Abstractive Summarization and Extrinsic Evaluation via Q/A

Akash Rastogi, Bhavika Reddy Jalli, Daniel D'souza, Vidya Mansur
University of Michigan, Ann Arbor

December 12, 2017

Abstract

Text summarization is a process to create a representative summary or abstract of the entire document, by finding the most informative portions of the article. There are two approaches for automatic summarization: extractive summarization and abstractive summarization. The current techniques to evaluate an summarizer are BLEU and Rogue-n scores. These metrics are based on the overlap between the predicted summaries and the summaries provided by human (generally mechanical turks or news headlines). These metrics can be good system to evaluate the extractive summaries because they extract word features from the input text. Hence, we expect there to be a huge overlap between the predicted and human-provided summaries. For abstractive summarizer which aims to understand the text and provide a summary, it is not necessary for them to have the same words as there are in the human-provided summaries. But due to non-availability of a better metric system, we are still using BLEU and Rogue-n scores to evaluate abstractive summaries. Our understanding, is that if a summary can answer the questions based on the text then it is a good summary. Hence, we propose to use the Question/Answering system as an evaluation metric to evaluate the summaries.

In this project, we model an abstractive text summarizer to convert a piece of text into an abstractive summary inspired by (Lopyrev, 2015) which is based on RNN attention based model. Our current model is able to answer nearly 20 percent of the questions related to the text.

1 Introduction

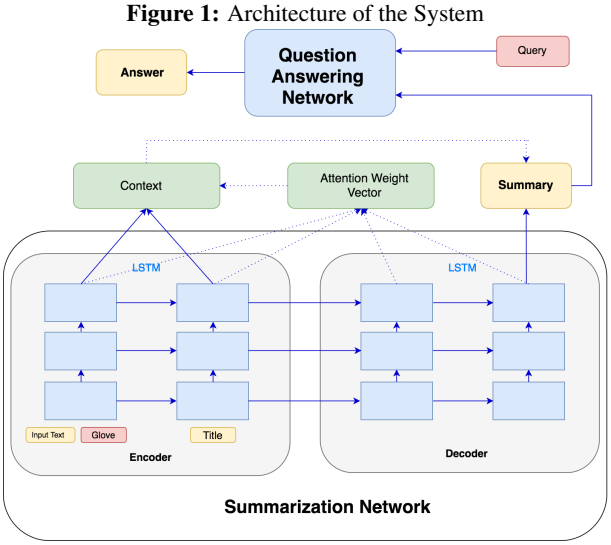
Every day, people rely on a variety of resources to consume information like news articles, blog posts, search results. Hence, there is a huge demand for a system which can compress large amount of information into compressed form. To summarize well, the system should be able to understand the document, distill important information and produce this information in human readable format.

Reading comprehensions can also serve as a useful test for the summarization networks. If the summarizer is able to extract useful information from the text then it should be able to answer the questions related to the text. With this in mind, we want to combine an abstractive summarizer and a Q/A network to improve the summarizer.

In this project we aim to create an abstractive summarizer inspired by (Lopyrev, 2015) which converts a piece of text into a summary by abstracting concepts from the original article. Simultaneously, we also want to create a Question and Answering network. This system will be based on (Minjoon Seo, 2017). Once we have the two systems in place, we would like to use the Q/A network to use our generated summaries as the input text for Q/A system and train the two systems simultaneously. We believe that if we can back propagate the loss through the two systems we can improve the accuracy of the two systems. After having trained the two modules Abstractive Summarization and the Question Answering module, we are planning to link the two models and train them together. We will pass the input to the abstractive summarizer network and create a summary. This summary will be passed to the Q/A network to generate the answer. We can then back prop this loss through the Q/A network and the summarization network. We believe this transfer learning can help us to improve the summarization network and the Q/A network. A visualization of how we are planning to attach the two network is shown.

1

Figure 1: Architecture of the System



2 Related Work

2.1 Abstractive Summarization

Summarization is largely considered an unsolved problem in natural language processing. The goal of a summarization task is to produce a condensed version of the input text while capturing the crux of the article. There are two general approaches to automatic summarization: extractive and abstractive. Extractive summarizer (Verma and Nidhi, 2017) extracts keywords from the input text to create the summary. These summarizer can help us to tag or flag the articles in a better manner. Abstractive summarizers tries to understand and create an embedding for the input text and then represent it in a human readable format.

