Computers, Materials & Continua
DOI:10.32604/cmc.2021.017441
Article

Tech Science Press

A Semantic Supervision Method for Abstractive Summarization

Sunqiang Hu[1], Xiaoyu Li[1], Yu Deng[1,*], Yu Peng[1], Bin Lin[2] and Shan Yang[3]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China
[2]School of Engineering, Sichuan Normal University, Chengdu, 610066, China
[3]Department of Chemistry, Physics, and Atmospheric Sciences, Jackson State University, Jackson, MS, 39217, USA
[*]Corresponding Author: Yu Deng. Email: 411180435@qq.com
Received: 30 January 2021; Accepted: 17 March 2021

Abstract: In recent years, many text summarization models based on pre-training methods have achieved very good results. However, in these text summarization models, semantic deviations are easy to occur between the original input representation and the representation that passed multi-layer encoder, which may result in inconsistencies between the generated summary and the source text content. The Bidirectional Encoder Representations from Transformers (BERT) improves the performance of many tasks in Natural Language Processing (NLP). Although BERT has a strong capability to encode context, it lacks the fine-grained semantic representation. To solve these two problems, we proposed a semantic supervision method based on Capsule Network. Firstly, we extracted the fine-grained semantic representation of the input and encoded result in BERT by Capsule Network. Secondly, we used the fine-grained semantic representation of the input to supervise the fine-grained semantic representation of the encoded result. Then we evaluated our model on a popular Chinese social media dataset (LCSTS), and the result showed that our model achieved higher ROUGE scores (including R-1, R-2), and our model outperformed baseline systems. Finally, we conducted a comparative study on the stability of the model, and the experimental results showed that our model was more stable.

Keywords: Text summarization; semantic supervision; capsule network

1 Introduction

The goal of text summarization is to deliver important information in the source text with a small number of words. In the current era of information explosion, it is an undeniable fact that text information floods the Internet. Hence, it is necessary for us to apply text summarization which can help us obtain useful information from the source text. With the rapid development of artificial intelligence, automatic text summarization was proposed, that is, computers can aid people in complex text summarization. By using machine learning, deep learning, and other methods, we can get a general model for automatic text summarization, which can replace humans to extract summary from source text.

Automatic text summarization is usually divided into two categories according to the implementation method: extractive summarization and abstractive summarization. And Extractive summarization is to extract some sentences containing key information from the source text and combine them to a summarization, while abstractive summarization is to compress and refine the information of source text to generate a new summarization. Compared with extractive summarization, abstractive summarization is more innovative, because machines can generate summary contents that are more informative and attractive. Abstractive text summarization models are usually based on a sequence-to-sequence model [1]. It contains two parts: the encoder and decoder. The encoder encodes the input as a fixed-length context vector which contains important information of the input text, and the decoder decodes the vector into the output we desire. In the early days, we will choose RNN or LSTM [2] as the encoder-decoder structure of seq2seq, and use the last hidden unit of RNN or LSTM as the context vector of the decoder.

BERT [3] is a pre-trained language model which is trained in advance through a large amount of unsupervised data. With a good capability of contextual semantic representation, BERT has achieved very good performance in many tasks of NLP. However, it is not suitable to complete generative tasks for lack of the decoder structure. Dong et al. [4] proposed a Unified Pre-trained Language Model (UNILM), whose submodule seq2seqLM could complete the task of natural language generation by modifying BERT's mask matrix. BERT can encode each word accurately according to the context, but it lacks a fine-grained semantic representation of the entire input text, which results in semantic deviations between the result encoded by BERT and the original input text. The traditional seq2seq model does not perform well in text summarization, so we consider using the pre-trained model BERT to improve the actual effect of the text summarization. However, BERT has its flaws mentioned above. Therefore, we hope to overcome the defects by applying some methods and improve the effectiveness of the text summarization model based on BERT.

Nowadays, Neural Network has been applied to many fields [5,6], and automatic text summarization is one of its hot research. In this paper, according to the idea of seq2seqLM, we modified the mask matrix of BERT and used BERT-base to complete abstractive summarization. To reduce semantic deviations, we introduced a semantic supervision method based on Capsule Network [7] into our model. Following previous work, we evaluated our proposed model on the LCSTS dataset [8], the experimental results showed that our model is superior to the baseline system, and the proposed semantic supervision method can indeed improve the effectiveness of BERT.

The remainder of this paper is organized as follows. The related work will be discussed in Section 2. The proposed model will be presented in Section 3. Details of the experiment will be explained in Section 4. Comparison and discussion of experimental results will be made in Section 5. Conclusions and Future work will be drawn in Section 6.

2 Related Works

2.1 Seq2seq Model

The research on abstractive summarization mainly depends on the seq2seq model proposed by Cho et al. [1], which solves the length inequality of input and output in generative tasks. The seq2seq model contains two parts: encoder and decoder. The encoder encodes the input into a context vector C, and the decoder decodes the output by C. The Seq2seq model was originally used for Neural Machine Translation (NMT), and firstly proposed by Rush et al. [9] based on attention mechanism [10] for abstractive summarization, and it proved to have good performance.

