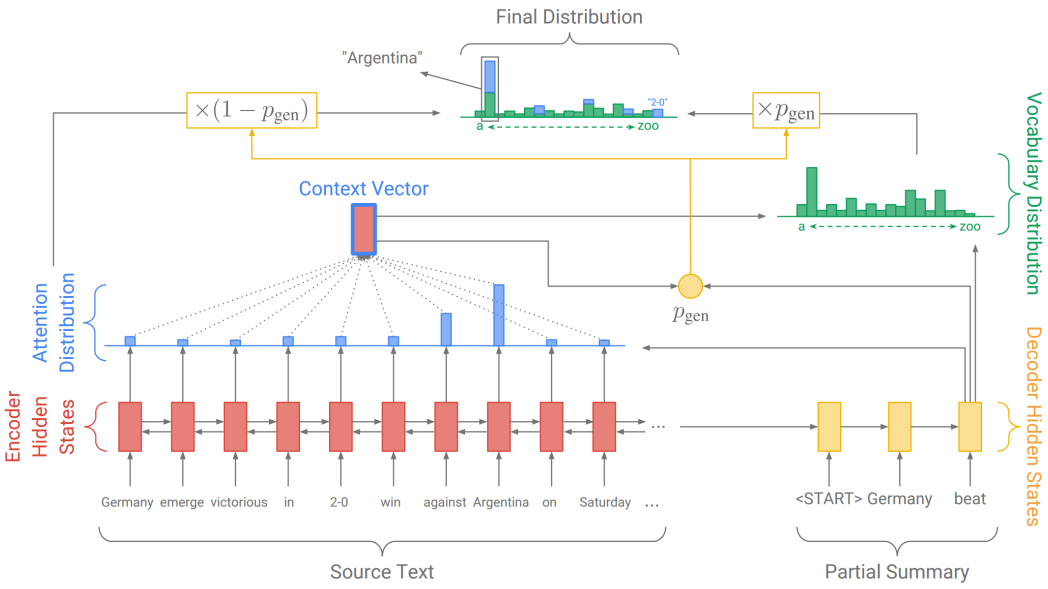

Figure 3.3: Hybrid sequence to sequence model[1]

3.4 IBM Model

The approach is based on the encoder-decoder recurrent neural network with attention, developed for machine translation. A word embedding for input words is used, in addition to an embedding for tagged parts of speech and discretized TF and IDF features. This richer input representation was designed to give the model better performance on identifying key concepts and entities in the source text.

Finally, the model is hierarchical in that the attention mechanism operates both at the word-level and at the sentence level on the encoded input data. This model is illustrated

in Figure 3.4[2].

Unfortunately, we could not implement and test this model due to shortage of time.

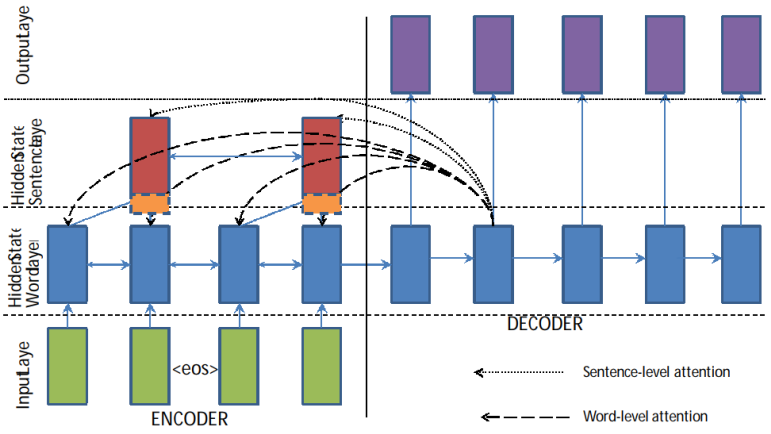

Figure 3.4: Hierarchical encoder with hierarchical attention.

