Abstractive Chinese Summarization Model Integrating Linguistic Features (Postprint)

Authors: Hu Demin, Wang Rongrong, Hu Demin

Date: 2019-03-13T00:00:00+00:00

Abstract

To address the challenge that Chinese summaries generated by traditional abstractive summarization models struggle to preserve semantic information from the source text, we propose an abstractive Chinese summarization model that integrates linguistic features. The model introduces a concatenation layer that appends features including part-of-speech, named entities, word position, and TF-IDF to word embeddings, thereby enriching the input word vectors with multi-dimensional semantic information for identifying key entities. This is combined with a pointer mechanism that selectively copies keywords from the source text into the generated summary, thus enhancing its semantic relevance. Experiments on the LCSTS news dataset demonstrate that our model achieves ROUGE scores surpassing those of baseline models. Analysis indicates that the proposed model can produce Chinese summaries with high semantic relevance.

Full Text

Abstractive Chinese Summarization Model with Linguistic Features

Hu Demin, Wang Rongrong
(School of Optical-Electrical & Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: To address the problem that Chinese summaries generated by traditional abstractive models struggle to preserve the semantic information of the original text, this paper proposes an abstractive Chinese summarization model that incorporates linguistic features. The model adds a concatenation layer that splices features such as part-of-speech, named entities, word position, and TF-IDF onto word vectors, enabling the input word vectors to contain richer semantic information for identifying key entities. Combined with a pointer

mechanism that selectively copies keywords from the source text into the summary, this approach improves the semantic relevance of generated summaries. Experiments on the LCSTS news dataset demonstrate that our model achieves higher ROUGE scores than baseline models. Analysis shows that the model can generate Chinese summaries with higher semantic relevance.

Key words: abstractive summarization model; linguistic features; key entities; word vector

0 Introduction

The ultimate goal of automatic text summarization is to generate concise, coherent summaries that preserve key information. Based on different information extraction methods, automatic text summarization techniques can be broadly divided into two categories: extractive summarization and abstractive summarization [1]. Current Chinese summarization research predominantly employs extractive methods, which calculate sentence weights based on linguistic features and copy important sentences to form summaries. However, these methods fail to consider coherence between sentences and cannot fully express the meaning of the article. Abstractive summarization methods apply neural network models trained on large amounts of data to generate new sentences that demonstrate deep understanding of the original text.

Unlike extractive methods that extract existing sentences from the source text, abstractive summarization compresses and reinterprets the main content of the document, rephrasing it using vocabulary not present in the original document. This approach generates summaries closer to human-written ones. The sequence-to-sequence model (seq2seq) proposed by Sutskever et al. [2] and the Attention mechanism introduced by Bahdanau et al. [3] have propelled the development of abstractive automatic summarization. However, abstractive methods remain in early stages with certain limitations: they rely on large-scale, high-quality training datasets; they work better for short text summarization and perform poorly on long texts; and they often generate summaries with low semantic relevance, frequently containing grammatical and semantic errors.

To improve the relevance between abstractive summaries and source texts, this paper proposes an abstractive summarization model that fuses linguistic features (referred to as LF_model). We argue that capturing key entities from the source text can make summaries more aligned with the article' s topic. Considering the impact of linguistic features of input vocabulary on summary quality, we vectorize features including part-of-speech tagging, named entity recognition, word position, and TF-IDF from the source text, then concatenate them with original word vectors. This enables the input vectors to capture key entities from the source text through multi-dimensional semantic information. Since most out-of-vocabulary (OOV) words are named entities in the source text, addressing the OOV problem helps the model output key entities from the source text. Our model combines the Pointer mechanism proposed by Gulcehre et al. [5] to

selectively copy words from the source text into the summary, thereby generating summaries with high semantic relevance to the source text. We train the model on the LCSTS news dataset and compare the evaluation scores of generated summaries with baseline models, achieving better experimental results.

