Autoencoder as Assistant Supervisor: Improving Text Representation for Chinese Social Media Text Summarization

Shuming Ma\textsuperscript{1}, Xu Sun\textsuperscript{1,2}, Junyang Lin\textsuperscript{3}, Houfeng Wang\textsuperscript{1}
\textsuperscript{1}MOE Key Lab of Computational Linguistics, School of EECS, Peking University
\textsuperscript{2}Deep Learning Lab, Beijing Institute of Big Data Research, Peking University
\textsuperscript{3}School of Foreign Languages, Peking University
\{shumingma, xusun, linjunyang, wanghf\}@pku.edu.cn

Abstract

Most of the current abstractive text summarization models are based on the sequence-to-sequence model (Seq2Seq). The source content of social media is long and noisy, so it is difficult for Seq2Seq to learn an accurate semantic representation. Compared with the source content, the annotated summary is short and well written. Moreover, it shares the same meaning as the source content. In this work, we supervise the learning of the representation of the source content with that of the summary. In implementation, we regard a summary autoencoder as an assistant supervisor of Seq2Seq. Following previous work, we evaluate our model on a popular Chinese social media dataset. Experimental results show that our model achieves the state-of-the-art performances on the benchmark dataset.\textsuperscript{1}

1 Introduction

Text summarization is to produce a brief summary of the main ideas of the text. Unlike extractive text summarization (Radev et al., 2004; Woodsend and Lapata, 2010; Cheng and Lapata, 2016), which selects words or word phrases from the source texts as the summary, abstractive text summarization learns a semantic representation to generate more human-like summaries. Recently, most models for abstractive text summarization are based on the sequence-to-sequence model, which encodes the source texts into the semantic representation with an encoder, and generates the summaries from the representation with a decoder.

\textsuperscript{1}The code is available at https://github.com/lancopku/superAE

The contents on the social media are long, and contain many errors, which come from spelling mistakes, informal expressions, and grammatical mistakes (Baldwin et al., 2013). Large amount of errors in the contents cause great difficulties for text summarization. As for RNN-based Seq2Seq, it is difficult to compress a long sequence into an accurate representation (Li et al., 2015), because of the gradient vanishing and exploding problem.

Compared with the source content, it is easier to encode the representations of the summaries, which are short and manually selected. Since the source content and the summary share the same points, it is possible to supervise the learning of the semantic representation of the source content with that of the summary.

In this paper, we regard a summary autoencoder as an assistant supervisor of Seq2Seq. First, we train an autoencoder, which inputs and reconstructs the summaries, to obtain a better representation to generate the summaries. Then, we supervise the internal representation of Seq2Seq with that of autoencoder by minimizing the distance between two representations. Finally, we use adversarial learning to enhance the supervision. Following the previous work (Ma et al., 2017), We evaluate our proposed model on a Chinese social media dataset. Experimental results show that our model outperforms the state-of-the-art baseline models. More specifically, our model outperforms the Seq2Seq baseline by the score of 7.1 ROUGE-1, 6.1 ROUGE-2, and 7.0 ROUGE-L.

2 Proposed Model

We introduce our proposed model in detail in this section.

2.1 Notation

Given a summarization dataset that consists of $N$ data samples, the $i^{th}$ data sample $(x_i, y_i)$ con-
Figure 1: The overview of our model. The model consists of a sequence-to-sequence model and an autoencoder model. At the training stage, we use the autoencoder to supervise the sequence-to-sequence model. At the test stage, we use the sequence-to-sequence model to generate the summaries.

2.2 Supervision with Autoencoder

Figure 1 shows the architecture of our model. At the training stage, the source content encoder compresses the input content $x$ into the internal representation $z_t$ with a Bi-LSTM encoder. At the same time, the summary encoder compresses the reference summary $y$ into the representation $z_s$. Then both $z_t$ and $z_s$ are fed into an LSTM decoder to generate the summary. Finally, the semantic representation of the source content is supervised by the summary.

We implement the supervision by minimizing the distance between the semantic representations $z_t$ and $z_s$, and this term in the loss function can be written as:

$$L_S = \frac{\lambda}{N_h}d(z_t, z_s)$$  \hspace{1cm} (1)

where $d(z_t, z_s)$ is a function which measures the distance between $z_t$ and $z_s$. $\lambda$ is a tunable hyper-parameter to balance the loss of the supervision and the other parts of the loss, and $N_h$ is the number of the hidden unit to limit the magnitude of the distance function. We set $\lambda = 0.3$ based on the performance on the validation set. The distance between two representations can be written as:

$$d(z_t, z_s) = \|z_t - z_s\|_2$$  \hspace{1cm} (2)

