Read, Highlight and Summarize: A Hierarchical Neural Semantic Encoder-based Approach

Abstract

Traditional sequence-to-sequence (seq2seq) models and other variations of the attention-mechanism such as hierarchical attention have been applied to the text summarization problem. Though there is a hierarchy in the way humans use language by forming paragraphs from sentences and sentences from words, hierarchical models have usually not worked that much better than their traditional seq2seq counterparts. This effect is mainly because either the hierarchical attention mechanisms are too sparse using hard attention or noisy using soft attention. In this paper, we propose a method based on extracting the highlights of a document; a key concept that is conveyed in a few sentences. In a typical text summarization dataset consisting of documents that are 800 tokens in length (average), capturing long-term dependencies is very important, e.g., the last sentence can be grouped with the first sentence of a document to form a summary. LSTMs (Long Short-Term Memory) proved useful for machine translation. However, they often fail to capture long-term dependencies while modeling long sequences. To address these issues, we have adapted Neural Semantic Encoders (NSE) to text summarization, a class of memory-augmented neural networks by improving its functionalities and proposed a novel hierarchical NSE that outperforms similar previous models significantly. The quality of summarization was improved by augmenting linguistic factors, namely lemma, and Part-of-Speech (PoS) tags, to each word in the dataset for improved vocabulary coverage and generalization. The hierarchical NSE model on factored dataset outperformed the state-of-the-art by nearly 4 ROUGE points. We further designed and used the first GPU-based self-critical Reinforcement Learning model.

Introduction

When there are a very large number of documents that need to be read in limited time, we often resort to

reading summaries instead of the whole document. Automatically generating (abstractive) summaries is a problem with various applications, e.g., automatic authoring BIBREF0. We have developed automatic text summarization systems that condense large documents into short and readable summaries. It can be used for both single (e.g., BIBREF1, BIBREF2 and BIBREF3) and multi-document summarization (e.g.,BIBREF4, BIBREF3, BIBREF5).

Text summarization is broadly classified into two categories: extractive (e.g., BIBREF3 and BIBREF6) and abstractive summarization (e.g., BIBREF7, BIBREF8 and BIBREF9). Extractive approaches select sentences from a given document and groups them to form concise summaries. By contrast, abstractive approaches generate human-readable summaries that primarily capture the semantics of input documents and contain rephrased key content. The former task falls under the classification paradigm, and the latter belongs to the generative modeling paradigm, and therefore, it is a much harder problem to solve. The backbone of state-of-the-art summarization models is a typical encoder-decoder BIBREF10 architecture that has proved to be effective for various sequential modeling tasks such as machine translation, sentiment analysis, and natural language generation. It contains an encoder that maps the raw input word vector representations to a latent vector. Then, the decoder usually equipped with a variant of the attention mechanism BIBREF11 uses the latent vectors to generate the output sequence, which is the summary in our case. These models are trained in a supervised learning setting where we minimize the cross-entropy loss between the predicted and the target summary. Encoder-decoder models have proved effective for short sequence tasks such as machine translation where the length of a sequence is less than 120 tokens. However, in text summarization, the length of the sequences vary from 400 to 800 tokens, and modeling long-term dependencies becomes increasingly difficult.

Despite the metric's known drawbacks, text summarization models are evaluated using ROUGE BIBREF12, a discrete similarity score between predicted and target summaries based on 1-gram, 2-gram, and n-gram overlap. Cross-entropy loss would be a convenient objective on which to train the model

since ROUGE is not differentiable, but doing so would create a mismatch between metrics used for training and evaluation. Though a particular summary scores well on ROUGE evaluation comparable to the target summary, it will be assigned lower probability by a supervised model. To tackle this problem, we have used a self-critic policy gradient method BIBREF13 to train the models directly using the ROUGE score as a reward. In this paper, we propose an architecture that addresses the issues discussed above.