1 Related Work

Current extractive summarization techniques are relatively mature, with most Chinese summarization research adopting extractive approaches. These methods calculate sentence weights based on various textual features such as sentence length, sentence position, similarity between sentences and article titles, and linguistic rules, then rank sentences according to their total weight and select high-weight sentences as summary sentences.

Rush et al. [6] first applied the Seq2Seq+Attention model to sentence summarization tasks. The Seq2Seq model, also known as the Encoder-Decoder model, uses a recurrent neural network as an encoder to read input sentences, compressing the entire sentence' s information into a continuous intermediate semantic vector. Another recurrent neural network serves as the decoder to read this intermediate semantic vector and decompress it into a target language sentence [3]. The Attention mechanism enables the model to focus on specific parts of the input at each output node, rather than treating the input as a whole and feeding it equally to every output as in previous work [3], facilitating understanding of how information in the input sequence influences the final generated sequence. The authors also proposed a method for constructing large parallel sentence pairs using Gigaword, making neural network training feasible, though the model is more suitable for generating summaries for single sentences. Lopyrev et al. [7] described an application using LSTM (Long Short-Term Memory) as the recurrent neural network computation unit, combined with attention mechanisms to generate news summaries, but did not address the OOV (Out of Vocabulary) problem. To handle OOV issues, Gu et al. [8] proposed an incorporated copying mechanism that allows parts of the summary to copy content from the source text. Nallapati et al. [9] studied the critical role of keywords in automatic summarization, using a Feature-rich Encoder to attempt to capture key concepts and entities from sentences. They also proposed a Generator-Pointer mechanism that enables the encoder to generate sentences from the source text. Paulus et al. [1] introduced an internal attention mechanism and new training methods that effectively improved text summarization quality. Hu et al. [10] constructed a large-scale Chinese corpus and provided baselines, facilitating research on Chinese summarization. Ma et al. made numerous attempts to improve the quality of abstractive summarization, introducing a similarity evaluation component in [4] to improve semantic relevance, and using an autoencoder as an auxiliary supervisor in [11] to improve Chinese news text summarization.

Drawing inspiration from extractive methods, this paper investigates the impact of linguistic features on summarization and proposes an abstractive Chinese summarization model that fuses linguistic features. Using an attention-based

encoder-decoder model as the foundational framework, we add a concatenation layer at the model' s input to fuse features including part-of-speech, named entity, word position, and TF-IDF with original word vectors to form the final input word vectors. We use Bi-LSTM (bi-directional long short-term memory) as the encoder to generate intermediate semantic vectors from both forward and backward directions, and a unidirectional LSTM as the decoder to read the intermediate semantic vector and generate target sequences. The model combines a pointer mechanism that uses a switch function at each decoding step to decide whether to predict from the vocabulary normally or copy words from the source text, ultimately generating summaries with high semantic relevance to the source text.

2.1 Linguistic Feature-Based Word Representation

Abstractive summarization methods predict and generate new summary sentences through training on large amounts of data. Summary sentences may contain sentences not present in the input document. While abstractive methods consider grammatical correctness and coherence of summary sentences, they often ignore semantic relevance between generated summaries and source documents, resulting in summaries unrelated to the source text [4]. As shown in Figure 1 [Figure 1: see original paper], RNN-generated summaries are fluent but have little association with the input source text. To address this problem, this paper extracts features including part-of-speech, named entities, word position, and TF-IDF to capture key entities from the source text. We believe that incorporating linguistic features such as part-of-speech and named entities into word vectors can help the model avoid grammatical errors and generate better summaries. The TF-IDF feature value of vocabulary can assess the word' s importance to the source text. Additionally, based on characteristics of news articles, we incorporate word position in news text as another feature into word vectors.

1) **Part-of-Speech:** Part-of-speech is a fundamental grammatical attribute of vocabulary that determines semantic orientation [12]. Extracting part-of-speech features helps explore and identify relationships between adjacent words and resolve polysemy issues in natural language, playing an important role in semantic understanding. Research shows that in summarization tasks, nouns and verbs often better reflect key information from the source text compared to other parts-of-speech. Therefore, this paper performs part-of-speech tagging on vocabulary in the training set, represents part-of-speech through embedding, and concatenates it with word vectors to enable vocabulary word vectors to contain part-of-speech features.