2.2 Pre-Trained Model and BERT

Pre-trained language model has become an important technology in NLP field in recent years. The main idea is that the model's parameters are no longer randomly initialized, but trained in advance by some tasks (such as Language Model) and large-scale text corpus. Then they are fine-tuned on the small dataset of specific tasks, and it makes it easy to tarin a model. The early pre-trained Language Model is Embeddings from Language Model (ELMo) [11], which can complete the feature extraction by bidirectional LSTM and fine-tune the downstream tasks. A Generative Pre-Training Language Model (GPT) can achieve very good performance by replacing LSTM with Transformer [12] in the text generation task. Based on GPT, Devlin et al. [3] considered using bidirectional Transformer and higher quality large-scale dataset for pre-training and obtained a better pre-trained language model BERT.

Liu et al. [13] proposed BERTSum for extractive summarization, a simple variant of BERT, and the model outperformed baseline on the CNN/DailyMail dataset. Later, Liu et al. [13] joined the decoder structure based on BERTSum to complete abstractive summarization and conducted experiments on the previous dataset. The experimental results showed that their model was superior to the previous model in both extraction summarization and abstractive summarization. The goal of UNILM proposed by Li et al. [4] is to adapt BERT to generative tasks, which is the same as that of Masked Sequence to Sequence Pre-training Model (MASS) proposed by Song et al. [14]. But UNILM is more succinct, sticking to BERT's idea and only using encoders to complete various NLP tasks. The UNILM is trained based on three objectives: Unidirectional LM (left-to-right and right-to-left), Bidirectional LM, and seq2seqLM. Seq2seqLM can complete abstractive summarization. It defines the source text as the first sentence and the corresponding summary as the second sentence. The first sentence is encoded by Bidirectional LM, and the second sentence is encoded by Unidirectional LM (left-to-right).

2.3 Semantic Supervision and Capsule Network

Ma et al. [15] proposed a method to improve semantic relevance in seq2seq model. By calculating the cosine similarity between the semantic vector of the source text and the summary, we can get the measure of semantic relevance between them. The larger the cosine value is, the more relevant they are, and the negative value of the cosine similarity is added to the loss function to maximize the semantic relevance between them. At the same time, Ma et al. [16] also proposed an autoencoder as an assistant supervisor method to improve the text representation. By minimizing the L2 distance between the summary encoder vector and the source text encoder vector, we can supervise the semantic representation of the source text and improve the semantic representation of the source text.

In 2017, Sabour et al. [7] proposed a new neural network structure called Capsule Network. The input and output of Capsule Network are all in the form of vectors, and the results of image classification experiments showed that Capsule Network has a strong ability of feature aggregation. Zhao et al. [17] proposed a model based on Capsule Network to do text classification. As a result, the model performed better than the baseline system in the experiment.

Based on the methods mentioned above, we complete abstractive summarization by adopting the idea of seq2seqLM, and added the semantic supervision method into the model. We conducted relevant experiments on the Chinese dataset LCSTS [8], and analyzed the experimental results.

3 Proposed Model

3.1 BERT for Abstractive Summarization

Our model structure is shown in Fig. 1, and it is composed of four parts. Embedding Layer is responsible for transforming the input token into a vector representation. Transformer Layer is responsible for encoding the token vector representation according to the context information. Output Layer is used to parse the encoded result of Transformer Layer. And the last part is the Semantic Supervision module proposed by us, which is responsible for supervising the semantic encoding of Transformer Layer.

Embedding Layer

BERT's embedding layer contains Token Embedding, Segment Embedding and Position Embedding. Token Embedding is the vector representation of tokens, which is obtained by looking up the embedding matrix with token Id. Segment Embedding is used to express whether the current token comes from the first segment or the second segment. Position Embedding is the position vector of the current token. Fig. 1 shows Embedding Layer of BERT. The input representation follows that of BERT. We added a special token ([CLS]) at the beginning of input, and added a special token ([SEP]) at the end of every segment. $T = \{T_1, T_2, \ldots, T_n\}$ represents the token sequence of the source text, and $S = \{S_1, S_2, \ldots, S_n\}$ represents the token sequence of the summary. We got the input $X = \{[\text{CLS}], \text{T}_1, \text{T}_2, \ldots, \text{T}_n, [\text{SEP}], \text{S}_1, \ldots, \text{S}_m, [\text{SEP}]\}$ of the model by splicing T, S and special token. By summing corresponding Token Embedding, Position Embedding and Segment Embedding, we can get a vector representation of each input token.

Transformer Layer

Transformer Layer consists of N Transformer Blocks which share the same structure but have different parameters to be trained. Transformer was originally proposed by Vaswani et al. [12], but only the Encoder part of Transformer is used in BERT. The reason why BERT can perform well in many NLP tasks is that it depends on a large amount of unsupervised data and the excellent semantic encoding capability of Transformer.