Introduction ::: Problem Formulation

Let $D=\lbrace d_{1}, d_{2}, ..., d_{N}\rbrace $ be the set of document sentences where each sentence $d_{i}, 1 \le i \le N$ is a set of words and $S=\lbrace s_{1}, s_{2}, ..., s_{M}\rbrace $ be the set of summary sentences. In general, most of the sentences in D are a continuation of another sentence or related to each other, for example: in terms of factual details or pronouns used. So, dividing the document into multiple paragraphs as done by BIBREF4 leaves out the possibility of a sentence-level dependency between the start and end of a document. Similarly, abstracting a single document sentence as done by BIBREF9 cannot include related information from multiple document sentences. In a good human-written summary, each summary sentence is a compressed version of a few document sentences. Mathematically,

Where C is a compressor we intend to learn. Figure FIGREF3 represents the fundamental idea when using a sequence-to-sequence architecture. For a sentence s in summary, the representations of all the related document sentences $d_{1}, d_{2}, ..., d_{K}$ are expected to form a cluster that represents a part of the highlight of the document.

First, we adapt the Neural Semantic Encoder (NSE) for text summarization by improving its attention mechanism and compose function. In a standard sequence-to-sequence model, the decoder has access

to input sequence through hidden states of an LSTM BIBREF14, which suffers from the difficulties that we discussed above. The NSE is equipped with an additional memory, which maintains a rich representation of words by evolving over time. We then propose a novel hierarchical NSE by using separate word memories for each sentence to enrich the word representations and a document memory to enrich the sentence representations, which performed better than its previous counterparts (BIBREF7, BIBREF3, BIBREF15). Finally, we use a maximum-entropy self-critic model to achieve better performance using ROUGE evaluation.

Related Work

The first encoder-decoder for text summarziation is used by BIBREF1 coupled with an attention mechanism. Though encoder-decoder models gave a state-of-the-art performance for Neural Machine Translation (NMT), the maximum sequence length used in NMT is just 100 tokens. Typical document lengths in text summarization vary from 400 to 800 tokens, and LSTM is not effective due to the loss in memory over time for very long sequences. BIBREF7 used hierarchical attentionBIBREF16 to mitigate this effect where, a word LSTM is used to encode (decode) words, and a sentence LSTM is used to encode (decode) sentences. The use of two LSTMs separately for words and sentences improves the ability of the model to retain its memory for longer sequences. Additionally, BIBREF7 explored using a hierarchical model consisting of a feature-rich encoder incorporating position, Named Entity Recognition (NER) tag, Term Frequency (TF) and Inverse Document Frequency (IDF) scores. Since an RNN is a sequential model, computing at one time-step needs all of the previous time-steps to have computed before and is slow because the computation at all the time steps cannot be performed in parallel. BIBREF8 used convolutional layers coupled with an attention mechanism BIBREF11 to increase the speed of the encoder. Since the input to an RNN is fed sequentially, it is expected to capture the positional information. But both works BIBREF7 and BIBREF8 found positional embeddings to be quite useful for reasons unknown. BIBREF3 proposed an extractive summarization model that classifies

sentences based on content, saliency, novelty, and position. To deal with out-of-vocabulary (OOV) words and to facilitate copying salient information from input sequence to the output, BIBREF2 proposed a pointer-generator network that combines pointing BIBREF17 with generation from vocabulary using a soft-switch. Attention models for longer sequences tend to be repetitive due to the decoder repeatedly attending to the same position from the encoder. To mitigate this issue, BIBREF2 used a coverage mechanism to penalize a decoder from attending to same locations of an encoder. However, the pointer generator and the coverage model BIBREF2 are still highly extractive; copying the whole article sentences 35% of the time. BIBREF18 introduced an intra-attention model in which attention also depends on the predictions from previous time steps.

One of the main issues with sequence-to-sequence models is that optimization using the cross-entropy objective does not always provide excellent results because the models suffer from a mismatch between the training objective and the evaluation metrics such as ROUGE BIBREF12 and METEOR BIBREF19. A popular algorithm to train a decoder is the teacher-forcing algorithm that minimizes the negative log-likelihood (cross-entropy loss) at each decoding time step given the previous ground-truth outputs. But during the testing stage, the prediction from the previous time-step is fed as input to the decoder instead of the ground truth. This exposure bias results in error accumulation over each time step because the model has never been exposed to its predictions during training. Instead, recent works show that summarization models can be trained using reinforcement learning (RL) where the ROUGE BIBREF12 score is used as the reward (BIBREF18, BIBREF9 and BIBREF4).