While research on the former has been the focus for many years, with the advent of word2Vec (Tomas Mikolov, 2013) and similar word representations, abstractive techniques have now taken the spotlight. In this method of automatic summarization, articles are shortened not just by selecting informative sections of the original text, but also paraphrasing the article. (Lopyrev, 2015) were successful in generating informative headlines from news articles using encoder-decoder Recurrent Neural Networks. Salesforce has recently achieved impressive results using models with intra-attention and reinforcement learning (Romain Paulus, 2017). We aim to construct a model with an architecture similar to (Lopyrev, 2015) but inspired by (Romain Paulus, 2017), optimized with a QA system.

2.2 Q/A system

Previous works in end-to-end machine comprehension use attention mechanisms in three distinct ways. The first group (largely inspired by (Dzmitry Bahdanau and Bengio, 2015)) uses a dynamic attention mechanism, in which the attention weights are updated dynamically given the query and the context as well as the previous attention. The second group computes the attention weights once, which are then fed into an output layer for final prediction (e.g., (Rudolf Kadlec and Kleindienst., 2016)). The third group considered as variants of Memory Network (Felix Hill and Weston, 2016) repeats computing an attention vector between the query and the context through multiple layers, typically referred to as multi-hop (Alessandro Sordoni and Bengio, 2016), (Tomas Mikolov, 2013)).

3 Data collection method description, data annotation method

3.1 Question Answering

The dataset we are using is the Stanford Question Answering Dataset (SQuAD) for the Question and Answering part of the Project. It is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. SQuAD contains 107,785 question-answer pairs on 536 articles. The data is then divided into 80% training and 10% development data. The dataset is freely available at `https://stanford-qa.com`. The GloVe word embeddings (Jeffrey Pennington and Manning, 2014) used in the summarization network were the 100d embeddings obtained and used in the SQuAD Q&A project.

3.2 Summarization

The summarization network is trained on the Signal Media One-Million news dataset(`<http://research.signalmedia.co/newsir16/signal-dataset.html>`). This dataset was originally collected from a variety of news sources for a period of pone month and contains 1 million articles that are mainly English,

Figure 2: Sample data from the SQuAD Dataset

A sample data:

Context:

The further decline of Byzantine state-of-affairs paved the road to a third attack in 1185, when a large Norman army invaded Dyrrachium, owing to the betrayal of high Byzantine officials. Some time later, Dyrrachium—one of the most important naval bases of the Adriatic—fell again to Byzantine hands.

Question: What was the naval base called?

Ground Truth Answers: Dyrrachium

but they also include non-English and multi-lingual articles.

3.3 Preprocessing

For the Summarization Dataset, the articles were limited to a word count of 70-100 for consistency. Further more, the articles were lower-cased and stripped of any punctuations and non-alphabetical characters. For this project, only English articles were selected. Bi-lingual and Multi-lingual articles were eliminated to reduce learning complexity.

For the Question and Answering system, JSON files were parsed into individual components containing the Context, Query and the span of the answer.

Figure 3: Preprocessed data : Summarization

```
Original Article :

Microsoft(MSFT : 71.4) has acquired an innovator in cloud security and a
leader in helping customers protect their critical assets across cloud.Th
is acquisition is the latest example of commitment to delivering innovati
ve Identity{} and security capabilities to our C@lients across both and -
- multiple Adallom expands on existingidentity and delivers a cloud acces
s security(INTERNAL) to give customers visibility and control over applic
ation access as well as their critical company data(#security) stored acr
oss THE cloud the source article at The Official Microsoft Blog!

Preprocessed Article :

microsoft has acquired an innovator in cloud security and a leader in hel
ping customers protect their critical assets across cloud this acquisitio
n is the latest example of commitment to delivering innovative identity a
nd security capabilities to our across both and multiple adallom expands
on existingidentity and delivers a cloud access security to give customer
s visibility and control over application access as well as their critica
l company data stored across cloud the source article at the official mic
rosoft blog
```

4 Methodology

4.1 Abstractive Summarization

The model for abstractive summarizer consists of an encoder-decoder with attention architecture. The encoder and decoder are made up of three layers of recurrent neural networks. Also, Bahadanu attention mechanism (Dzmitry Bahdanau and Bengio, 2015) is added to the decoder part to improve the final output. First, we create an embedding of the words using word2vec and feed these embeddings to our encoder. The encoder creates an embedding of our input text and this embedding is fed to the decoder. The decoder takes the embedding generated by encoder along with <SOS> as input and generates the output text. The decoder also uses three layers of RNN's along with the attention mechanism to generate the output.