The input of seq2seqLM is the same as that of BERT, but the main difference is that seq2seqLM changes the mask matrix of multi-head attention in Transformer. As shown on the left of Fig. 2, the source text's tokens can attend to each other from both directions (left-to-right and right-to-left), while every token of the summary can only attend to its left context (including itself) and all tokens in the source text. The mask matrix is designed as follows [4]:

$$M_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{1}$$

The element of the mask matrix is 0, which means the ith token can attend to the jth token. In contrast, the element is $-\infty$, which means the ith token can't attend to the jth token. On the right of Fig. 2, we showed the self-attention mask matrix M in Eq. (1), which is designed for the text summarization. The left part of M is set 0 so that all tokens can attend to the source text token. Our goal is to predict the summary, and attention from the source text to the summary is unnecessary, we set the upper right elements $-\infty$. On the bottom right side, we set its lower triangular matrix elements 0, and other elements $-\infty$, which prevents the current tokens of the summary from paying attention to the tokens after it.

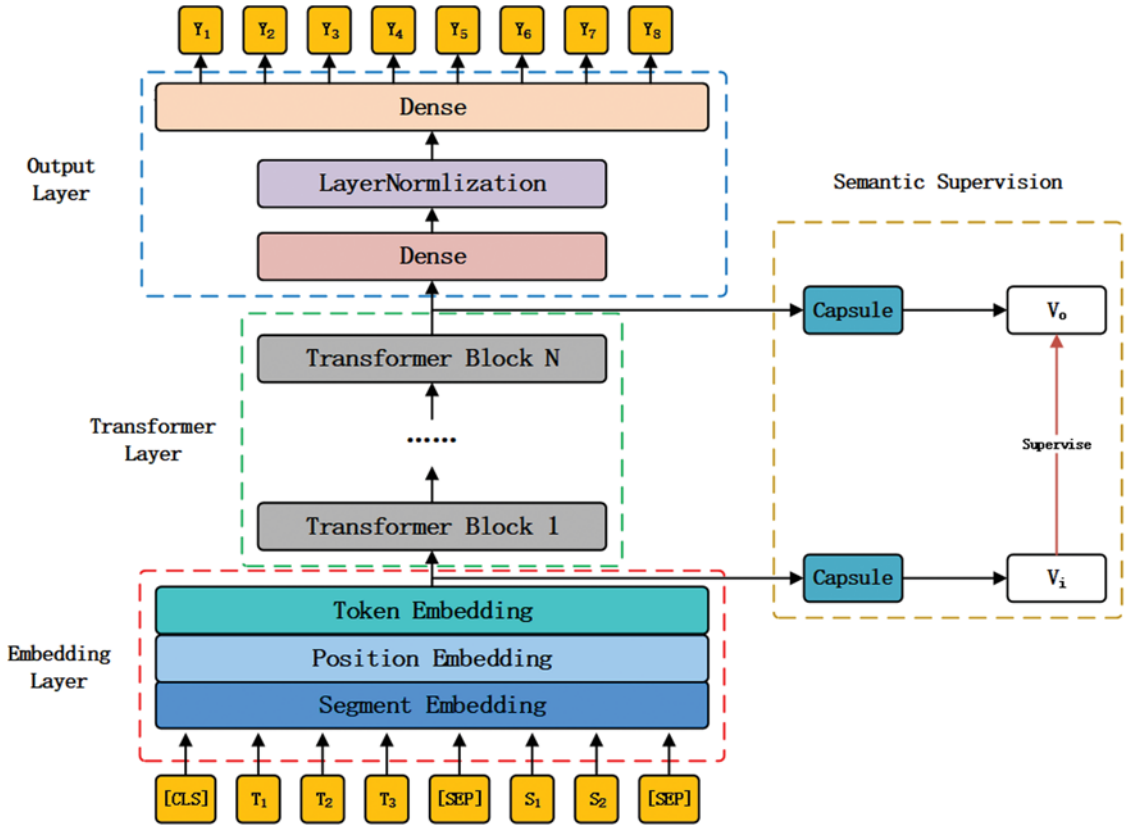

Figure 1: An overview of our model. Its main body (on the left) is composed of embedding layer, transformer layer, and output layer. Based on the main body, it contains a semantic supervision module (on the right)

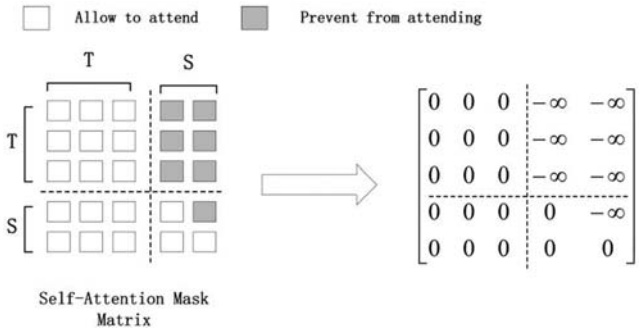

Figure 2: The overview of self-attention mask matrix

The output of Embedding Layer is defined as $T^0 = \{X_1, X_2, \ldots, X_n\}$, where X_i represents the vector representation of the ith token and n represents the length of the input sequence. We abbreviated the output of the lth Transformer block as: $T^l = Transformer_l(T^{l-1})$. In each Transformer Block, by aggregating multiple self-attention heads, we can get the output of the