2) **Named Entities:** Named entities are specific entities such as person names, location names, organization names, and proper nouns that carry particular meanings [12]. In summarization tasks, named entities are primary carriers of textual information. Identifying named entities in articles

not only helps the model determine key entities representing the article' s theme but also assists in handling OOV problems. Therefore, this paper performs named entity recognition on the corpus, embeds named entities and concatenates them with word vectors to enable vocabulary word vectors to possess named entity features.

3) **Word Position:** Word position in text is another important feature of news articles. The first sentence or paragraph of news articles often covers the main idea of the entire article, and words closer to the beginning of the article are more likely to approach the article' s theme. Therefore, we calculate word position features using Equation (1) to improve summary quality.

4) **TF-IDF:** Term frequency-inverse document frequency (TF-IDF) is a statistical method used to assess the importance of a specific term for a document within a corpus. TF represents term frequency, counting how often a term appears in the text. IDF represents inverse document frequency, identifying the importance of a term across the entire corpus. TF-IDF is the product of term frequency and inverse document frequency. A larger TF-IDF indicates higher discriminative power of the term for the article. The TF-IDF calculation formula is as follows.

In Equation (3), represents the number of times term appears in document , and represents the total number of terms in document . In Equation (4), N represents the total number of documents in the corpus, and represents the number of documents in the corpus containing term . When a term does not appear in the corpus, is zero. To avoid division by zero, we use to calculate IDF.

This paper embeds words into original word vectors, then adds embedded POS, NER, Loc, and TF-IDF features after the original word vectors. Thus, vocabulary input into the encoder is represented as . At each time step, we concatenate the hidden states of forward and backward LSTMs to obtain , where the hidden state contains information from both directions of the word vector. This paper uses a unidirectional LSTM as the decoder, where represents the decoder' s hidden state at time step i, as shown in Equation (9). At each decoding step, we introduce an attention mechanism to focus attention on specific parts of the input sequence, with the content vector used to encode words the system focuses on to generate the next summary word, as shown in Equations (10)~(12).

The model combines a Pointer mechanism that uses a switch function at the decoder to decide at each decoding step whether to predict from the vocabulary normally or copy words from the source text as summary words. If , the switch is open, indicating normal prediction from the vocabulary. If , the switch is closed, pointing to a position in the source text and outputting the pointed word. The probability of the switch being open is calculated as follows:

where is the hidden state, is the word vector from the previous time step, is the context weight, and are switch parameters. We use the attention distribution

over words in the document as the distribution for sampling the pointer.

2.2 LSTM Recurrent Neural Network

The neural network-based seq2seq model consists of two components: an encoder and a decoder. LSTM (Long Short-Term Memory) networks are a special recurrent structure whose computation units add a gating mechanism to solve the gradient vanishing problem in standard RNN models. The structure of an LSTM computation unit is shown in Figure 2 [Figure 2: see original paper]. In many tasks, recurrent neural networks using LSTM structures perform better than standard RNNs. This model uses LSTM as both encoder and decoder. The hidden state calculation formula for LSTM at time t is as follows:

where represents the input gate, represents the forget gate, represents the output gate, represents the update candidate vector, represents the learned weight matrix, represents the activation function, and represents element-wise operations.

2.3 Neural Network Model with Fused Linguistic Features

The neural network model with fused linguistic features is shown in Figure 3 [Figure 3: see original paper]. This model adds a concatenation layer at the input layer to concatenate original word vectors with linguistic features including part-of-speech, named entities, word position, and TF-IDF to generate final input word vectors. Original word vectors enter the concatenation layer, which calculates word position information in the article using Equation (1) and TF-IDF feature values for each word using Equation (2). It maps part-of-speech tags and named entity tags for each vocabulary to POS embedding and NER embedding. It concatenates POS embedding, NER embedding, Loc, IF-IDF with original word vectors for each word, ultimately forming a 512-dimensional vector.

