Abstractive Event Summarization on Twitter

Quanzhi Li
Alibaba Group, Bellevue, WA, USA
quanzhi.li@alibaba-inc.com

Qiong Zhang
Alibaba Group, Bellevue, WA, USA
qz.zhang@alibaba-inc.com

ABSTRACT

This paper presents a new approach for automatically summarizing a social media event. It utilizes the BERT model as the encoder and a Transformer architecture as the decoder. The framework also includes an event topic prediction component, and the predicted event topic will help the decoder focus more on the specific aspects of the topic category when generating summary. To make the summary more succinct and coherent, the most important messages from an event cluster are selected by a message selection model and encoded by the BERT model. Our preliminary experiment shows that our approach outperforms the baseline methods.

CCS CONCEPTS

• **Computing methodologies** → Natural language processing; Neural networks.

KEYWORDS

Twitter, event summary, BERT, event topic, social media

ACM Reference Format:
Quanzhi Li and Qiong Zhang. 2019. Abstractive Event Summarization on Twitter. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3366424.3382678

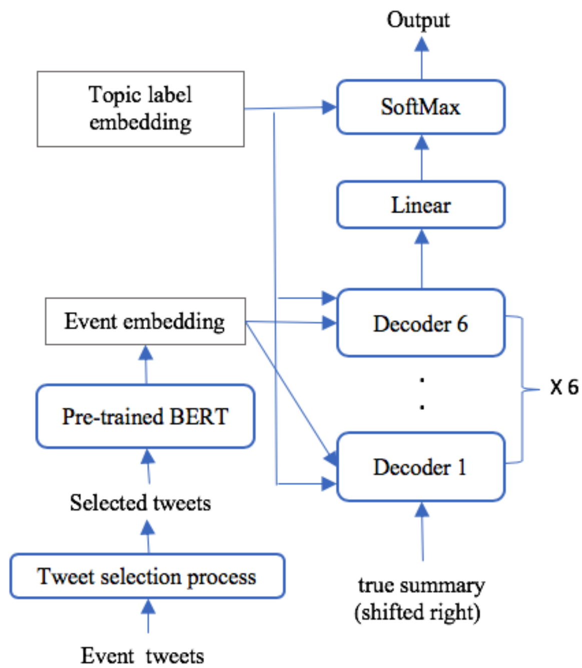

Figure 1: The high-level model structure

1 INTRODUCTION

An event discussed on Twitter usually has many tweets talking about it. The set of tweets talking about the same event form an event cluster. Then the event summarization task is, based on these tweets, to generate a summary for the event. This is a special multi-document abstractive summarization problem. This study proposed a summary generation framework that has the following features: 1. a tweet selection model selects the most important tweets in the event, by considering both how representative and informative a tweet is. Only these selected tweets will be used by the encoder. This will make the summary more precise and coherent. 2. An event topic prediction model predicts the topic category of the event, by utilizing rich topic classification datasets, which is much easier to be obtained than the event summary dataset. The topic information will be exploited by the decoding process to generate different summary styles for events in different topic areas, and this will make the model focus more on topic-specific aspects. 3. Our generation model is based on the Transformer [9] architecture, and we use the BERT model [1] as the encoder, to better exploit the grammar and semantic information of tokens learned from the pre-trained model. 4. Since the BERT encoder is pre-trained and the decoder is not, in order to smoothly integrate these two parts together, we employ two separate optimizers for these two components during the training process. Sequence to sequence (seq2seq) learning has been used in a variety of language generation applications. Our model also belongs to this widely used seq2seq paradigm [7, 8]. Pre-training has been widely used in natural language processing (NLP) tasks to learn better language representation. The pre-training on large amount of unlabeled data and fine-tuning with small scale labeled data are helpful for many tasks, and it is also used in the encoder part of our model in this work. Devlin et al. [1] proposed BERT based on masked language modeling and next sentence prediction, and achieved state-of-the-art results on multiple NLP tasks. The pre-trained BERT model [1] is used in the encoder part of our model. There are also some works on pre-training the encoder-decoder model for language generation [2, 6].

We conducted a preliminary experiment on a small dataset, and the result shows that our approach outperforms the baseline methods.