current multi-head attention. For the lth Transformer block, the output A_l of the multi-head attention is computed as follows:

$$A^l = Concat(head_1, head_2, \ldots, head_h) W^o \tag{2}$$

where $head_i = soft\max\left(\frac{QK^T}{\sqrt{d_k}} + M\right) V$

and $Q = T^{l-1} W_i^Q, \quad K = T^{l-1} W_i^K, \quad V = T^{l-1} W_i^V$

$T^{l-1} \in R^{n \times d_h}$ is the output of the $(l-1)$th Transformer Block, where n is the length of the input sequence and d_h is the embedding size. W_i^Q, W_i^K, $W_i^V \in R^{d_h \times d_k}$ and $W^o \in R^{d_h \times d_h}$ are the linearly projected matrices where $d_k = d_h/h$ and d_h is the number of parallel attention heads. $M \in R^{n \times n}$ is the mask matrix in Eq. (1).

Output Layer

We took the output of the last Transformer Block as the input of Output Layer. Output Layer consists of three parts: two full connection layers and one Layer Normalization.

The first full connection layer is used to add nonlinear operations to BERT's output, and we use GELU as the activation function, which is widely used in BERT. In Eq. (3), T^N is the output of the last Transformer Block, W_1 is the matrix to be trained, b_1 is the value of bias, and O_1 is the output of the first full connection layer.

$$O_1 = GELU(W_1 T^N + b_1) \tag{3}$$

Different from Batch Normalization [18], Layer Normalization [19] does not depend on batch size and the length of the input sequence. Adding Layer Normalization can avoid gradient disappearance. In Eq. (4), $LN(*)$ is Layer Normalization and O_2 is the output of $LN(*)$.

$$O_2 = LN(O_1) \tag{4}$$

The second full connection layer is used to parse the output, which contains $n \times I$ (n is the length of output and I is the size of vocabulary) units, and we use softmax as the activation function. The softmax function is commonly used in multi-classification, and it map the output of multiple neurons to the interval (0, 1). Predicting a word is equivalent to a multi-classification task. In Eq. (5), W_3 is the matrix to be trained, b_3 is the value of bias, and O_3 is the final output of our model.

$$O_3 = \text{softmax}(W_3 O_2 + b_3) \tag{5}$$

3.2 Semantic Supervision Based on Capsule Network

For lack of fine-grained semantic representation in BERT, it can't produce high-quality summaries when it was applied to text summarization. And there are semantic deviations between the original input and the encoded result passed multi-layer encoder. We hope to improve these problems by adding semantic supervision based on Capsule Network. The implementation of semantic supervision is shown on the right side of Fig. 1. At the training stage, we took the result of Token Embedding as the input of Capsule Network and got the semantic representation V_i of the input. At the same time, we did the same operation for the output of the last Transformer Block to get the semantic representation V_o of the output. We implemented the semantic supervision

by minimizing the distance $d(V_i, V_o)$ between the semantic representation V_i and V_o. $d(V_i, V_o)$ is calculated as Eq. (6).

$$d(V_i, V_o) = \| V_i - V_o \|_2 \tag{6}$$

Ma et al. [15] directly took the input and output results of the model as semantic representations, which had low generalization capability. So we added a Capsule Network [7] which is capable of high-level feature clustering so as to extract semantic features. The Capsule Network uses vectors as input and output, and vector has a good representational capability, such as using vectors to represent words in word2vec. Of course, our experiment also showed that Capsule Network performed better than LSTM [2] and GRU [20]. We define a set of input vectors $u = \{u_1, u_2, \ldots, u_n\}$, and the output of Capsule Network is $v = \{v_1, v_2, \ldots, v_n\}$. The output of Capsule Network is calculated as follows:

$$u_{j|i} = W_{ij} u_i \tag{7}$$

$$b_{ij} = u_{j|i} v_j \tag{8}$$

$$c_{ij} = softmax(b_{ij}) \tag{9}$$

$$s_j = \sum_{i=1}^{n} c_{ij} u_{j|i} \tag{10}$$

$$v_j = squash(s_j) = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \tag{11}$$

It can be seen from Eq. (8) that the calculation of b_{ij} requires v_j, but v_j is the final output, so it is impossible to calculate b_{ij} directly. b_{ij} is usually given an initial value and computed iteratively. Based on this idea, Sabour et al. [7] proposed a Dynamic Routing algorithm in their paper.