BIBREF5 made such an earlier attempt by using Q-learning for single-and multi-document summarization. Later, BIBREF15 proposed a coarse-to-fine hierarchical attention model to select a salient sentence using sentence attention using REINFORCE BIBREF20 and feed it to the decoder. BIBREF6 used REINFORCE to rank sentences for extractive summarization. BIBREF4 proposed deep communicating agents that operate over small chunks of a document, which is learned using a self-critical

BIBREF13 training approach consisting of intermediate rewards. BIBREF9 used a advantage actor-critic (A2C) method to extract sentences followed by a decoder to form abstractive summaries. Our model does not suffer from their limiting assumption that a summary sentence is an abstracted version of a single source sentence. BIBREF18 trained their intra-attention model using a self-critical policy gradient algorithm BIBREF13. Though an RL objective gives a high ROUGE score, the output summaries are not readable by humans. To mitigate this problem, BIBREF18 used a weighted sum of supervised learning loss and RL loss.

Humans first form an abstractive representation of what they want to say and then try to put it into words while communicating. Though it seems intuitive that there is a hierarchy from sentence representation to words, as observed by both BIBREF7 and BIBREF15, these hierarchical attention models failed to outperform a simple attention model BIBREF1. Unlike feedforward networks, RNNs are expected to capture the input sequence order. But strangely, positional embeddings are found to be effective (BIBREF7, BIBREF8, BIBREF15 and BIBREF3). We explored a few approaches to solve these issues and improve the performance of neural models for abstractive summarization.

Proposed Models

In this section, we first describe the baseline Neural Semantic Encoder (NSE) class, discuss improvements to the compose function and attention mechanism, and then propose the Hierarchical NSE. Finally, we discuss the self-critic model that is used to boost the performance further using ROUGE evaluation.

Proposed Models ::: Neural Semantic Encoder:

A Neural Semantic Encoder BIBREF21 is a memory augmented neural network augmented with an

encoding memory that supports read, compose, and write operations. Unlike the traditional sequence-to-sequence models, using an additional memory relieves the LSTM of the burden to remember the whole input sequence. Even compared to the attention-model BIBREF11 which uses an additional context vector, the NSE has anytime access to the full input sequence through a much larger memory. The encoding memory is evolved using basic operations described as follows:

Where, $x_{t} \in \mathbb {R}^D$ is the raw embedding vector at the current time-step. f_{r}^{LSTM} , f_{c}^{MLP} (Multi-Layer Perceptron), f_{w}^{LSTM} be the read, compose and write operations respectively. $e_{l} \in R^{l}$, $e_{k} \in R^{k}$ are vectors of ones, $\mathbf {1}$ is a matrix of ones and $\otimes $ is the outer product.

Instead of using the raw input, the read function f_{r}^{LSTM} in equation DISPLAY_FORM5 uses an LSTM to project the word embeddings to the internal space of memory M_{t-1} to obtain the hidden states o_{t}. Now, the alignment scores z_{t} of the past memory M_{t-1} are calculated using o_{t} as the key with a simple dot-product attention mechanism shown in equation DISPLAY_FORM6. A weighted sum gives the retrieved input memory that is used in equation DISPLAY_FORM8 by a Multi-Layer Perceptron in composing new information. Equation DISPLAY_FORM9 uses an LSTM and projects the composed states into the internal space of memory M_{t-1} to obtain the write states h_{t}. Finally, in equation DISPLAY_FORM10, the memory is updated by erasing the retrieved memory as per z_{t} and writing as per the write vector h_{t}. This process is performed at each time-step throughout the input sequence. The encoded memories $\lbrace M\rbrace _{t=1}^{T}$ are similarly used by the decoder to obtain the write vectors $\lbrace h\rbrace _{t=1}^{T}$ that are eventually fed to projection and softmax layers to get the vocabulary distribution.