The encoder compresses the entire source text into a continuous vector, learning vector representations for each word in the source text. This paper uses Bi-LSTM as the encoder. The forward LSTM reads the input sequence from left to right, generating hidden states . The backward LSTM reads the input sequence in reverse, generating . The decoder uses a unidirectional LSTM, where represents the decoder' s hidden state at the i-th time step, as shown in Equation (9). At each decoding step, we introduce an attention mechanism to focus on specific parts of the input sequence, with the content vector used to encode words the system focuses on to generate the next summary word, as shown in Equations (10)~(12).

The model combines a Pointer mechanism. At the decoder, a switch function decides at each decoding step whether to predict summary words from the vocabulary normally or copy words from the source text. If , the switch is open, indicating normal prediction from the vocabulary. If , the switch is closed,

pointing to a position in the source text and outputting the pointed word. The probability of the switch being open is calculated as:

where is the hidden state, is the word vector from the previous time step, is the context weight, and are switch parameters. We use the attention distribution over words in the document as the distribution for sampling the pointer.

During training, when OOV words appear in the summary, explicit pointer information is provided to the model. When the word at position i in the generated summary is an OOV word or named entity, is set to 0. The optimized likelihood function is as follows:

During testing, at each time step, the model decides whether to predict from the vocabulary normally or point to a position in the source text based on the estimated switch function.

2.4 Summary Generation Process

The news summary generation process is shown in Figure 4 [Figure 4: see original paper]: a) Read news text. b) Preprocessing: Tokenize the news text to generate vocabulary Vocab, create part-of-speech tags and named entity tags for words in Vocab. c) Calculate input sequence length m=count(Vocab), create input vector array new Text_Matrix[m][], vectorize words, part-of-speech tags, and named entity tags in Vocab, obtain original word vectors , part-of-speech tag vectors , and named entity tag vectors , calculate word position feature values using Equation (1) and TF-IDF values for each word. d) Concatenate linguistic feature vectors concatenate(Text_Matrix[i][]). e) Calculate encoder layer hidden states , compute output using Equation (16), use Beam Search algorithm to select top 5 scores and iteratively predict summaries. f) Output news summary.

3.1 Dataset

The quality, content, and scale of datasets directly affect summarization performance. LCSTS is currently the largest-scale Chinese dataset, crawled and filtered from Sina Weibo, containing over 2.4 million pairs of news texts and summaries. The dataset is high-quality and covers broad domains. It originates from influential official Weibo accounts such as "People' s Daily," "Economic Observer," and "Ministry of National Defense" [10]. These news contents are well-written with fluent sentences and almost no typos, making them highly suitable for deep learning research. LCSTS contains over 2.4 million pairs in the training set, over 10,000 pairs in the validation set, and over 1,000 pairs in the test set. The validation and test sets are manually annotated with relevance scores between 1-5 for body text and titles, where higher scores indicate better quality. This paper uses the dataset provided in LCSTS to train the model and tests the model using data with scores above 3 from the test set.

3.2 Preprocessing

This paper first preprocesses texts in the corpus. Words are the smallest linguistic units that can be used independently. We conduct experiments using word vectors based on word segmentation, using the Stanford CoreNLP toolkit for tokenization, part-of-speech tagging, and named entity recognition. Table 1 shows an example of a news article after preprocessing. We count the frequency of each vocabulary term in the dataset, sort words by frequency, and select high-frequency words to build a vocabulary of size 50,000. The `<unk>` symbol in the vocabulary is used to replace words in the test dataset that do not appear in the training vocabulary. Using the Gensim toolkit, we perform embedding on 40 types of Chinese part-of-speech tags including verbs, adjectives, nouns, pronouns, etc., to generate POS embeddings. We also perform embedding on 8 types of named entity tags including person names, organization names, location names, time, date, currency, percentage, and non-named entities to generate NER embeddings. We use the CBOW model to generate original word vectors for each word, mapping each word' s original word vector to its corresponding POS embedding and NER embedding.

