Project Report CS-323 : Abstractive Summarization

Group Members

1.Kanchumarthi H Pravallika 200101049
2.Ashwitha Banoth - 200101021
3.Pruthvi Raj G N - 200101088
4.Gandrathi Srija - 200101030

Abstract

This document serves as a report for the course project in CS-323 - Natural Language Processing. The
project was an adaptation of the Fast Abstractive Summarization model (Chen and Bansal, 2018) using select sentence rewriting. This report will highlight the challenges and insights in the problem statement, and our methods to improve the SOTA. We drew inspiration from how humans summarize documents, sometimes combining and rewriting multiple sentences in the summary. Moreover, we propose that the current evaluation metrics (ROUGE, METEOR) are an adequate measure of performance for the summarization task.

1 Introduction

Our main motivation for choosing this particular task was multifold. Firstly, text summarisation is a hard problem in NLP, due to the absence of standardized evaluation metrics, lack of *ground truth* summaries, and inability to manually annotate most training data to scale.

On top of this, abstractive text summarization takes it a step further, adding the complexity of sentence generation along with feature selection in documents. The summary produced not only has to be an accurate representation of the document, but also has to be succinct and grammatically and linguistically correct. The problem transforms to *"How would a human summarize this?"*

To further add to this task, most of the current evaluation metrics (BLEU, ROUGE, METEOR) just compare the generated summary with a *gold* summary, making the assumption that this is the only acceptable summarization of this document, and anything different is considered bad

With this motivation, and an interest in abstractive text summarisation, we had decided to choose the following as our project topic. We experimented with various modifications on the original model (Chen and Bansal, 2018) and looked at some of the failings of this model to get some insight into our path for progress

2 Methods

The task is split into a few sub tasks, namely selecting the sub-sentences to form the highlights, and rewrite these sentences to create a summary. So, we assume there exists an extractor function capable of selecting the apt sentences to form the summary. Furthermore, we assume there is an abstractor function, which rewrites these sentences into a desirable form. In a way, the extractor chooses the salient sentences from the document for the abstractor function to rewrite.

1. **Extractor**
   The extractor agent selects the salient sentences from the document to pass onto the abstractor agent. It uses a hierarchial neural model (Embeddings, Convolutions and RNNs) to learn the sentence representations of the document, and a selection network to select the best sentence representations from among them.

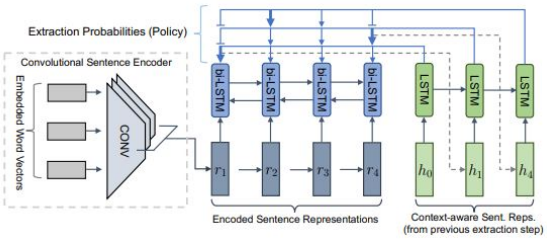

Figure 1: The extractor agent: the CNN encoder computes representation rj for each sentence. The RNN encoder (blue) computes context-aware representation hj and then the RNN decoder (green) selects sentence jt at time step t. With jt selected, hjt will be fed at time t+1. (Chen and Bansal, 2018)

In this agent, the sentence embeddings are calculated using a temporal convolution model (Kim, 2014) to compute the representations for each sentence in the document. This is then passed through an BiLSTM-RNN is applied on the sentence embedding to get a strong sentence representation which takes into account past and future sentences in the same document.

For the sentence selection, we feed this context aware sentence embeddings to another LSTM and train a pointer network (Vinyals et al., 2015) to extract sentences sequentially. These values are passed to the abstractor network.

2. **Abstractor**
   The abstractor network compresses and paraphrases the extracted document sentences into summary sentences. It is a sequence to sequence model (encoder-alignment-decoder) with the bilinear multiplicative attention function for the context vector. On top of this, it incorporates the copy mechanism (See et al., 2017) to predict OOVs directly from the input. An OOV word is replaced by the document word with the highest attention score.

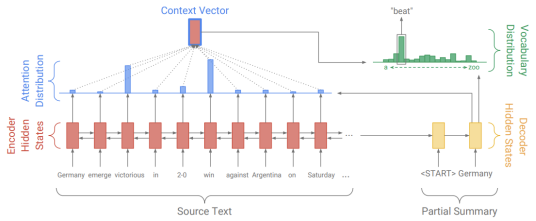

Figure 2: Baseline sequence-to-sequence model with attention. (See et al., 2017)

3. **RL Training**
   Since sentence selection is a non-differentiable hard problem, standard policy gradient methods to bridge the gap in back propagation. However, the extractor and abstractor agent, being randomly initialized, would interfere with each others training.
   Hence, each sub-model is trained separately (using Maximum Likelihood approach), and then reinforcement learning based approach is used to fit the full model end-to-end.