We took the output of Embedding layer $X = \{X_1, X_2, \ldots, X_n\}$ as the input $u = \{u_1, u_2, \ldots, u_n\}$ of Capsule Network and got the output $v = \{v_1, v_2, \ldots, v_n\}$ where $X \in R^{n \times d_h}$ (n is the length of the input sequence and d_h is the embedding size). Each vector v_i in v represents a property, and the length of the vector represents the probability that the property exists. We calculated the norm of each vector in v to form a new vector as shown in Eq. (12), and V_i is the fine-grained semantic representation of the input X. Similarly, we regarded the output $T^N \in R^{n \times d_h}$ of BERT as the input $u' = \{u'_1, u'_2, \ldots, u'_n\}$, and got the output $v' = \{v'_1, v'_2, \ldots, v'_n\}$ by Capsule Network. By calculating the norm of each vector in v', we got a new vector as shown in Eq. (13), and V_o is the fine-grained semantic representation of the BERT's output.

$$V_i = \{|v_1|, |v_2|, \ldots, |v_n|\} \tag{12}$$

$$V_o = \{|v'_1|, |v'_2|, \ldots, |v'_n|\} \tag{13}$$

We found that the longer the input sequence is, the larger the semantic deviations are. So we use different intensity semantic supervision for different lengths of the input. We controlled the intensity of supervision by the parameter λ in Eq. (14) where l_s is the length of the input sequence. The longer the input sequence is, the larger the supervision intensity is, and the shorter the input sequence is, the lower the supervision intensity is.

$$\lambda = \frac{l_s}{1 + l_s} \tag{14}$$

The loss function of Semantic Supervision can be written as follows:

$$L_s = \lambda d(V_i, V_o) \tag{15}$$

3.3 Loss Function and Training

There are two loss functions in our model that need to be optimized. The first one is the categorical cross-entropy loss in Eq. (16), where N is the number of all samples, $y \in R^n$ is the true label of the input sample, $\hat{y} \in R^n$ is the corresponding prediction label, D is the sample set, n is the length of summary and m is the vocabulary size. The other one is the semantic supervision loss defined in Eq. (15). Our objective is to minimize the loss function in Eq. (17).

$$L_c = -\frac{1}{N}\sum_{y\in D}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} log \hat{y}_{ij} \tag{16}$$

$$L = L_c + L_s \tag{17}$$

During training, we used Adam optimizer [21] with the setting: learning rate $\alpha = 1 \times 10^{-5}$, two momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1 \times 10^{-8}$.

4 Experiments

In this section, we will introduce our experiments in detail, including dataset, evaluation metric, experiment setting and baseline systems.

Table 1: Statistics of different datasets of LCSTS

Dataset	pairs	Scores >= 3
PART I	2400591	–
PART II	10666	8685
PART III	1106	725

4.1 Dataset

We conducted experiments on LCSTS dataset [8] to evaluate the proposed method. LCSTS is a large-scale Chinese short text summarization dataset collected from Sina Weibo, which is a famous social media website in China. As shown in Tab. 1, it consists of more than 2.4 million pairs (source text and summary) and is split into three parts. PART I includes 2,400,591 pairs, PART II includes 10,666 pairs, and PART III includes 1,106 pairs. Besides, the pairs of PART II and PART III also have manual scores (according to the relevance between the source text and summary) ranging from 1 to 5. Following the previous work [8], we only chose pairs with scores no less than 3 and used PART I as the training set, PART II as the validation set, and PART III as the test set.

4.2 Evaluation Metric and Experiment Setting

We used the ROUGE scores [22] to evaluate our summarization model which has been widely used for text summarization. They can measure the quality of the summary by computing the overlap between the generated summary and the reference summary. Following the previous

work [8], we used ROUGE-1 (1-gram), ROUGE-2 (bigrams), and ROUGE-L (longest common subsequence) scores as the evaluation metric of the experimental results.

We used the Chinese glossary of BERT-base, which contains 21,128 characters, but the number we counted all the characters in PART I of LCSTS is 10,728. To reduce the computation, we only used the characters of the intersection between them, including 7,655 characters. In our model, we used the default embedding size 768 of BERT-base, the number of heads $h = 12$, and the number of Transformer blocks $N = 12$. For Capsule network, we set the number of output capsules to 50 and the output dimension to 16, and the number of routes to 3. We set the batch size to 16, and we used Dropout [23] in our model. Our model was trained on a single NVIDIA 2080Ti GPU. Following the previous work [24], we implemented the Beam Search and set the beam size to 3.

4.3 Baseline Systems

We have compared the proposed model with the following model's ROUGE score, and we would briefly introduce them next.

RNN and RNN-context [8] are two seq2seq baseline models. The former uses GRU as encoder and decoder. Based on that, the latter adds attention mechanism.

CopyNet [25] is the attention-based seq2seq model with the copy mechanism. The copy mechanism allows some tokens of the generated summary to be copied from the source content and it can effectively improve the problem of abstractive summarization with repeated words.

DRGD [26] is a seq2seq-based model with a deep recurrent generative decoder. The model combines the decoder with a variational autoencoder and uses a recurrent latent random model to learn latent structure information implied in the target summaries.

WEAN [27] is a novel model based on the encoder-decoder framework and its full name is Word Embedding Attention Network. The model generates the words by querying distributed word representations, hoping to capture the meaning of the corresponding words.

