Attention Based Models for Summarization

Shah Manan Jayant **Koushik Sen** **Upasana Doley** **Rahul Dev**

Abstract

Neural Sequence to Sequence Models are most widely used models for Language Modelling task in Natural Language Processing. Text Summarization is one such application of Language Modelling. This project involves building neural sequence to sequence models for the purpose of abstractive text summarization. Attention based networks and Attention based pointer generator networks are the two models build for summarization in this work.The models are trained on CNN Daily News Dataset and achieves ROUGE score of 33 points.

1. Introduction

Automatic summarization techniques have gained a lot of importance which can summarize a document in short and focus on the important points in the document. Summarization can be classified into extractive and abstractive. An abstractive approach on the other hand, tries to use novel words for summarizing the text. This appears to be more significant as it succeeds to create human like summaries. But these systems have problems of inaccurately reproducing factual details, an inability to deal with out-of-vocabulary words and repeating themselves.

See et al., 2017 has addressed these three issues for multi sentence summaries by creating a hybrid pointer-generator network which copies words from the source text via pointing which deals with the factual details and out-of-vocabulary problems but also has the ability to generate new words. They also proposed a coverage mechanism by introducing a coverage vector which represents the degree of coverage of words from the source text.

In this project, An Attention based Sequence to Sequence Model and Attention Based Pointer Generator Model (See et al., 2017)are trained for summarization The current implementation of model uses different variant of Recurrent Neural Network (RNN) i.e Gated Recurrent Unit (GRU's) and use of pretrained GLoVe vectors in the model.

2. Model

In this section we describe the two models 1.Attention based sequence to sequence models and 2.Pointer Generator Models.

2.1. Attention Based Model

1. Attention Model (Nallapati et al., 2016) consist of Recurrent Neural Network as encoder to which word tokens from input sentences are fed one by one. This gives a series of encoder hidden states. The decoder consists of a unidirectional RNN unit with the same hidden state size as that of the encoder. Attention distribution is calculated as stated in Bahdanau et al., 2014 based on encoder hidden states h_t and decoder output state s_t as follows:

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \quad (1)$$

$$a^t = softmax(e_i^t) \quad (2)$$

Here v,W_h,W_s,b_{attn} are learnable parameters. A weighted sum of encoder hidden state is calculated based on attention weights is called context vector.Basically,this implements a mechanism of attention in the model.The model can decide the parts of the sentences to pay attention.

$$h_t^* = \sum_i a^t h^t \quad (3)$$

After this a probability distribution over all words in the vocabulary is generated by concatenating the context word and the decoder state and it is fed through two linear layers to predict the decoder words as follows:

$$P_{vocab} = softmax(V^{'}(V[s_t, h_t^*] + b) + b^{'}) \quad (4)$$

Here V,$V^{'}$,b,$b^{'}$ are learnable parameters. This is essentially the final distribution for our model to predict words w as follows:

$$P_w = P_{vocab}(w) \quad (5)$$

Negative log-likelihood of the target word w_t^* during training stage is used as follows:

$$loss_t = -logP(w_t^*) \quad (6)$$

The overall loss for the complete input sequence is:

$$Totalloss = \frac{1}{T}\sum_{t=0}^{T} loss_t \quad (7)$$

2. Three different types of attention mechanism are implemented to calculate the attention distribution which inturn lead to final vocabulary distribution:
$e \in \mathrm{R}^N$, $h_1, h_2, ..., h_N \in \mathrm{R}^{d1}$ and $s \in \mathrm{R}^{d2}$
   (a) Basic Dot-Product Attention
   $\mathrm{e}_i^t = s_t^T * h_i \in \mathrm{R}$
   (b) Multiplicative Attention
   $\mathrm{e}_i^t = s_t^T * W * h_i \in \mathrm{R}$
   (c) Additive Attention
   $\mathrm{e}_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn})$
   $W_h \in \mathrm{R}^{d3*d1}, W_s \in \mathrm{R}^{d3*d2}, v \in \mathrm{R}^{d3}$

2.2. Pointer-Generator Model

In the previously discussed attention based model, it cannot handle OOV words as the final distribution consists of only the vocabulary words and and the OOV words present in the source text.In Pointer Generator networks done bySee et al., 2017, the model can point(copies OOV words directly) from the original input text in case the words are not present in the vocabulary as well as generate new words form the fixed vocabulary.Here the generation probability $p_{gen} \in [0, 1]$ and the copying probability $1 - p_{gen} \in [0, 1]$ is controlled in each time step t as follows:

$$p_{gen} = \sigma(w_{h*}^T * h_t^* + w_s^T * s_t + w_x^T * x_t + b_{ptr}) \quad (8)$$

Here the vectors w_{h*},w_s,w_x,b_{ptr} are the learning parameters.Now our extended vocabulary is a union of the words from the fixed dictionary and the source words present in the text document.It ensures that the model is able to generate new words as well as copy words from the source text as required.The final distribution is calculated over the extended vocabulary as follows:

$$P_w = p_{gen} * P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \quad (9)$$

In case the word w is an OOV word the first component becomes zero whereas if the word is not present in the source text the second component becomes zero. So the primary advantage of this model is that it can also generate OOV words unlike the attanetion based model.