As mentioned earlier, we decided to reuse the 100d GloVe vectors associated with the SQuAD project to assist the summarizer in the embedding space. Once projected into the word embedding space, these vectors are fed into a standard encoder-decoder network followed by attention. The encoder-decoder models in context of recurrent neural networks (RNNs) are sequence to sequence mapping models. An RNN encoder-decoder takes a sequence as input and generates another sequence as output.For example, a sentence in English can be considered as a sequence (of words) which can be input and a French translation of the sentence is generated as an output which again is a sequence (of words). As the name suggests, encoder-decoder models consist of two parts: an encoder and a decoder. The encoder network is that part of the network that takes the input sequence and maps it to an encoded representation of the sequence. The encoded representation is then used by the decoder network to generate an output sequence.

The added mechanism of Attention helps the summarizer to know what parts of the input to concentrate on when forming the summary. Hence in terms of the number of attention weights, it will be product of the number of inputs and outputs. In this way, every combination of an input word with an output word is weighted with the amount of attention to be paid while generating the summary.

4.2 Question Answering

The Bi-Directional Attention Flow (BIDAF) network, a hierarchical multi-stage architecture for modeling the representations of the context paragraph at different levels of granularity (Figure 4) is used to build the Q/A system. BIDAF includes character-level, word-level, and contextual embed-

Figure 4: Architecture of the Summarizer

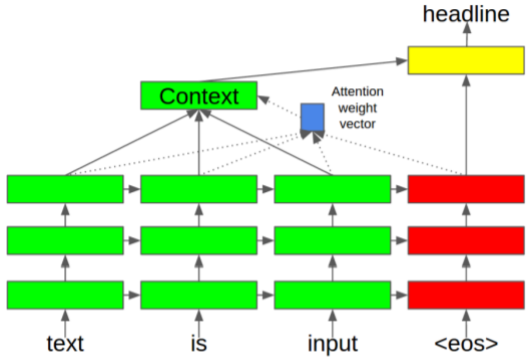
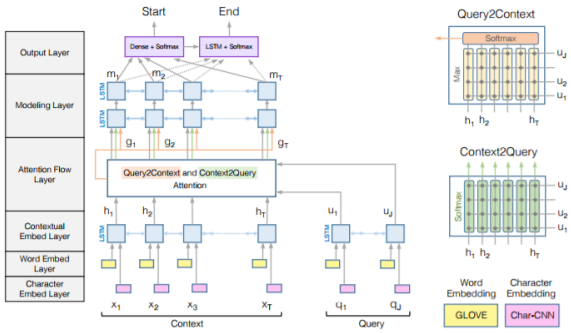


Figure 5: Architecture of the Q/A system



dings, and uses bi-directional attention flow to obtain a query-aware context representation. The machine comprehension model used in BiDAF is a hierarchical multi-stage process and consists of six layers:

1. Character Embedding Layer maps each word to a vector space using character-level CNNs.

2. Word Embedding Layer maps each word to a vector space using a pre-trained word embedding model.

3. Contextual Embedding Layer utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context.

4. Attention Flow Layer couples the query and context vectors and produces a set of query aware feature vectors for each word in the context.

5. Modeling Layer employs a Recurrent Neural Network to scan the context.

6. Output Layer provides an answer to the query.

The attention flow layer of BiDAF architecture is the most novel part of this model. Unlike traditional attention mechanisms which try to create a single feature vector from context and query, they used two attention mechanisms - Context to query attention and query to context attention. Both of these attentions were derived from a common similarity matrix between the embedding of the context and query. The context to query attention signifies which query words are most relevatn to each context word and the query to context attention signifies that which context words are the closest to the query words and hence should have more weightage to resolve the query.