Seq2Seq + superAE [16] is a seq2seq-based model with an assistant supervisor. The assistant supervisor uses the representation of the summary to supervise that of the source content. And the model uses the autoencoder as an assistant supervisor. Besides, to determine the strength of supervision more dynamically, Adversarial Learning is introduced in the model.

Table 2: ROUGE scores of our model and baseline systems on LCSTS (W: word level; C: character level)

Models	ROUGE-1	ROUGE-2	ROUGE-3
RNN(W) [8]	17.7	8.5	15.8
RNN(C) [8]	21.5	8.9	18.6
RNN-context(W) [8]	26.8	16.1	24.1
RNN-context(C) [8]	29.9	17.4	27.2
CopyNet(W) [25]	35.0	22.3	32.0
CopyNet(C) [25]	34.4	21.6	31.3
DRGD(C) [26]	37.0	24.2	34.2
WEAN(C) [27]	37.8	25.0	35.2
Seq2seq + superAE(C) [16]	39.2	26.0	36.2
BERT-seq2seqLM(C) (our impl.)	39.84	25.47	34.62
+**SSC**(C) (this paper)	**40.63**	**26.4**	35.75

5 Results and Discussion

For clearer clarification, we named the BERT with the modified mask matrix as BERT-seq2seqLM, and denote our model with semantic supervision based on Capsule Network as **SSC**.

After we compared our model with baseline systems, the experimental results of these models on LCSTS datasets are shown in Tab. 2. Firstly, we compared our model with BERT-seq2seqLM, and it proved SSC outperformed BERT-seq2seqLM in the scores of ROUGE-1, ROUGE-2, and ROUGE-L. And it indicated that the semantic supervision method can improve the generation effect of Bert-seq2seqLM. Moreover, we compared the ROUGE scores of our model with the recent summarization systems and it showed that our model outperformed the baseline systems, and achieved higher scores on ROUGE-1 and ROUGE-2, while it was slightly lower than the baseline on ROUGE-L.

