A Multi-task Learning Framework for Abstractive Text Summarization

Yao Lu[1,3] **Linqing Liu**[3] **Zhile Jiang**[4] **Min Yang**[5*] **Randy Goebel**[1,2]
[1]Alberta Machine Intelligence Institute [2]Department of Computing Science, University of Alberta
[3]David R. Cheriton School of Computer Science, University of Waterloo [4]Sichuan University
[5]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
{yao.lu, linqing.liu}@uwaterloo.ca jiangzhile.zz@gmail.com
min.yang@siat.ac.cn rgoebel@ualberta.ca

Abstract

We propose a Multi-task learning approach for Abstractive Text Summarization (*MATS*), motivated by the fact that humans have no difficulty performing such task because they have the capabilities of multiple domains. Specifically, *MATS* consists of three components: (i) a text categorization model that learns rich category-specific text representations using a bi-LSTM encoder; (ii) a syntax labeling model that learns to improve the syntax-aware LSTM decoder; and (iii) an abstractive text summarization model that shares its encoder and decoder with the text categorization and the syntax labeling tasks, respectively. In particular, the abstractive text summarization model enjoys significant benefit from the additional text categorization and syntax knowledge. Our experimental results show that *MATS* outperforms the competitors.[1]

Introduction

Abstractive text summarization aims to generate condensed and concise summaries that retain the salient information of a source text. The abstracted summaries potentially contain new phrases and sentences that don't appear in the source text. Inspired by the recent success of sequence-to-sequence (seq2seq) models in statistical machine translation (Bahdanau, Cho, and Bengio 2015), most abstractive summarization systems employ a seq2seq framework to generate summaries (Nallapati et al. 2016; See, Liu, and Manning 2017).

Despite progress, significant generation and syntax conforming challenges remain: (1) writing styles and words in different categories can significantly vary. But existing methods apply a uniform model to generate summaries for the source texts in different categories, which tend to generate trivial and generic summaries which easily under represent salient aspects of the source text. (2) syntactic information plays a crucial role in sentence generation, and enforcing syntactic conformance addresses issues like incomplete sentences. Despite its usefulness, the exploitation of syntax has received little attention in abstractive summarization.

Here we consider an integration of these aforementioned approaches to the noted challenges, and show measurable progress in how they improve abstractive summarization. Our framework consolidates these improvements by exploiting the recent successes of the encoder-decoder framework to generate abstractive summaries. While this standard framework is common in related approaches, we additionally propose extended regularizations using multi-task learning. Our encoder and decoder are regularized with the co-training required to perform additional text categorization and syntax annotation task, respectively. This co-training is not intended to maximize performance on these auxiliary tasks, but rather to compensate for the missing regularization requirement of text summarization in the standard framework. We also employ reinforcement learning to maximize long-term rewards of generation. Overall, the integration of multi-task learning approach in the framework provides significant improvements in abstractive summarization.

Methodology

Assume each input article $X = \{x_1, x_2, ..., x_n\}$ has corresponding reference summary $Y = \{y_1, y_2, ..., y_k\}$ and category label L, where n and k denote the length of input article and reference summary, respectively. Given an input article X, the abstractive summarization task tries to generate a summary $\hat{Y} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_m}\}$, where m denote the length of the generated summary. For the text categorization task, given an input article X, our objective is to predict the category label $\hat{L}$ for the input text. For syntax labeling, we have $Z = \{z_1, z_2, ..., z_m\}$ denoting the CCG supertag sequence for the corresponding summary $\hat{Y}$ of source text X.

Shared Bi-LSTM encoder: The summarization shares encoder with the text categorization task. Each word x in the source text is mapped through the embedding layer. Then, given the input word embedding v_i at time step i, the hidden state h_i can be updated with the previous hidden state h_{i-1} as $h_i = \text{Bi-LSTM}(h_{i-1}, v_i)$. In this way, we obtain the hidden states $H = \{h_1, ..., h_n\}$ for the source text X. For text categorization, we use the final state h_n as the representation of source text. h_n is then fed into a task-specific fully connected layer with *softmax* to predict the category label.

Shared LSTM decoder: LSTM decoder is shared by summarization and syntax labeling tasks. On each decoding step t, the decoder receives the input u_t (while training, u_t is the embedding of previous word of reference sum-

*Min Yang is correponding author.

[1]Supplementary material: https://github.com/yaolu/MATS

mary; at test time it is the embedding of previous word emitted by the decoder) and update its hidden state s_t as $s_t = \text{LSTM}(s_{t-1}, c_t, u_t)$, where c_t is the context vector at time step t. It can be computed follow the widely used attention mechanism (Bahdanau, Cho, and Bengio 2015). We then feed the concatenated vector $[c_t, s_t]$ to a linear function to produce the hidden vector of the decoder O_t. The generation probabilities of the t^{th} word (w) and CCG supertag (tag) can be written as the following:

$$P_t^{\text{w}} = P(y_t|Y_{1:t-1}; X) = \text{softmax}(U^w O_t + b^w) \quad (1)$$

$$P_t^{\text{tag}} = P(z_t|Y_{1:t-1}; X) = \text{softmax}(U^{tag} O_t + b^{tag}) \quad (2)$$