3.5 Baseline Model: Encoder Decoder Network

[Sutskever et al., 2014] is used as a reference for implementing this model. Our model is a character level encoder decoder network for generating headlines one character at a time. We used LSTM units for both encoder and decoder. We implemented this model in Keras[3] using Theano[4] as backend. This model is illustrated in Figure 3.5[5]

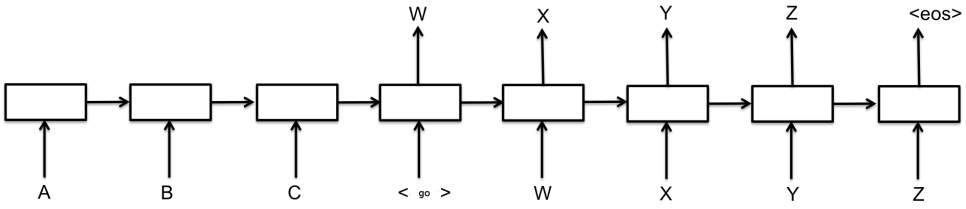

Figure 3.5: Character Level Sequence to Sequence Model

[2]This figure is taken from [Nallapati et al., 2016b]
[3]https://keras.io/
[4]http://deeplearning.net/software/theano/
[5]This figure is taken from [Sutskever et al., 2014]

Chapter 4

Experiments

4.1 Dataset

Our dataset consists of 12 lac Time of India (TOI) articles from various domains like city, India, life-style, sports etc. However due to time constraints we decided to train our models on articles from entertainment domain.

4.2 Pre Processing

- Categorized the mixed raw data into 10 broad categories which had > 50K news articles and discarded rest assuming it won't be sufficient for deep learning
- Separated the article caption from the body
- Removed all the non-ascii characters from the articles
- Since, our goal is to come up with headline for news article, we just truncated the news articles to first 5 sentences, assuming a brief headline is extractable within that many lines and reduce the data so that the model learns quicker

4.3 Experimental Platform

- We trained pointer generator network on Nvidia GTX 1080 GPU. This model took around 30 hours for training.
- We trained our baseline model which is character level sequence to sequence model on Nvidia Titan X GPU. We trained this model for 20 epochs which took around 10 hours.
- Moses is trained on Krishna Server[1].

[1] http://www.cfilt.iitb.ac.in/

4.4 Code Description

Entire code is available at https://github.com/saketbhojane/AML-Project

Preprocessing of raw data This is done using **preprocess.sh** bash script which calls **categorise_news_article.py** (50 lines of code). It does the job of cutting off artcicles from unwanted domains and the copies the articles in the respective directory. The bash script deletes all the folders/domain articles which have less than 50K articles.

data_reframing_categorization.py script segregates the news articles into train, validation and test data in ratio 8:1:1. (50 lines)

make_datafiles.py scipt in pointer generator model does customized version of preprocessing for our (TOI) data like [2] (250 lines of code). This script tokenizes all the news articles using stanford's NLP toolkit [3], converts data in to binary file and then chunks it such that each chunk contains 1K article + headline

4.5 Results

The results we got for different models were as follows

1. Statistical MT model (Using MOSES toolkit) provided bad results which were of no use considering the translation produced a large no. of UNK tokens. This caused the BLEU score metric to fail producing a BLEU score of less than 5.
2. Sequence to Sequence NMT model implemented using Keras provided a BLEU score of 7.2 which is better than SMT based approach but still far from state of the art.
3. Facebook's NAMAS produced a BLEU score of 21.8
4. The best results were produced by Pointer generator networks which produced a BLEU score of 31.3

[2] https://github.com/abisee/cnn-dailymail/blob/master/make_datafiles.py
[3] https://stanfordnlp.github.io/CoreNLP/

Chapter 5

Conclusion

5.1 Summary

During the course of this project, we investigated several text summarization methods. We hypothesized that text summarization is similar to monolingual machine translation but the difference in lengths between source and target proved too complicated for Statistical Machine Translation Systems. Even the neural models that we investigated, like Sequence to Sequence models produced unsatisfactory results. We theorize that this might be due to the large vocabulary. Pointer generator networks give better results since they employ best of both techniques: extractive and abstractive methods.

5.2 Effort

The first challenge that we faced was obtaining the dataset, cleaning it and dividing it into relevant categories. Running the Statistical Machine Translation model MOSES was relatively straightforward as it was already set up in one of the CFILT Lab servers.We were able to run Facebook's model, NAMAS after minor tinkering. The lions share of time and effort went into running the Pointer generator network and implementing the Sequence to Sequence Learning Model. This was easily the most challenging part of the entire project.

Every member of the team contributed equally to the project.

Bibliography

[Koehn et al., 2007] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL*. The Association for Computer Linguistics.

[Nallapati et al., 2016a] Nallapati, R., Xiang, B., and Zhou, B. (2016a). Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.

[Nallapati et al., 2016b] Nallapati, R., Zhai, F., and Zhou, B. (2016b). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *CoRR*, abs/1611.04230.

[Rush et al., 2015] Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685.

[See et al., 2017] See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.