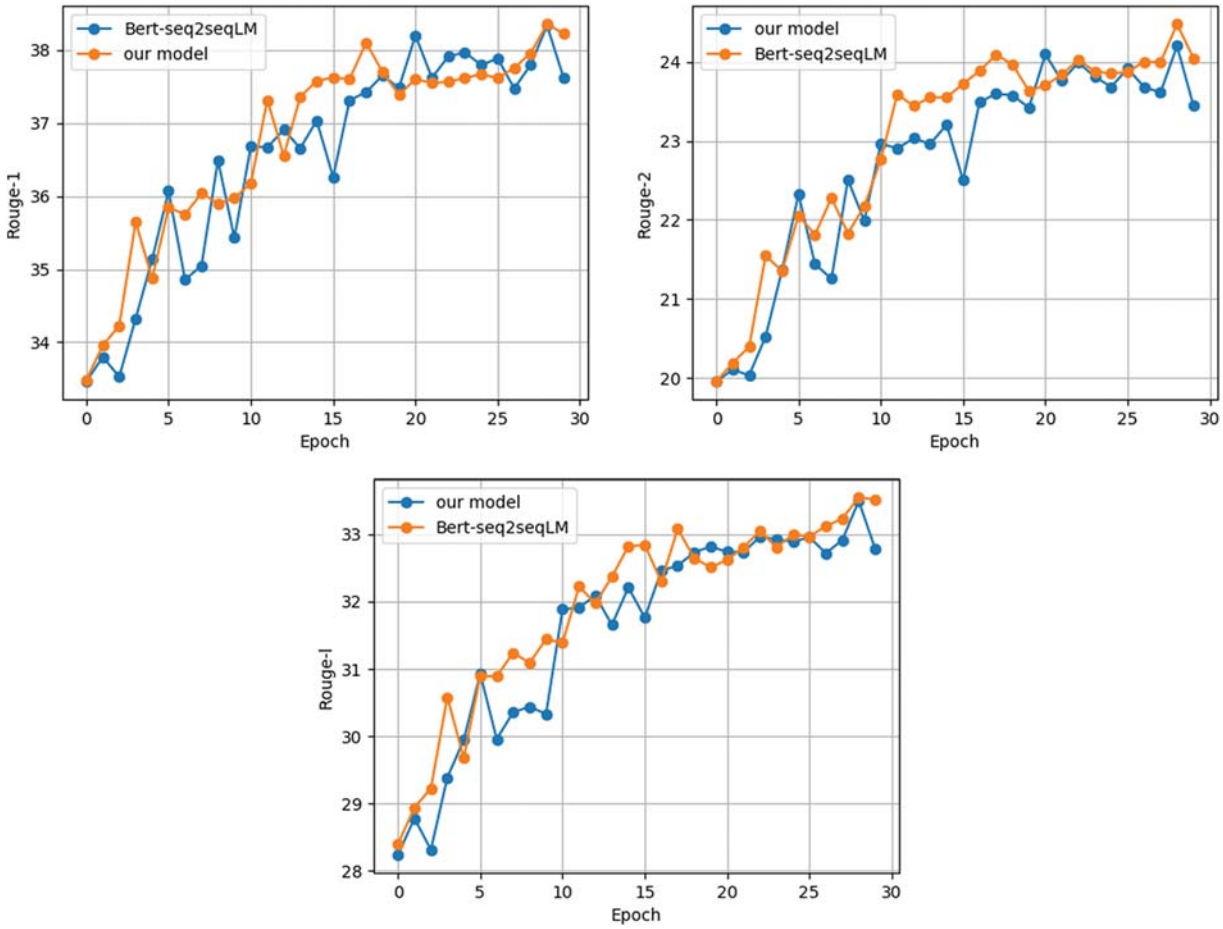

Figure 3: ROUGE scores curve of BERT-seq2seqLM and our model under different epoch training (including ROUGE-1, ROUGE-2, ROUGE-L scores curve)

In addition, we also compared the ROUGE scores of models under different epochs, as shown in Fig. 3. It respectively contains the scores of ROUGE-1, ROUGE-2, and ROUGE-L of the models under different epochs. From the three subgraphs, we can see that the training effect of

BERT-seq2seqLM is more stable and the overall evaluation score is higher after adding semantic supervision.

As for semantic supervision, in addition to Capsule Network, we also tried to use LSTM and GRU. However, after comparative experiments, we found that Capsule Network was more suitable. As shown in Tab. 3, we can see that the ROUGE-1, ROUGE-2 and ROUGE-L scores of the semantic supervision based on LSTM were higher than the BERT-seq2seqLM without the introduction of the semantic supervision. And the semantic supervision based on GRU and Capsule Network were also better than BERT-seq2-seqLM. Therefore, by experimental comparison, it is very necessary to introduce the semantic supervision method in BERT-seq2seqLM to improve the problem of fine-grained semantic representation. And the best improvement can be achieved by using Capsule Network for semantic supervision.

Table 3: ROUGE scores of the semantic supervision network with different structures on the LCSTS

Models	ROUGE-1	ROUGE-2	ROUGE-3
BERT-seq2seqLM	39.84	25.47	34.62
+LSTM	40.19	25.79	35.14
+GRU	40.34	26.0	35.22
+Capsule	40.63	26.4	35.75

Table 4: Some generated summary examples on the LCSTS test dataset

Article 1: 2007 年乔布斯向人们展示 iPhone 并宣称“它将会改变世界”，还有人认为他在夸大其词，然而在 8 年后，以 iPhone 为代表的触屏智能手机已经席卷全球各个角落。未来，智能手机将会成为“真正的个人电脑”，为人类发展做出更大的贡献。
In 2007, Steve Jobs showed people the iPhone and declared, “it’s going to change the world”. Some thought he was exaggerating. But eight years later, touch-screen smartphones like the iPhone have taken over every corner of the globe. In the future, smartphones will become “real personal computers”, and make greater contributions to human development.
Reference: 经济学人：智能手机将成为“真正的个人电脑”。
Economists said smartphones will become “real personal computers”.
BERT-seq2seqLM: iPhone 将成为个人电脑。
iPhone will become real personal computers.
BERT-seq2seqLM+SSC: 未来智能手机将成为“真正的个人电脑”。
In the future, smartphones will become “real personal computers”.
Article 2: 马克.库班，曾在大学打过四年橄榄球，1983 年创立计算机资讯公司 Micro Solutions，并在网络经济最繁荣的时候卖给了美国最大的在线服务公司。2000 年，库班买下了 NBA 球队达拉斯小牛。回顾一生，库班认为 25 岁时与两个上司的冲突是他一生的转折点。
Mark Cuban, who played football for four years in college, founded the computer information company Micro Solutions in 1983 and sold it to the largest online service company in the United States when the network economy was at its most prosperous. In 2000, Cuban bought the NBA team Dallas Mavericks. Looking back on his life, Cuban believed that the conflict with his two bosses at the age of 25 was a turning point in his life.
Reference: 马克.库班：25 岁时的经历改变一生。
Mark Cuban: The experience at the age of 25 changed his life.
BERT-seq2seqLM: 马克.库班：25 岁时与两个上司的冲突。
Mark Cuban: A conflict with two bosses at the age of 25.
BERT-seq2seqLM+SSC: 马克·库班：一生的转折点。
Mark Cuban: A turning point in his life.

As shown in Tab. 4, we listed two examples of the test dataset generated by our model. These examples include the source text, the reference summary, the summary generated by the BERT-seq2seqLM model and the generated summary by our model. The first example is about smartphones and personal computers. The generation result of the bert-seq2seqLM model takes the frequently appearing word "iPhone" as the main body of the summary, which leads to the deviation. The second example is a summary of Mark Cuban's life. From the source text, we can see that the last sentence is a summary of the whole article, but BERT-seq2seqLM chose the wrong content as the summary. BERT-seq2seqLM with semantic supervision can generate the content close to the reference summary. From the content of the generated summary, we can see that our semantic supervision method can get better results. By comparing the generated results, we can see that the semantic supervision method based on Capsule Network can reduce the semantic deviations of BERT encoding to some extent.

