Fusing External Language Model in Abstractive Summarization

Anonymous submission

Abstract

Recent sequence-to-sequence neural network models provide a viable new solution to abstractive text summarization, which aims to rewrite a long text into a short and concise form while preserving the most crucial information. However, these models face significant challenges when generating both semantically and syntactically correct summaries. In this work, we explore the potential approaches to incorporate an external (pre-trained) language model to augment the linguistic quality of text generation. This allows the internal (decoder) language model to focus more on jointly learning summary content selection and generation. Fused with the external language model, our abstractive summarization model achieves the results comparable to state-of-the-art models in terms of ROUGE scores, and meanwhile shows significant improvements in both perplexity and human evaluations.

1 Introduction

Text summarization aims to generate a short natural language summary that compress the information in the original longer text. Summarization approaches fall into two broad categories: extractive and abstractive. Extractive approaches (Cheng and Lapata, 2016; Narayan et al., 2018) typically assemble summaries from passages taken directly from the source text, while abstractive approaches (Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017) are supposed to freely express with new words and phrases not featured in the source text. The recent success of sequence-to-sequence neural network models (Bahdanau et al., 2014) makes abstractive summarization a viable option. The summaries generated by state-of-the-art abstractive summarization models may have high word overlaps when compared against the gold summaries. However, when taking a closer look, the repeated text and the un-grammatical sentences are not uncommon in generated summaries. High ROUGE (Lin and Hovy, 2003) score does not guarantee the good quality and readability of summaries. In light of this problem, (See et al., 2017) introduce a coverage mechanism to address the repeated text issue. Meanwhile (Paulus et al., 2017) propose the intra-decoder attention and (Liu et al., 2018) equip with an additional discriminator to improve summary fluency. (Paulus et al., 2017) also notice that even though the best ROUGE score can be achieved by replacing the maximum likelihood objective with Reinforcement Learning (RL) to directly optimize the ROUGE metric, their RL approach tends to produce non-grammatical text and thus performs the worst in human evaluations. While existing approaches have shown to improve summary readability and fluency to some extent, they share the following limitations. The proposed mechanisms or strategies mainly cope with the language quality problem from very specific points of views, and thus cannot provide the general solution. For example, the coverage mechanism (See et al., 2017) helps generate summaries with less repeated text, but the disfluency and ungrammatical problem still severe. More important, training sequence-to-sequence models to improve the inadequate readability would be a great burden on the decoder. In essence, the decoder has two roles. One is related to summarization, i.e., to copy and fuse different parts in the source sentence using the attention mechanism. The other is related to text generation, i.e., to function like a language model. We refer to the decoder model as the Internal Language Model (ILM) considering it is a component within the sequence-to-sequence model. The supervised-learning nature limits its ability to sufficiently learn the language modeling ability with the current available manually generated summarization training data. In this work, we

propose to empower the language ability of a neural summarization model with the External Language Model (ELM), which theoretically can be pre-trained on the unlimited amount of raw text. ELM releases the burden of IML by taking more care of language fluency. Consequently, it enables ILM to focus more on important content selection and summary generation. With this idea, we expect that the generated summaries can be improved both semantically and syntactically. This is demonstrated by our experimental results.

2 Model

Neural abstractive summarization model was first proposed in (Rush et al., 2015). In what follows, we will introduce (1) the basic framework of neural abstractive summarization, (2) our approaches to fuse the external language model, and (3) the soft switching mechanism used to adjust the relative contributions of ILM and ELM at each time step during generation.

2.1 Neural Abstractive Summarization Framework

Neural abstract summarization model has followed the literature proposed by (Bahdanau et al., 2014). The encoder, i.e., a Bi-directional RNN (BiRNN), obtains the representation of a word by concatenating its corresponding forward hidden state and the backward hidden state, i.e., $h_j = [\overrightarrow{h_j}; \overleftarrow{h_j}]$. In this way, the representation is sensitive to both the preceding words and the following words. The decoder is essentially a conditioned language model which learns to predict a probability distribution based on the word predicted at the last timestep, hidden state, and context vector in the current timestep, which is formulated as:

$$p(y_t|y_1, ..., y_{t-1}, x) = g(y_{t-1}, s_t, c_t) \quad (1)$$

where y_t is the predicted word, s_t is the RNN hidden state at time t, computed by:

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

The context vector c_t is derived from $(h_1, ..., h_j)$, for example as a weighted sum of h_j, i.e.,

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j \quad (2)$$

The weight α_{tj} of h_j is computed by

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T_x} exp(e_{tk})} \quad (3)$$

where $e_{tj} = a(s_{t-1}, h_j)$. The context vector is then concatenated with the decoder state s_t and fed through two linear layers to produce the vocabulary distribution P_{vocab}.