3.3 Evaluation Metrics

Effectively and reasonably evaluating text summarization quality is challenging. Current text summarization evaluation methods fall into two categories: intrinsic evaluation methods that compare generated summaries with reference summaries, where higher similarity indicates better quality; and extrinsic evaluation methods that apply summaries to specific tasks and evaluate effectiveness based on task improvement. This paper uses the popular intrinsic evaluation method ROUGE. ROUGE is an automatic text summarization evaluation method proposed by Lin et al. [13] that evaluates summary quality by counting overlapping basic units between automatically generated summaries and reference summaries. The reference summaries used in this paper are human-written summaries from the dataset.

Commonly used ROUGE evaluation metrics include ROUGE-N and ROUGE-L. ROUGE-N represents the n-gram recall rate of system-generated summaries, while ROUGE-L represents the longest common subsequence between system summaries and reference summaries. The ROUGE-N calculation method is shown in Equation (17):

where represents the number of overlapping n-grams between system output summaries and reference summaries, and R represents reference summaries. ROUGE-L considers word order in summaries, providing more reasonable evaluation. This paper uses the standard toolkit provided by Lin, selecting ROUGE-1, ROUGE-2, and ROUGE-L to evaluate summary quality generated by our model.

3.4 Experimental Setup

Incorporating linguistic features including part-of-speech, named entities, position features, and TF-IDF expands word vector dimensions, enabling input word vectors to contain richer semantic information. Part-of-speech information helps the model focus on nouns and verbs in the source text, recognizing named entities aids in outputting OOV words, TF-IDF helps identify important vocabulary in the corpus, and position information helps the model focus on how word position affects summarization. Ultimately, this generates summaries closer to the original theme.

This paper conducts experiments using the TensorFlow framework. Original word vectors have a dimension of 350, and after the concatenation layer, input word vectors to the encoder have a dimension of 512, with hidden layer size of 512 and batch size of 64. We use the Adam optimizer with default parameters. During testing, to obtain summaries that best fit the language model, we use the Beam Search algorithm with beam size set to 5 to generate summaries.

3.5 Experimental Results and Analysis

This paper validates the model' s generation effectiveness using the LCSTS Chinese dataset, comparing experimental results with the RNN context model proposed by Hu et al. [10], the COPYNET model proposed by Gu et al. [8], and the SRB model proposed by Ma et al. [11]. Our model achieves higher ROUGE scores than the aforementioned models, as shown in Table 2 .

The RNN context model is a context-aware summarization generation model used by Hu et al. [10]. The authors compared word-level and character-level tokenization on the LCSTS dataset, with experimental results showing that character-level tokenization outperforms word-level tokenization. This is because the dictionary generated from character tokenization is far smaller than that from word tokenization. Character dictionaries can cover more source content and effectively reduce OOV problems. Results from the RNN context model have become the baseline for subsequent Chinese summarization research using LCSTS data.

The COPYNET model, proposed by Gu et al. [8], addresses OOV problems by introducing a copying mechanism in the Seq2seq+Attention model, allowing parts of the summary to copy content from the source text. It achieves higher ROUGE scores on word-based summarization tasks.

The SRB model, proposed by Ma et al. [11], is a semantic relevance-based neural model that encourages semantic similarity between text and summary. SRB can generate summaries with high semantic relevance for character-based summarization tasks but achieves lower ROUGE scores than COPYNET as it does not consider OOV problems.

Since Chinese characters often have multiple meanings, using character-based vectors may misinterpret sentence meanings. This paper argues that word-based

representation can more accurately capture article semantics. Therefore, our model uses word-based vectors as input. Additionally, combining the Pointer mechanism to selectively output words from the source text can effectively solve OOV problems and output keywords from the source text.

This paper compares summaries generated by the LF_model fused with part-of-speech, named entities, TF-IDF values, and Loc features against summaries generated by a model lacking these features. Results are shown in Table 3 . Experimental results demonstrate that the model incorporating linguistic features achieves higher ROUGE scores, proving the impact of linguistic features on summary quality.