Proposed Models ::: Improved NSE

Although the vanilla NSE described above performed well for machine translation, just a dot-product attention mechanism is too simplistic for text summarization. In machine translation, it is sufficient to compute the correlation between word-vectors from the semantic spaces of different languages. In contrast, text summarization also needs a word-sentence and sentence-sentence correlation along with the word-word correlation. So, in search of an attention mechanism with a better capacity to model the complex semantic relationships inherent in text summarization, we found that the additive attention mechanism BIBREF11 given by the equation below performs well.

Where, v, W, U, b_{attn} are learnable parameters. One other important difference is the compose function: a Multi-layer Perceptron (MLP) is enough for machine translation as the sequences are short in length. However, text summarization consists of longer sequences that have sentence-to-sentence dependencies, and a history of previously composed words is necessary for overcoming repetition BIBREF1 and thereby maintaining novelty. A powerful function already at our disposal is the LSTM; we replaced the MLP with an LSTM, as shown below:

In a standard text summarization task, due to the limited size of word vocabulary, out-of-vocabulary (OOV) words are replaced with [UNK] tokens. pointer-networks BIBREF17 facilitate the ability to copy words from the input sequence to the output via pointing. Later, BIBREF2 proposed a hybrid pointer-generator mechanism to improve upon pointing by retaining the ability to generate new words. It points to the words from the input sequence and generates new words from the vocabulary. A generation probability $p_{gen} \in (0, 1)$ is calculated using the retrieved memories, attention distribution, current input hidden state o_{t} and write state h_{t} as follows:

Where, $W_{m}, W_{h}, W_{o}, b_{ptr}$ are learnable parameters, and $\sigma $ is the sigmoid activation function. Next, p_{gen} is used as a soft switch to choose between generating a word from the vocabulary by sampling from p_{vocab}, or copying a word from the input sequence by sampling from

the attention distribution z_{t}. For each document, we maintain an auxiliary vocabulary of OOV words in the input sequence. We obtain the following final probability distribution over the total extended vocabulary:

Note that if w is an OOV word, then $p_{vocab}(w)$ is zero; similarly, if w does not appear in the source document, then $\sum _{i:w = w_{i}} z_{i}^{t}$ is zero. The ability to produce OOV words is one of the primary advantages of the pointer-generator mechanism. We can also use a smaller vocabulary size and thereby speed up the computation of output projection and softmax layers.

Proposed Models ::: Hierarchical NSE

When humans read a document, we organize it in terms of word semantics followed by sentence semantics and then document semantics. In a text summarization task, after reading a document, sentences that have similar meanings or continual information are grouped together and then expressed in words. Such a hierarchical model was first introduced by BIBREF16 for document classification and later explored unsuccessfully for text summarization BIBREF3. In this work, we propose to use a hierarchical model with improved NSE to take advantage of both augmented memory and also the hierarchical document representation. We use a separate memory for each sentence to represent all the words of a sentence and a document memory to represent all sentences. Word memory composes novel words, and document memory composes novel sentences in the encoding process that can be later used to extract highlights and decode to summaries as shown in Figure FIGREF17.

Let $D = \lbrace (w_{ij})_{j=1}^{T_{in}}\rbrace _{i=1}^{S_{in}}$ be the input document sequence, where S_{in} is the number of sentences in a document and T_{in} is the number of words per sentence. Let $\lbrace M_{i}\rbrace _{i=1}^{S_{in}}, M_{i} \in R^{T_{in} \times D}$ be the sentence memories that encode all the words in a sentence and $M^{d}, M^{d} \in R^{S_{in} \times D}$ be the document memory

that encodes all the sentences present in the document. At each time-step, an input token x_{t} is read and is used to retrieve aligned content from both corresponding sentence memory $M_{t}^{i, s}$ and document memory M_{t}^{d}. Please note that the retrieved document memory, which is a weighted combination of all the sentence representations forms a highlight. After composition, both the sentence and document memories are written simultaneously. This way, the words are encoded with contextual meaning, and also new simpler sentences are formed. The functionality of the model is as follows:

Where, f_{attn} is the attention mechanism given by equation(DISPLAY_FORM12). $Update$ remains the same as the vanilla NSE given by equation(DISPLAY_FORM10)and $Concat$ is the vector concatenation. Please note that NSE BIBREF21 has a concept of shared memory but we use multiple memories for representing words and a document memory for representing sentences, this is fundamentally different to a shared memory which does not have a concept of hierarchy.