2.3 Adversarial Learning

We further enhance the supervision with the adversarial learning approach. As shown in Eq. 1, we use a fixed hyper-parameter as a weight to measure the strength of the supervision of the autoencoder. However, in the case when the source content and summary have high relevance, the strength of the supervision should be higher, and when the source content and summary have low relevance, the strength should be lower. In order to determine the strength of supervision more dynamically, we introduce the adversarial learning. More specifically, we regard the representation of the autoencoder as the “gold” representation, and that of the sequence-to-sequence as the “fake” representation. A model is trained to discriminate between the gold and fake representations, which is called a discriminator. The discriminator tries to identify the two representations. On the contrary, the supervision, which minimizes the distance of the representations and makes them similar, tries to prevent the discriminator from making correct predictions. In this way, when the discriminator can distinguish the two representations (which means the source content and the summary has high relevance), the strength of supervision will be decreased, and when the discriminator fails to distinguish, the strength of supervision will be improved.

In implementation of the adversarial learning, the discriminator objective function can be written as:

$$L_D(\theta_D) = -\log P_{y_D}(y = 1|z_t)$$

$$-\log P_{y_D}(y = 0|z_s)$$  \hspace{1cm} (3)

where $P_{y_D}(y = 1|z)$ is the probability that the discriminator identifies the vector $z$ as the “gold” representation, while $P_{y_D}(y = 0|z)$ is the probability that the vector $z$ is identified as the “fake” representation, and $\theta_D$ is the parameters of the discrimi-
inator. When minimizing the discriminator objective, we only train the parameters of the discriminator, while the rest of the parameters remains unchanged.

The supervision objective to be against the discriminator can be written as:

\[
L_G(\theta_E) = -\log P_{\theta_D}(y = 0 | z_t) - \log P_{\theta_D}(y = 1 | z_s)
\] (4)

When minimizing the supervision objective, we only update the parameters of the encoders.

2.4 Loss Function and Training

There are several parts of the objective functions to optimize in our models. The first part is the cross entropy losses of the sequence-to-sequence and the autoencoder:

\[
L_{Seq2seq} = -\sum_{i=1}^{N} p_{Seq2seq}(y_i | z_s)
\] (5)

\[
L_{AE} = -\sum_{i=1}^{N} p_{AE}(y_i | z_t)
\] (6)

The second part is the L2 loss of the supervision, as written in Equation 1. The last part is the adversarial learning, which are Equation 3 and Equation 4. The sum of all these parts is the final loss function to optimize.

We use the Adam (Kingma and Ba, 2014) optimization method to train the model. For the hyper-parameters of Adam optimizer, we set the learning rate \(\alpha = 0.001\), two momentum parameters \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\) respectively, and \(\epsilon = 1 \times 10^{-8}\). We clip the gradients (Pascanu et al., 2013) to the maximum norm of 10.0.

3 Experiments

Following the previous work (Ma et al., 2017), we evaluate our model on a popular Chinese social media dataset. We first introduce the datasets, evaluation metrics, and experimental details. Then, we compare our model with several state-of-the-art systems.

3.1 Dataset

Large Scale Chinese Social Media Text Summarization Dataset (LCSTS) is constructed by Hu et al. (2015). The dataset consists of more than 2,400,000 text-summary pairs, constructed from a famous Chinese social media website called Sina Weibo. It is split into three parts, with 2,400,591 pairs in PART I, 10,666 pairs in PART II and 1,106 pairs in PART III. All the text-summary pairs in PART II and PART III are manually annotated with relevant scores ranged from 1 to 5. We only reserve pairs with scores no less than 3, leaving 8,685 pairs in PART II and 725 pairs in PART III. Following the previous work (Hu et al., 2015), we use PART I as training set, PART II as validation set, and PART III as test set.

3.2 Evaluation Metric

Our evaluation metric is ROUGE score (Lin and Hovy, 2003), which is popular for summarization evaluation. The metrics compare an automatically produced summary with the reference summaries, by computing overlapping lexical units, including unigram, bigram, trigram, and longest common subsequence (LCS). Following previous work (Rush et al., 2015; Hu et al., 2015), we use ROUGE-1 (unigram), ROUGE-2 (bi-gram) and ROUGE-L (LCS) as the evaluation metrics in the reported experimental results.

3.3 Experimental Details

The vocabularies are extracted from the training sets, and the source contents and the summaries share the same vocabularies. In order to alleviate the risk of word segmentation mistakes, we split the Chinese sentences into characters. We prune the vocabulary size to 4,000, which covers most of the common characters.