Figure 5 [Figure 5: see original paper] shows an example of a summary generated by our model. Through observation, we can see that traditional RNN models generate summaries with poor readability, low semantic relevance, and OOV problems. Our model captures key vocabulary from the news article such as "Premier Li Keqiang," "entrepreneurs," and "free trade zone," demonstrating stronger readability and alleviating OOV problems. Results show that the proposed model generates summaries closer to human-written ones with higher semantic relevance to source content.

4 Conclusion

Considering the impact of input vectors on summarization, this paper proposes a neural network summarization model that fuses linguistic features to address low semantic relevance problems. The model' s concatenation layer incorporates linguistic features into word vectors, enabling the model to capture key entities from source text, while the pointer mechanism solves OOV problems and outputs key entities. Experimental results on the LCATS dataset demonstrate that the proposed model not only achieves higher ROUGE scores than baseline models but also captures key entities from source text, alleviates OOV problems, and generates summaries with high semantic relevance to source text.

References

[1] Paulus R, Xiong Caiming, Socher R. A deep reinforced model for abstractive summarization. [EB/OL]. (2015). http://cn.arxiv.org/abs/

[2] Sutskever I, Vinyals O, Le Quoc V. Sequence to sequence learning with neural networks [EB/OL]. (2014-12-14). https://arxiv.org/pdf/1409.3215v3.pdf.

[3] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [EB/OL]. (2014) . http://cn.arXiv.org/abs/

[4] Ma Shuming, Sun Xu, Xu Jinging. Improving semantic relevance for sequence-to-sequence learning of Chinese social media summarization [C]//Proc of Meeting of Association for Computational Linguistics. 2017: 635-640.

[5] Gulcehre C, Ahn S, Nallapati R. Pointing the unknown words [EB/OL]. (2016) . http//cn.arxiv.org/abs/1603.08148.

[6] Rush A M, Chopra S, Weston J. A neural model for abstractive sentence summarization [C]//Empirical Methods in Natural Language Processing. 2015: 379-389.

[7] Lopyrev K. Generating new headlines with recurrent neural networks [J]. Computer Science, 2015.

[8] Gu Jiatao, Lu Zhengdong, Li Huang. Incorporating copying mechanism in sequence-to-sequence learning [C]//Proc of the 54th Annual Meeting of the Association for Computational Linguistics. ACL. 2016:

[9] Nallapati R, Zhou Bowen, dos Santos C. Abstractive summarization using sequence-to-sequence RNNs and beyond [C]//Proc of the 20th SIGNLL Conference on Computational Natural Language Learning. 2016: 280-290.

[10] Hu Baotian, Chen Qingcai, Zhu Fangze. LCSTS: a large scale Chinese short text summarization dataset [J]. Computer Science, 2015,

[11] Ma Shuming, Sun Xu, Lin Junyang. Autoencoder as assistant supervisor: improving representation for chinese social media summarization [EB/OL]. (2018) . http//cn. arxiv. org/abs/1805. 04869.

[12] 宗成庆. 统计自然语言处理 [M].2 版. 北京: 清华大学出版社, 2013: 150-175. (Zong Chengqing, Statistical natural language processing [M]. 2nd ed. Beijing: Tsinghua University, 2013: 150-175.)

[13] Lin C Y, Hovy E. Automatic evaluation of summaries using n-gram co-occurrence statistics [C]//Proc of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Association for Computational Linguistics, 2003:

[14] Tan Jiwei, Wan Xiaojun, Xiao Jianguo. Abstractive document summarization with a graph-based attentional neural model [C]//Proc of Meeting of the Association for Computational Linguistics. 2017:

[15] See A, Liu P J, Manning C D. Get to the point: summarization with pointer-generator networks [EB/OL]. (2017-04-25) . http://arxiv.org/abs/1704.04368.

[16] Li Piji, Lam Wai, Bing Lidong. Deep recurrent generative decoder for abstractive text summarization [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2017: 2091–2100.

Note: Figure translations are in progress. See original paper for figures.