5 Evaluation Methodology

5.1 Summarization Network

For the **Summarization Network**, we will use the BLEU, Rouge and F1 score evaluation metric. BLEU looks at what fraction of n-grams of different lengths from the expected summarization are actually output by the model. Rouge measures how many words or n-grams in the ground truth summaries appeared in the machine generated summaries.

$$F1measure = 2 * (\frac{BLEU * Rouge}{BLEU + Rouge}) \quad (1)$$

5.2 Question/Answering system

For **Question Answering** two metrics are used to evaluate the models.Both metrics ignore punctuations and articles (a, an, the).

1. Exact match: This metric measures the percentage of predictions that match any one of the ground truth answers exactly.

2. F1 score: This metric measures the average overlap between the prediction and ground truth answer. The prediction and ground truth are treated as bags of tokens for the F1 computation. The maximum F1 is taken over all of the ground truth answers for a given question, and then averaged over all of the questions.

6 Results and Discussion

6.1 The Q/A Network

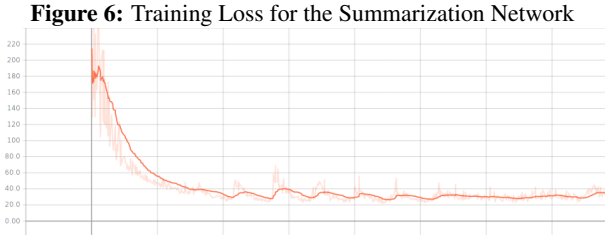
For Q/A system, we decided to use the hyperparameters provided by the authors to train the BiDAF model. They used 100D filters for CNN char embedding, each with a width of 5. To train the network, we used the AdaDelta optimizer (Zeiler, 2012) with a batch size of 100, learning rate of 0.1 and dropout rate of 0.2 for all the RNN cells. The hidden state of the RNN cells has a hidden dimension size of 100.

After training for 37 Epochs, the BiDAF model was evaluated on the SQuAD Validation Dataset with the following results:
EM score : 0.67
F1 score : 0.77.

6.2 Summarization Network

For abstractive summarization, we tried three variety of models. We first created a plain vanilla abstractive summarizer with one layer of RNN cells (GRU). We then added three layers of RNN cells to increase the capacity of the model and this results in slightly better results. After this we added the Bahadanu attention to the network to improve the summarization. To train the network, we used learning rate of 0.001 with a minibatch size of 100 and Adam optimizer (Diederik P. Kingma, 2014) for training. There are three layers of RNN cells with hidden dimension of 512 and a dropout rate of 0.2 was used for their internal states. We also used gradient clipping to help convergence. This summarizer was trained for 32 Epochs till the loss converged and it took us 1 day to train this network. We used the sparse softmax loss function between the actual and predicted summary to train the network. The BLEU score was calculated as the Baseline for the evaluation of the summarization network.
BLEU score : 0.31

Figure 6: Training Loss for the Summarization Network



The Summarizer network was then given the SQuAD Validation Data paragraphs as the input, the resultant summaries were then given to the Q/A network along with the query. The answers from the Q/A system were evaluated. The resultant metrics obtained are as follows
F1 score : 0.63
EM score : 0.52

Figure 7: Resultant summary and the answers from the Q&A system

Paragraph: Tesla gained experience in telephony and electrical engineering before emigrating to the United States in 1884 to work for Thomas Edison in New York City. He soon struck out on his own with financial backers, setting up laboratories and companies to develop a range of electrical devices. His patented AC induction motor and transformer were licensed by George Westinghouse, who also hired Tesla for a short time as a consultant. His work in the formative years of electric power development was involved in a corporate alternating current/direct current "War of Currents" as well as various patent battles.