Proposed Models ::: Self-Critical Sequence Training

As discussed earlier, training in a supervised learning setting creates a mismatch between training and testing objectives. Also, feeding the ground-truth labels in training time-step creates an exposure bias while testing in which we feed the predictions from the previous time-step. Policy gradient methods overcome this by directly optimizing the non-differentiable metrics such as ROUGE BIBREF12 and METEOR BIBREF19. It can be posed as a Markov Decision Process in which the set of actions $\mathcal {A}$ is the vocabulary and reward $\mathcal {R}$ is the ROUGE score itself. So, we should find a policy $\pi (\theta)$ such that the set of sampled words $\tilde{y} = \lbrace \tilde{y}_{1}, \tilde{y}_{2}, ..., \tilde{y}_{T}\rbrace $ achieves highest ROUGE score among all possible summaries.

We used the self-critical model of BIBREF13 proposed for image captioning. In self-critical sequence training, the REINFORCE algorithm BIBREF20 is used by modifying its baseline as the greedy output of

the current model. At each time-step t, the model predicts two words: $\hat{y}_{t}$ sampled from $p(\hat{y}_{t} | \hat{y}_{1}, \hat{y}_{2}, ..., \hat{y}_{t-1}, x)$, the baseline output that is greedily generated by considering the most probable word from the vocabulary and $\tilde{y}_{t}$ sampled from the $p(\tilde{y}_{t} | \tilde{y}_{1}, \tilde{y}_{2}, ..., \tilde{y}_{t-1}, x)$. This model is trained using the following loss function:

Using the above training objective, the model learns to generate samples with high probability and thereby increasing $r(\tilde{y})$ above $r(\hat{y})$. Additionally, we have used enthttps://stackoverflow.com/questions/19053077/looping-over-data-and-creating-individual-figuresropy regularization.

Where, $p(\tilde{y}_{t})=p(\tilde{y}_{t} | \tilde{y}_{1}, \tilde{y}_{2}, ..., \tilde{y}_{t-1}, x)$ is the sampling probability and V is the size of the vocabulary. It is similar to the exploration-exploitation trade-off. $\alpha $ is the regularization coefficient that explicitly controls this trade-off: a higher $\alpha $ corresponds to more exploration, and a lower $\alpha $ corresponds to more exploitation. We have found that all TensorFlow based open-source implementations of self-critic models use a function (tf.py_func) that runs only on CPU and it is very slow. To the best of our knowledge, ours is the first GPU based implementation.

Experiments and Results ::: Dataset

We used the CNN/Daily Mail dataset BIBREF7, which has been used as the standard benchmark to compare text summarization models. This corpus has 286,817 training pairs, 13,368 validation pairs, and 11,487 test pairs, as defined by their scripts. The source document in the training set has 766 words spanning 29.74 sentences on an average while the summaries consist of 53 words and 3.72 sentences BIBREF7. The unique characteristics of this dataset such as long documents, and ordered multi-sentence

summaries present exciting challenges, mainly because the proven sequence-to-sequence LSTM based models find it hard to learn long-term dependencies in long documents. We have used the same train/validation/test split and examples for a fair comparison with the existing models.

The factoring of lemma and Part-of-Speech (PoS) tag of surface words, are observed BIBREF22 to increase the performance of NMT models in terms of BLEU score drastically. This is due to the improvement of the vocabulary coverage and better generalization. We have added a pre-processing step by incorporating the lemma and PoS tag to every word of the dataset and training the supervised model on the factored data. The process of extracting the lemma and the PoS tags has been described in BIBREF22. Please refer to the appendix for an example of factoring.

Experiments and Results ::: Training Settings

For all the plain NSE models, we have truncated the article to a maximum of 400 tokens and the summary to 100 tokens. For the hierarchical NSE models, articles are truncated to have a maximum of 20 sentences and 20 words per sentence each. Shorter sequences are padded with `PAD` tokens. Since the factored models have lemma, PoS tag and the separator `|` for each word, sequence lengths should be close to 3 times the non-factored counterparts. For practical reasons of memory and time, we have used 800 tokens per article and 300 tokens for the summary.