$$P_{vocab}(\hat{y}_t) = softmax(V^{'}(V[s_t, c_t] + b) + b^{'}) \quad (4)$$

where $\hat{y}_t$ is the predicted word, $V, V^{'}, b, b^{'}$ are learnable parameters. During training, the loss at each time step is a negative log likelihood of the target word y_t^* and the overall loss for the entire generated sequence is:

$$loss = \frac{1}{T} \sum_{t=0}^{T} -logP(\hat{y}_t, y_t^*) \quad (5)$$

2.2 Fusion With External Language Model

The previously proposed *Parameterized-based Combination* approach (Gülçehre et al., 2015), which incorporates a language model with a decoder for machine translation, makes the following change during inference.

$$\hat{y} = \underset{y}{\text{argmax}}(logP(y|x) + \lambda logP_{ELM}(y)) \quad (6)$$

where $P_{ELM}(\cdot)$ is the probability produced by the external language model RNN-LM (Mikolov et al., 2010) trained on the monolingual target language data. $P(\cdot)$ is the probability of a typical sequential-to-sequential model (Bahdanau et al., 2014). λ is a hyper-parameter experimentally tuned on the development dataset. This approach is a good starting point of our work. However, the most significant disadvantage of it is that ILM is trained independently of ELM. It limits the ILMs ability to sufficiently absorb the prior language knowledge from ELM. To solve this problem, we propose the following three new fusion approaches to incorporate ELM into ILM during both training and inference.

2.2.1 Prediction-based Fusion

Since both the external language model and the summarization model predict output words, the natural point to connect the two models is joining them at the output prediction stage by unifying their probability distributions. More specifically, at each time step t, the final predicted probability distribution is determined by:

$$P_{Fuse}(y) = P_{ILM}(y|x; \theta) + g_t \cdot P_{ELM}(y; \theta^{'}) \quad (7)$$

where $P_{ILM}(\cdot)$ and $P_{ELM}(\cdot)$ are the probabilities of ILM and ELM, respectively. g_t is a

soft switch balancing the relative contributions of ELM and ILM. We will explain the soft-switching mechanism in more detail later. We call this approach *prediction-based fusion* (*PredFusion* for short) approach. *Parameterized-based combination* (Gülçehre et al., 2015) can be regarded as a special case of our method when g_t is fixed to λ. However, the main difference is that we train the summarization decoder from scratch jointly with ELM, which enables the decoder to learn the language structure from ELM during training and thus eliminates the training and inference bias.

2.2.2 Feature-based Fusion

On the other hand, the *feature-based fusion* (*FeatFusion* for short) approach concatenates the hidden states of ELM and ILM when computing the probability of the next output word. Compared with *prediction-based fusion*, it tightens the connection between ELM and ILM in the feature space. Unlike the vanilla ILM, the RNN hidden state after fusion at time t is computed by:

$$s_t^{ILM} = f(s_{t-1}^{ILM}, y_{t-1}^{ILM}, c_t^{ILM}) \tag{8}$$

$$s_t^{ELM} = f'(s_{t-1}^{ELM}, y_{t-1}^{ILM}) \tag{9}$$

$$s_t^{Fusion} = s_t^{ILM} + g_t \cdot s_t^{ELM} \tag{10}$$

where s_t^{ILM} and s_t^{ELM} are hidden states of ILM and ELM, respectively, s_t^{Fusion} is the hidden state after fusion. Here, we also apply the soft-switching mechanism to balance the contributions of ELM and ILM. In contrast to *PredFusion*, *FeatFusion* adjusts the importance of each language model component in the feature space. It thus acts as an experienced linguistic teacher who does not just tell the student what the potential candidates are but also makes the student understand the semantic structure entailed in each generation through the representation fusion.

2.2.3 Vocabulary-Constrained Fusion

Another intuitive solution is to narrow down the vocabulary with top-ranked word candidates selected according to the probability distribution of ELM, when the ILM decoder predicts the next word. We call it the *vocabulary-constrained fusion* (*VocaFusion* for short) approach. At each time step, ELM first computes the probability of every possible next word in the vocabulary. All words are ranked according to their respective probability numerical value $\{\hat{y}_t^{(j)}\}_{j=1,\ldots,k,\ldots,n}$. The top K ones are then suggested to the decoder via a mask so that the potential next words are constrained within a limited vocabulary. This helps to accelerate the compution speed and to decrease the noise.

$$P_{Fusion}(y_t) = P_{ILM}(y_t|x;\theta) \times P_{mask}(y_t) \tag{11}$$

where

$$P_{mask}(y_t^{(i)}) = \begin{cases} 1, & P(y_t^{(i)}) \geq P(\hat{y}_t^{(k)}) \\ 0, & P(y_t^{(i)}) < P(\hat{y}_t^{(k)}) \end{cases} \tag{12}$$

$P_{mask}(y_t^{(i)})$ is the mask value for the ith index in the vocabulary, $P(\hat{y}_t^{(k)})$ is the minimum value of the ranked top K probability. *Vocabulary-constrained fusion* is an interactive learning process. ELM neither provides the prediction probability to ILM like *Prediction-based Fusion* nor concatenates representations like *Feature-based Fusion*. It effectively guides the learning of ILM by removing unnecessary vocabulary. Notice that in all the proposed approaches, the input to ELM is the word predicted by ILM (rather than ELM itself) at the last time step.