Predicted Summary: togetherness bizarre tesla financial allman allman antar antar Edison newland blanking jock seduced burgundy telluride lunged marbled chona memberships salesian swaths

Question: What other inventor did he work with?
Answer: antar Edison

6.3 Insights

The BLEU metric tries to look at the percentage of n-grams that match in both the generated summary and the ground truth summary. This metric performs poorly when evaluating an abstractive summary as the generated summary does not contain the exact words from the paragraph but rather words similar to the words in the paragraph in an abstractive manner.
However, the Q/A system looks to see if the paragraph and the abstractive summary obtained from the summarization network contain the same information. As seen in the example in Figure 6, even though the summary does not contain all the words

from the paragraph, it contains the word Edison which is closely related to the word inventor(from the Query). Therefore, it can be said that the summary was able to retain the information from the original paragraph even though the BLEU score was sub optimal. Hence, we propose the Q/A system as a better evaluation metric for Abstractive Summarization networks.

7 Conclusion and Future Work

We have implemented a Recurrent Neural Network attention based model for abstractive summarization. This summary is evaluated using the Q/A network which is developed using the BiDAF(Bidirectional Attention Flow) model. We were able to demonstrate that the Exact Match and the F1 metrics from the Q/A network are a better metric to evaluate an abstractive summarizer.

As a next step we would like to scale this system to generate paragraph-level summaries. We believe that when we create a single sentence summaries like News headlines, it is tough to condense also possible information in the article. Hence, such summaries can't be benefit much with the help of a question and answering system. If we have paragraph-level summaries, they can probably aim to answer all the questions related to the text in a coherent manner.

Also, we would like to train both the network simultaneously or together. We believe that both the networks have a symbiotic relationship and can learn from each other. The Q/A system can learn how to produce answers even when their is some missing information in the input text. Similarly, the Summarisation network can learn to improve the summaries based on the feedback from the Q/A system. Hence, we can try to do transfer learning between these two networks. We can back propagate the network loss through the Q/A network and subsequently through the summarization network. This way we can improve the information content of the generated abstractive summary.

8 Individual Contributions

While all team members contributed to all parts of the project throughout the semester, Akash and Daniel focused on collecting the dataset and preprocessing the dataset for the summarization system, developing and implementing the summarization network. Bhavika and Vidya focused on collecting dataset and preprocessing the dataset for the Question/Answering system and then implementing the Q/A system based on the BiDaF model. All the team members contributed equally in documenting the developments of the project .

Acknowledgement

The team members would like to thank Prof. Rada Mihalcea for her insights about the approach of the project and the GSI's Steven Wilson, Laura Wendlandt for their constant assistance and timely reviews which played a vital role in the completion of the project.

References

[Alessandro Sordoni and Bengio2016] Phillip Bachman Alessandro Sordoni and Yoshua Bengio. 2016. *Iterative alternating neural attention for machine reading*.

[Diederik P. Kingma2014] Jimmy Ba Diederik P. Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Dzmitry Bahdanau and Bengio2015] Kyunghyun Cho Dzmitry Bahdanau and Yoshua Bengio. 2015. *Neural machine translation by jointly learning to align and translate*.

[Felix Hill and Weston2016] Sumit Chopra Felix Hill, Antoine Bordes and Jason Weston. 2016. *The goldilocks principle: Reading childrens books with explicit memory representations*. ICLR.

[Jeffrey Pennington and Manning2014] Richard Socher Jeffrey Pennington and Christopher D. Manning. 2014. *GloVe: Global Vectors for Word Representation*.

[Lopyrev2015] Konstantin Lopyrev. 2015. *Generating News Headlines with Recurrent Neural Networks*. arXiv.

[Minjoon Seo2017] Ali Farhadi Hananneh Hajishirzi Minjoon Seo, Aniruddha Kembhavi. 2017. *Bi-Directional attention flow for machine comprehension*. ICLR.

[Romain Paulus2017] Richard Socher Romain Paulus, Caiming Xiong. 2017. *A Deep Reinforced Model for Abstractive Summarization*. arXiv.

[Rudolf Kadlec and Kleindienst.2016] Ondrej Bajgar Rudolf Kadlec, Martin Schmid and Jan Kleindienst.

2016. *Text understanding with the attention sum reader network*.

[Tomas Mikolov2013] Greg Corrado Jeffrey Dean Tomas Mikolov, Kai Chen. 2013. *Efficient Estimation of Word Representations in Vector Space*.

[Verma and Nidhi2017] Sukriti Verma and Vagisha Nidhi. 2017. Extractive summarization using deep learning. *arXiv preprint arXiv:1708.04439*.

[Zeiler2012] Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.