3. Implementation

3.1. Attention based Models

3.1.1. Data Pre-processing

Dataset used for training the models is of CNN Daily News. It consist of 4 million small articles of 40-70 words and 10-20 word summary of each article. Out of 4 million,1 million examples were used for training the models.The first step was data cleaning part. From each text article and summary, the special characters were removed. This includes digits as well. The articles are tokenized and 40,000 most common words are taken to form a vocabulary for the model.4 extra tokens for start sentence, end sentence, unknown word and padding were also added. The model used by – train the word vectors form scratch. Here pretrained GLoVe vectors(Pennington et al., 2014) of 50 dimensions are used to get word embeddings for these 40,000 words and they are not trained by the model. Embedding Matrix of 40,000 X 50 elements is formed which serves as part of embedding layer to the model.The article length and summary length are restricted to 50 words and 15 words respectively. The articles whose length is greater than 50 words are truncated and articles whose length is less than 50 words are padded with pad token. In same way, the summary are made to same length of 15 words. This forms the training set of 1 million articles of 50 words each 15 words summary for each example.

3.1.2. Model Parameters

The model parameters used are as follows:

Table 1. Model Parameters for Attention Based Networks

Batch Size	512
Training Set Examples	1 Million
Input Sequence Length	50
Output Sequence Length	15
Encoder Hidden State Dimension	200
Vocabulary Size	40000
Epochs trained	15
Word Vector Dimension	50

3.1.3. Attention Mechanism

Gated Recurrent Unit (GRU) are used as building blocks for encoder decoder model for this model.Models with all three variants of attention mentioned earlier are trained.

3.2. Pointer Generator Networks

3.2.1. Data Pre-processing

Data preprocessing for pointer generator network model involves some extra steps along with preprocessing steps mentioned earlier. Two tokenized representation for each training example is maintained. In first representation, word not found in vocabulary is replaced by id of unknown token.In second representation, 1st word not found in vocabulary is replaced vocabulary size plus 1, 2nd word by vocabulary size plus 2 and so on. A list of Out Of Vocabulary (OOV) words is maintained for each example which helps

to retrieve the word corresponding to ids.The maximum out of vocabulary words of in an example is 30. So extended vocabulary size is 40030 for each batch of training.

3.2.2. Model Parameters

The model parameters used are as follows:

Table 2. Model Parameters for Pointer Generator Networks

Batch Size	512
Training Set Examples	1 Million
Input Sequence Length	50
Output Sequence Length	15
Encoder Hidden State Dimension	200
Vocabulary Size	40000
Extended Vocabulary Size	40030
Epochs trained	15
Word Vector Dimension	50

3.2.3. Pointer Generator Mechanism

Gated Recurrent Unit (GRU) are used as building blocks for encoder decoder model for this model.The decoder hidden state, context vector and encoder input is used to get probability of generation (pgen) from single layer neural network.The attention distribution of 50 words in encoder input is mapped to probability of extended vocabulary of 40030 words. The vocabulary distribution of 40000 words from decoder unit is also projected to distribution of 40030 words.These two distribution are added to get final vocabulary distribution.

4. Experiments

Two models namely Pointer Generator Models and Attention models were trained on 1 Million examples for Summarization. Two test set were made 1) Test set 1: consisting of 10 examples with less number of OOV words 2) Test set 2: consisting of 9 examples with more number of OOV words.

5. Results

5.1. Performance of Different Models on Test set 1

Table 3 shows the comparison of differnt models on test set 1. Dot product attention model and pointer generator model is found to perform well on test set 1. These two models were further considered

5.2. Performance on Different Models on Test set 2

Example Case:

- Article:
  china's economy for the fourth quarter of this year will see a moderate growth over the third quarter , a survey by the State economic and trade commission shows
- Reference Summary:
  china's economy predicted to be stronger
- Summary from Dot Product Attention Model:
  china 's economic growth to profit up percent
- Summary from Multiplicative Attention Model:
  china's economy to cut growth in third quarter
- Summary from Bahdanu Attention Model:
  china's growth slows one-shot of dollar growth
- Summary from Pointer Generator Model:
  china's economy grows to cut of gdp growth

Table 3. Model evaluation with Test set 1

	F1-Score Rouge 1	Rouge 2	Rouge L
Dot Product Attention Model	31.171	4.011	28.271
Multiplicative Attention Model	25.47	2.666	22.119
Bahdanu Attention Model	28.41	1.33	25.73
Pointer Generator Model with Bahdanu Attention	33.458	5.523	29.813
F1 score See et al., 2017	36.441	5.66	33.42

Table 4. Model evaluation with Test set 2

	F1-Score Rouge 1	Rouge 2	Rouge L
Dot Product Attention Model	23.542	3.931	22.521
Pointer Generator Model with Bahdanu Attention	27.457	8.635	25.297

6. Conclusion

Different types of Attention Models are evaluated for summarization in this project.Pointer Generator Model with Bahdanu Attention is found to perform well on both test set considered. it improves rouge scores on text with only vocabulary words and also performs better than other models on examples with Out of Vocabulary Words.

References

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

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.