2.3 Soft Switching Mechanism

In order to balance the importance of internal language model and external language model in the training process, we utilize a soft switch to achieve fine-grained adjustment in *prediction-based fusion* and *feature-based fusion*.

$$g_t = \sigma(W[s_t^{ILM}; s_t^{ELM}] + b) \tag{13}$$

where g_t is the switch value at time t, s_t^{ILM} is the hidden state of the internal language model, and s_t^{ELM} is the hidden state of the external language model. Using different switch values at different time steps allows the summarization model to incorporate ELM flexibly, because the switch is able to decide which component should be emphasized based on the current hidden state.

3 Experiment

3.1 Dataset

CNN/Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016) is a benchmark dataset for document abstract summarization, in which the online news articles are paired with multi-sentence

summaries. There are 287,226 training pairs, 13,368 validation pairs, and 14,490 test pairs. For a fair comparison with previous approaches, we use the original text like (Nallapati et al., 2016; See et al., 2017; Liu et al., 2018; Kryscinski et al., 2018). We use the CNN/Daily Mail dataset to train our summarization model and external language model, following the same training, validation, and test splits.

3.2 Experiment Setup

Our BiRNN encoder, decoder and external language model are all 512-dimensional LSTMs. We pretrain ELM and fix it. ILM is then expected to learn language features from ELM. We train entire system end-to-end with Adam (Kingma and Ba, 2014) and with a batch size of 16. We limit the size of input articles to the first 400 tokens, and the summaries to 100 tokens. During interference, we used beam search with a fixed beam size of 5 for all of our experiments.

3.3 Results

We compare our approaches with three existing NN-based methods with the best reported ROUGE scores, including the Pointer Generator Network (See et al., 2017), SumGAN (Liu et al., 2018), and NovelSum (Kryscinski et al., 2018). Table 1 shows the ROUGE scores of our models and other models, where the second section is our implement of pointer generator network, with scores slightly lower than scores reported in the paper as the coverage mechanism implements different in Pytorch. We can see that our fusion models achieve better results compared with state-of-the-art models which optimized by maximum likelihood. Furthermore, the optimization objective of SumGAN and NovelSum is different, which directly optimize the ROUGE metric by Reinforcement Learning proved to promote ROUGE scores in a large margin. In spite of this, our models achieve comparable results. Among them, feature-based fusion performs the best. This is consistent with our intuition. We also run the ablation studies to exam the impact of soft switch. We find that the soft switch is an essential component especially in *FeatFusion*.

We further conduct human evaluations with 100 randomly sampled documents generated by each method. We ask five raters to evaluate the following metrics in a summary: Less Repeat (LR), Fluency (FC), Grammar Correct (GC) and the overall evaluation on readability and quality of summaries (RQ). The results are presented in Table 2. Our methods contribute significantly to improving readability of summaries, both semantically and syntactically correct, which allows readers read smooth and be easy to focus on the critical parts of sentences.

Model	R-1	R-2	R-L
Pointer Generator	39.53	17.28	36.38
SumGAN	39.92	**17.68**	36.71
NovelSum	**40.19**	17.38	**37.52**
Our Implementation	38.02	16.25	35.82
PredFusion	38.35	16.94	35.87
PredFusion (-Switch)	38.13	16.62	35.59
FeatFusion	**40.36**	**17.81**	**38.08**
FeatFusion (-Switch)	39.58	16.48	36.68
VoC Fusion	39.84	16.59	36.83

Table 1: ROUGE Evaluation on the CNN/Daily Mail test set. All our ROUGE F1 scores have a 95% confidence interval of at most 0.25 as reported by the official ROUGE script.

Model	RQ	LR	FC	GC
Ground Truth	8.00	8.27	8.33	8.20
Pointer Generator	6.26 / 6.76*	**6.86**	**6.76**	6.80
SumGAN	**6.67 / 6.79***	**6.87**	6.73	**7.47**
NovelSum	- / 6.35*	-	-	-
PredFusion	6.67	**7.37**	7.13	**7.67**
PredFusion (-Switch)	6.58	7.12	7.04	7.40
FeatFusion	**6.93**	7.31	**7.18**	7.40
FeatFusion (-Switch)	6.61	7.20	6.94	7.34
VoC Fusion	6.80	6.87	6.40	7.20

Table 2: Human Rating using 1-10 scoring scheme (10 is the best, 5 is acceptable).$*$ indicates the results reported by NovelSum (Kryscinski et al., 2018).

4 Conclusion

In this paper, we attempt to incorporate the external language model into the neural abstractive summarization model to enable the decoder of the sequence-to-sequence model to focus its capacity more on coping and fusing different parts in the source text by relying on the external language model to take care of language fluency. Experiments on the CNN/Daily Mail dataset show that our models outperform the state-of-the-art baselines in terms of automatic evaluation metrics, and meanwhile improve the readability of summaries by reducing repetitiveness, disfluency and grammatical problem by a large margin.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 93–98.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Wojciech Kryscinski, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1808–1817.

Chin-Yew Lin and Eduard H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative adversarial network for abstractive text summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 8109–8110.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1747–1759.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.