The extractor is trained using a simple metric of selecting the max overlapping sentence in the document to each sentence in the summary. Using this, we we can train the extractor to pick out the most relevant sentences from the summary. Similarly, the abstractor is trained by taking the pair of extracted-summary sentences to learn a generator model of summary sentences.

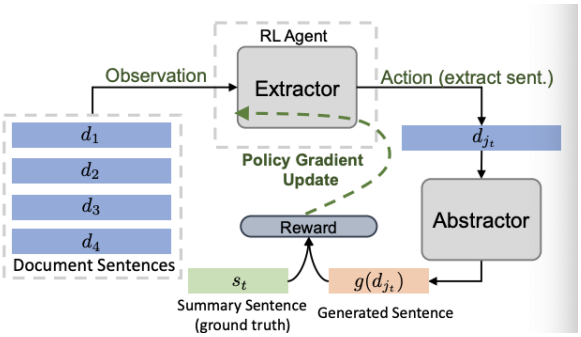

Figure 3: Reinforcement Training (one step) and it's interaction with the abstractor (Chen and Bansal, 2018)

3) Experiment 1. Dataset and Metrics

All experimentation was done on the non-anonymised version of the CNN/Dailymail dataset (modified for summarization) (Nallapati et al., 2016)

Dataset Split	
Training	287,227 docs
Validation	13,368 docs
Testing	11,490 docs
Data Statistics	
Words/document	766
Sentences/document	29.74
Words/summary	766
Sentences/summary	29.74

Table 1: Details about the CNN/Dailymail dataset used

For the evaluation, as per previous work (Nallapati et al., 2016) (See et al., 2017), we use ROUGE-1, ROUGE-2 & ROUGE-L (Lin, 2004) F-1 scores. Also evaluated on METEOR (Denkowski and Lavie, 2014) for some additional analysis. These metrics don't accurately capture the quality of the produced summary, since it assumes that the gold summary is absolute, and any deviation is considered a fault. This will result in many equally good summaries getting discarded.

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
(See et al., 2017) (w/o coverage)	36.44	15.66	33.42	16.65
(See et al., 2017)	39.53	17.28	36.38	18.72
(Fan et al., 2017) (controlled)	39.75	17.29	36.54	-
(Chen and Bansal, 2018) (without rerank)	40.04	17.61	37.59	21
(Chen and Bansal, 2018) (with rerank)	40.88	17.8	38.54	**20.38**
(Chen and Bansal, 2018) (Replicated by us)	**41.18**	**18.17**	**38.77**	20.53
Our model (abstractive++)	*38.26*	*17.2*	*36.82*	*19.22*

Table 2: Results on the CNN/DailyMail dataset

4 Our Changes

We noticed, while looking at the predicted summaries, a major trend across poor performers, was the gold summary was a product of the concatenation and rewriting of multiple document sentences. Since our model emulates a pseudo-fixed summary size (due to the end of summary tag), it is unable to accurately capture the information of multiple sentences into a single summary sentence, as done in these gold summaries.
To tackle this, we increased the feed-forward into the abstractor network, by concatenating the top-k sentences which overlapped with the summary sentence, rather than just the top-1. This resulted in a larger sentence for the abstractor, which then learned to rewrite a sentence as a high level summary of multiple sentences.

Apart from this, there were a few other changes that we unsuccessfully tried out:

- We retrained the word embeddings, but used Google News word2vec embeddings (since dataset consists of news as well) but got a reduction in performance (40.9% ROUGE-1)
- Increasing the number of sentences concatenated (k¿4) led to a further drop in accuracy and significantly increased the RL training time
- We experimented with using a GANs setting for comparing summaries. The intuition behind it was we never use the document itself while evaluating a generated summary. We hypothesized a GANs setting where we have a generative network creating a summary given a document, and then a discriminative network which evaluates the matching of a document and a summary. However, due to logistical constraints (training time/ resources) we were unable to see this to fruition
- Earlier proposal, Abstractive summarization of long documents. Since the documents in this dataset (Cohan et al., 2018) were roughly 6-7 times as large as CNN/Dailymail, we were again unable to work with these due to logistical constraints.

4 Conclusion & future scope

During the course of this project, we realized the need for developing an accurate evaluation metric for summaries which take into account the document itself (rather than some gold summary). This will lead to a evaluator which is able to understand the underlying structure of the document and summary, comparing these two, rather than just a word-by-word comparison as is the state of the art. Our proposed model doesn't do well in a quantitative sense (ROUGE, METEOR scores), but we believe by Human Evaluation we can showcase the improvements in our model, and further go on to create a more abstractive version of the (Chen and Bansal, 2018) model we worked with.