We tune the hyper-parameters based on the ROUGE scores on the validation sets. We set the word embedding size and the hidden size to 512, and the number of LSTM layers is 2. The batch size is 64, and we do not use dropout (Srivastava et al., 2014) on this dataset. Following the previous work (Li et al., 2017), we implement the beam search, and set the beam size to 10.

3.4 Baselines

We compare our model with the following state-of-the-art baselines.

- **RNN and RNN-cont** are two sequence-to-sequence baseline with GRU encoder and decoder, provided by Hu et al. (2015). The difference between them is that RNN-context has attention mechanism while RNN does not.
- **RNN-dist** (Chen et al., 2016) is a distraction-based neural model, which the attention
### 3.5 Results

For the purpose of simplicity, we denote our supervision with autoencoder model as superAE. We report the ROUGE F1 score of our model and the baseline models on the test sets.

Table 1 summarizes the results of our superAE model and several baselines. We first compare our model with Seq2Seq baseline. It shows that our superAE model has a large improvement over the Seq2Seq baseline by 7.1 ROUGE-1, 6.1 ROUGE-2, and 7.0 ROUGE-L, which demonstrates the efficiency of our model. Moreover, we compare our model with the recent summarization systems, which have been evaluated on the same training set and the test sets as ours. Their results are directly reported in the referred articles. It shows that our superAE outperforms all of these models, with a relative gain of 2.2 ROUGE-1, 1.8 ROUGE-2, and 2.0 ROUGE-L. We also perform ablation study by removing the adversarial learning component, in order to show its contribution. It shows that the adversarial learning improves the performance of 1.5 ROUGE-1, 0.7 ROUGE-2, and 1.0 ROUGE-L.

We also give a summarization examples of our model. As shown in Table 3, the SeqSeq model captures the wrong meaning of the source content, and produces the summary that “China United Airlines exploded in the airport”. Our superAE model captures the correct points, so that the generated summary is close in meaning to the reference summary.

#### 3.6 Analysis of text representation

We want to analyze whether the internal text representation is improved by our superAE model. Since the text representation is abstractive and hard to evaluate, we translate the representation into a sentiment score with a sentiment classifier, and evaluate the quality of the representation by means of the sentiment accuracy.

We perform experiments on the Amazon Fine Foods Reviews Corpus (McAuley and Leskovec, 2013). The Amazon dataset contains users’ rating labels as well as the summary for the reviews, making it possible to train a classifier to predict the sentiment labels and a seq2seq model to generate summaries. First, we train the superAE model and
4 Related Work

Rush et al. (2015) first propose an abstractive based summarization model, which uses an attentive CNN encoder to compress texts and a neural network language model to generate summaries. Chopra et al. (2016) explore a recurrent structure for abstractive summarization. To deal with out-of-vocabulary problem, Nallapati et al. (2016) propose a generator-pointer model so that the decoder is able to generate words in source texts. Gu et al. (2016) also solved this issue by incorporating copying mechanism, allowing parts of the summaries are copied from the source contents. See et al. (2017) further discuss this problem, and incorporate the pointer-generator model with the coverage mechanism. Hu et al. (2015) build a large corpus of Chinese social media short text summarization, which is one of our benchmark datasets. Chen et al. (2016) introduce a distraction based neural model, which forces the attention mechanism to focus on the difference parts of the source inputs. Ma et al. (2017) propose a neural model to improve the semantic relevance between the source contents and the summaries.

Our work is also related to the sequence-to-sequence model (Cho et al., 2014), and the autoencoder model (Bengio, 2009; Liou et al., 2008, 2014). Sequence-to-sequence model is one of the most successful generative neural model, and is widely applied in machine translation (Sutskever et al., 2014; Jean et al., 2015; Luong et al., 2015), text summarization (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016), and other natural language processing tasks. Autoencoder (Bengio, 2009) is an artificial neural network used for unsupervised learning of efficient representation. Neural attention model is first proposed by Bahdanau et al. (2014).

5 Conclusion

We propose a novel model, in which the autoencoder is a supervisor of the sequence-to-sequence model, to learn a better internal representation for abstractive summarization. An adversarial learning approach is introduced to further improve the supervision of the autoencoder. Experimental results show that our model outperforms the sequence-to-sequence baseline by a large margin, and achieves the state-of-the-art performances on a Chinese social media dataset.

Acknowledgements

Our work is supported by National Natural Science Foundation of China (No. 61433015, No. 61673028), National High Technology Research and Development Program of China (863 Program, No. 2015AA015404), and the National Thousand Young Talents Program. Xu Sun is the corresponding author of this paper.
References


Shuming Ma, Xu Sun, Wei Li, Suijian Li, Wenjie Li, and Xuancheng Ren. 2018. Query and output: Generating words by querying distributed word representations for paraphrase generation. In *NAACL 2018*.