6 Conclusion

According to the idea of UNILM, we transformed the mask matrix of BERT-base to accomplish the abstractive summarization. At the same time, we introduced the semantic supervision method based on Capsule Network into our model and improve the performance of text summarization model on the LCSTS dataset. Experimental results showed that our model outperformed baseline systems. In this paper, Semantic Supervision method was only used in the pre-trained language model. As for other neural network models, we have not do experiments for verification yet. In this experiment, we only used the Chinese dataset and did not verify on other datasets. In the future, we will improve the semantic supervision method and experiments for its problems.

Acknowledgement: We would like to thank all the researchers of this project for their effort.

Funding Statement: This work was partially supported by the National Natural Science Foundation of China (Grant No. 61502082) and the National Key R&D Program of China (Grant No. 2018YFA0306703).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Cho, B. V. Merrienboer, C. Gulcehr, D. Bahdanau, Y. Bengio *et al.,* "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Conf. on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1724–1734, 2014.

[2] J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Conf. of the North American Chapter of the Association for Computational Linguistics*, Minneapolis, USA, vol. 1, pp. 4171–4186, 2019.

[4] L. Dong, N. Yang, W. H. Wang, F. Wei and X. D. Liu *et al.,* "Unified language model pre-training for natural language understanding and generation," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 13063–13075, 2019.

[5] Z. G. Qu, S. Y. Chen and X. J. Wang, "A secure controlled quantum image steganography algorithm," *Quantum Information Processing*, vol. 19, no. 380, pp. 1–25, 2020.

[6] D. Zheng, Z. Ran, Z. Liu, L. Li and L. Tian, "An efficient bar code image recognition algorithm for sorting system," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1885–1895, 2020.

[7] S. Sabour, N. Frosst and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, Long Beach, California, USA, pp. 3856–3866, 2017.

[8] B. T. Hu, Q. C. Chen and F. Z. Zhu, "LCSTS: A large scale Chinese short text summarization dataset," in *Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1967–1972, 2015.

[9] A. M. Rush, S. Chopra and J. Weston, "A neural attention model for abstractive sentence summarization," in *Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 379–389, 2015.

[10] D. Bahdanau, K. H. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Int. Conf. on Learning Representations*, San Diego, USA, pp. 1–15, 2015.

[11] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark *et al.,* "Deep contextualized word representations," in *Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, USA, pp. 2227–2237, 2018.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Advances in Neural Information Processing Systems*, Long Beach, California, USA, pp. 5998–6008, 2017.

[13] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing*, Hong Kong, China, pp. 3721–3731, 2019.

[14] K. T. Song, X. Tian, T. Qing, J. F. Lu and T. Y. Liu, "MASS: Masked sequence to sequence pre-training for language generation," in *Int. Conf. on Machine Learning*, Long Beach, California, USA, pp. 5926–5936, 2019.

[15] S. M. Ma, X. Sun, J. J. Xu, H. F. Wang and W. J. Li, "Improving semantic relevance for sequence-to-sequence learning of Chinese social media Text Summarization," in *Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, vol. 2, pp. 635–640, 2017.

[16] S. M. Ma, X. Sun, J. Y. Lin and H. F. Wang, "Autoencoder as assistant supervisor: Improving text representation for Chinese social media text summarization," in *Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, vol. 2, pp. 725–731, 2018.

[17] W. Zhao, J. Ye, M. Yang, Z. Lei, S. F. Zhang *et al.,* "Investigating capsule networks with dynamic routing for text classification," in *Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 3110–3119, 2018.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, Lille, France, pp. 448–456, 2015.

[19] J. L. Ba, J. R. Kiros and G. E. Hinton, "Layer Normalization," *Stat*, vol. 1050, pp. 21, 2016.

[20] K. Cho, B. V. Merrienboer, D. Bahdanau and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Conf. of the North American Chapter of the Association for Computational Linguistics*, Denver, Colorado, USA, pp. 103–112, 2015.

[21] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. for Learning Representations*, San Diego, USA, pp. 1–15, 2015.

[22] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, pp. 74–81, 2004.

[23] N. Srivastrava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov *et al.,* "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] P. Koehn, "Pharaoh: A beam search decoder for phrase-based statistical machine translation models," in *Machine Translation: from Real Users to Research*, Washington, DC, USA, pp. 115–124, 2004.

[25] J. T. Gu, Z. D. Lu, H. Li and V. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, vol. 1, pp. 1631–1640, 2016.

[26] P. J. Li, W. Lam, L. D. Bing and Z. H. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 2091–2100, 2017.

[27] S. M. Ma, X. Sun, W. Li, S. J. Li, W. J. Li *et al.,* "Query and output: Generating words by querying distributed word representations for paraphrase generation," in *Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, USA, vol. 1, pp. 196–206, 2018.