For all the models, including the pointer-generator model, we use a vocabulary size of 50,000 words for both source and target. Though some previous works BIBREF7 have used large vocabulary sizes of 150,000, since our models have a copy mechanism, smaller vocabulary is enough to obtain good performance. Large vocabularies increase the computation time. Since memory plays a prominent role in retrieval and update, it is vital to start with a good initialization. We have used 300-dimensional pre-trained GloVe BIBREF23 word-vectors to represent the input sequence to a model. Sentence

memories are initialized with GloVe word-vectors of all the words in that sentence. Document memories are initialized with vector representations of all the sentences where a sentence is represented with the average of the GloVe word-vectors of all its words. All the models are trained using the Adam optimizer with the default learning rate of 0.001. We have not applied any regularization as the usage of dropout, and L_{2} penalty resulted in similar performance, however with a drastically increased training time.

The Hierarchical models process one sentence at a time, and hence attention distributions need less memory, and therefore, a larger batch size can be used, which in turn speeds up the training process. The non-factored model is trained on 7-NVIDIA Tesla-P100 GPUs with a batch size of 448 (64 examples per GPU); it takes approximately 45 minutes per epoch. Since the factored sequences are long, we used a batch size of 96 (12 examples per GPU) on 8-NVIDIA Tesla-V100 GPUs. The Hier model reaches optimal cross-entropy loss in just 8 epochs, unlike 33-35 epochs for both BIBREF7 and BIBREF2. For the self-critical model, training is started from the best supervised model with a learning rate of 0.00005 and manually changed to 0.00001 when needed with $\alpha =0.0001$ and the reported results are obtained after training for 15 days.

Experiments and Results ::: Evaluation

All the models are evaluated using the standard metric ROUGE; we report the F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L, which quantitively represent word-overlap, bigram-overlap, and longest common subsequence between reference summary and the summary that is to be evaluated. The results are obtained using pyrouge package. The performance of various models and our improvements are summarized in Table TABREF37. A direct implementation of NSE performed very poorly due to the simple dot-product attention mechanism. In NMT, a transformation from word-vectors in one language to another one (say English to French) using a mere matrix multiplication is enough because of the one-to-one correspondence between words and the underlying linear structure imposed in learning the

word vectors BIBREF23. However, in text summarization a word (sentence) could be a condensation of a group of words (sentences). Therefore, using a complex neural network-based attention mechanism proposed improved the performance. Both dot-product and additive BIBREF11 mechanisms perform similarly for the NMT task, but the difference is more pronounced for the text summarization task simply because of the nature of the problem as described earlier. Replacing Multi-Layered Perceptron (MLP) in the NSE with an LSTM further improved the performance because it remembers what was previously composed and facilitates the composition of novel words. This also eliminates the need for additional mechanisms to penalize repetitions such as coverage BIBREF2 and intra-attention BIBREF18. Finally, using memories for each sentence enriches the corresponding word representation, and the document memory enriches the sentence representation that help the decoder. Please refer to the appendix for a few example outputs. Table TABREF34 shows the results in comparison to the previous methods. Our hierarchical model outperforms BIBREF7 (HIER) by 5 ROUGE points. Our factored model achieves the new state-of-the-art (SoTA) result, outperforming BIBREF4 by almost 4 ROUGE points.

Conclusion

In this work, we presented a memory augmented neural network for the text summarization task that addresses the shortcomings of LSTM-based models. We applied a critical pre-processing step by factoring the dataset with inherent linguistic information that outperforms the state-of-the-art by a large margin. In the future, we will explore new sparse functions BIBREF24 to enforce strict sparsity in selecting highlights out of sentences. The general framework of pre-processing, and extracting highlights can also be used with powerful pre-trained models like BERT BIBREF25 and XLNet BIBREF26.

Appendix

Figure FIGREF38 below shows the self-critical model. All the examples shown in Tables

TABREF39-TABREF44 are chosen as per the shortest article lengths available due to space constraints.