However, the pure generation model sometimes suffers from the out-of-vocabulary(OOV) generation issue and produces many "UNK" tokens in the summary. We adopt the copy mechanism (See, Liu, and Manning 2017) to alleviate this limitation. The generation probability $p_{\text{gen}} \in [0, 1]$ for time step t is calculated from the context vector c_t, the decoder state s_t, and the decoder input u_t:

$$P_{\text{gen}} = \sigma(V_c^T c_t + V_s^T s_t + V_u^T u_t + b_{gen}) \quad (3)$$

We then incorporate a switching pointer-generator network to use either word generator from fixed vocabulary or pointer copying OOV words from the source. We can get the final probability $P(w_j)$ of each token $\hat{y_t}$ in the summary.

Multi-task Learning: Our model consists of three subtasks, each has its own training objective. For text categorization, the objective is to minimize its cross entropy loss: $J_{\text{ml}}^{\text{text}}(\theta) = -\sum_{i=1}^{N} L_i log(\hat{L_i})$. For the summarization and syntax labeling subtasks, we employ the minimum negative log-likelihood estimation: $J_{\text{ml}}^{\text{sum}}(\theta) = -\sum^{T} \log(P_t^w)$ and $J_{\text{ml}}^{\text{syn}}(\theta) = -\sum^{T} \log(P_t^s)$. For the purpose of improving shared encoder and decoder, we train the three tasks simultaneously. The joint multi-task objective is minimized by:

$$J_{\text{ml}} = \lambda_1 J_{\text{ml}}^{\text{text}} + \lambda_2 J_{\text{ml}}^{\text{sum}} + \lambda_3 J_{\text{ml}}^{\text{syn}} \quad (4)$$

where $\lambda_1 = \lambda_2 = 0.45$ and $\lambda_3 = 0.1$ are hyper-parameters that determines weights of three objectives by performing grid search on validation set.

To maximize long-term rewards and alleviate the exposure bias, we also optimize directly for ROUGE-1 using policy gradient, and minimize the negative expected rewards: $J_{\text{rl}}^{\text{sum}}(\theta) = (r(\hat{y}) - r(y^s)) \sum^{n} \log p(y_t^s|Y_{1:t-1}^s; X)$, where $r(\hat{y})$ is the reward of greedy decoding generated sequence and $r(y^s)$ is the reward of sequence generated by sample among the vocabulary at each step. After pre-training the proposed model by minimizing joint ML objective Eq. 4, we switch the model to minimize mixed training objective: $J_{\text{mixed}} = \beta J_{\text{ml}} + (1 - \beta) J_{\text{rl}}^{\text{sum}}$, where $\beta = 0.1$ is a hyper-parameter.

Experiments

We evaluate our model on the CNN/Daily Mail Corpus, which comprises news stories paired with multi-sentences human generated summaries. For the text categorization task, the source webpage of each news story indicates the specific category of each story. For syntax annotation, the training data is annotated with CCG supertags, where each word has a corresponding dependency label of supertags.

We compare our model with several state-of-the-art methods including ABS (Nallapati et al. 2016), Lead-3 and SummaRuNNer[2] (Nallapati, Zhai, and Zhou 2017), PGC (See, Liu, and Manning 2017), DeepRL (Paulus, Xiong, and Socher 2018), and GANsum (Liu et al. 2018).

Methods	ROUGE (F1 Score)			Human
	1	2	L	Score
ABS	35.46	13.3	32.65	1.10
Lead-3*	39.2	15.7	35.5	-
SummaRuNNer*	39.6	16.2	35.3	-
PGC	39.53	17.28	36.38	3.38
DeepRL	39.87	15.82	36.90	4.48
GANsum	39.92	17.65	36.71	4.7
MATS w/o text	40.37	17.79	36.75	4.52
MATS w/o syntax	40.62	17.96	37.01	4.64
Our model (MATS)	**40.74**	**18.14**	**37.15**	**5.18**

Table 1: Experiment results

Quantitative Evaluation Table 1 shows that our model outperforms the baseline methods by a noticeable margin. To investigate the effect of each component of our model, we also perform the ablation test of MATS in terms of discarding text categorization (w/o text) and syntax generation (w/o syntax), respectively. Our model substantially outperforms the baseline methods by a noticeable margin.

Qualitative Evaluation We also evaluate the informativeness and fluency of the generated summaries by randomly select 50 examples from the test set. Two human evaluators are required to perform ranking of summaries by taking the above 2 factors into consideration, where 1 indicates the lowest level and 7 indicates the highest level. The experimental results based on human evaluation are summarized in Table 1. *MATS* achieves the best results.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Liu, L.; Lu, Y.; Yang, M.; Qu, Q.; Zhu, J.; and Li, H. 2018. Generative adversarial network for abstractive text summarization. In *AAAI*.

Nallapati, R.; Zhou, B.; dos Santos, C.; glar Gulçehre, Ç.; and Xiang, B. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Paulus, R.; Xiong, C.; and Socher, R. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

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

[2]The two models are not directly comparable to ours as they are based on an anonymized version of the dataset