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3382678

2 METHODOLOGY

Figure 1 shows the high-level model structure. It has several components: the tweet selection process, the event topic embedding, the BERT based encoder, and the Transformer decoder. Given an event cluster, we generate its summary through the following three steps: 1. all its tweets are first processed by the tweet selection model, which selects a set of tweets to represent the event; 2. This set of selected tweets are passed to an event topic classifier, and an event topic is predicted; 3. This set of selected tweets are concatenated together according to the order given by the tweet selection model, and then they are encoded by a pre-trained BERT model. The output of the BERT model, i.e. event embedding, and the event topic embedding are fed to each of the six decoder layers in a Transformer model. The topic label embedding is also used at the final softmax layer of the Transformer decoder, in order to add more context to help the decoder choose the correct token.

Tweet selection model: This model selects a set of important tweets for an event cluster, using the following steps: 1. Order the tweets according to their timestamp in ascending order. 2. The first one is selected, and from the second one, each tweet will be compared to each of the already selected ones, based on cosine similarity measure. If none of these cosine values is greater than a threshold, this tweet will be added to the selected group, otherwise it will be discarded. Because the pre-trained BERT model takes at most 512 tokens, the selection process will stop when it reaches the length of 510 tokens. We use this selection process to ensure that the newly added tweet will not have the same content as the already selected ones, and hope it will provide different aspects of the event from the already chosen tweets. The cosine threshold will be learned from training data.

Event topic prediction: Given a set of tweets of an event, this model predicts its topic. Compared to event summarization dataset, tweet text classification datasets are much richer and easy to get. In this study, the topic of an event is predicted using TweetSift [3], an efficient and effective real time tweet topic classifier. TweetSift exploits external tweet-specific entity knowledge to provide more topical context for a tweet, and integrates them with topic enhanced word embeddings for topic classification.

Encoder-decoder model: The encoder is a pre-trained BERT model. Using training data, the BERT-base model is fine-tuned by the concatenated tweet text from the tweet selection model. The tweet sequence is determined by the selection model. The output from the last layer of BERT is fed to the decoder side. There are two multi-head attention layers in each decoder, one is a masked multi-head self-attention, and the other one is the decoder attended on the two types of contexts, i.e. event embedding and topic label embedding. After each attention layer and the feedforward layer, there is a Add&Normalize layer. In our model, the encoder is based on a pre-trained BERT-based model, and the decoder component is not pre-trained. It is obvious that there is a mismatch between these two components. One way to handle this is to use two different optimizers for these two components. In our approach, two different optimizers are used, and each has its own learning rates and warmup steps. Previous studies have used similar approaches [6].

Table 1: Algorithm comparison result

Method	ROUGE-1	ROUGE-2	ROUGE-L
Earliest	54.6	35.2	51.6
Most retweeted	57.6	37.8	54.8
Seq2seq-LSTM-attention	65.9	47.3	62.3
Our approach	69.2	51.3	66.1

Experiment: Reuters Tracer is a real-time event detection system on Twitter [4, 5]. It clusters tweets talking about the same story into the same cluster at real-time, by considering tweet link, retweet, entity name, hashtag, etc. We used the events generated by this system in our experiment. These events mainly belong to disaster related topic areas: flood, storm, fire, armed conflict, terrorism, and infrastructure breakdown. The topics of these events are predicted by TweetSift mentioned before. To get the "true" summary, each event was reviewed by an annotator and a summary is manually written. This data set was split into training, validation and test parts. For evaluation, we use the F-measure of three ROUGE scores as the performance metrics.

To evaluate our approach, we compared it to the following approaches: 1. *The earliest tweet* - the first tweet in the event. This tweet is the first one describing the event, and so it is a special one. It is like the first sentence in an article. This is the simplest approach. 2. *The most retweeted tweet.* Retweeting reinforces a message and the number of retweets shows the popularity of the tweet. 3. *Seq2seq-LSTM-attention.* This is the basic seq2seq model with attention, and also both the encoder and decoder use a bi-LSTM model. Table 1 presents the comparison result. The result shows that our proposed approach performs better than the other methods, and the differences are statistically significant at $p=0.01$ using *t-test.* This experiment is on a small data set and the event topics are mainly about disaster. One of our future research directions is to create a bigger data set with various topics, and jointly learn the event topic prediction model and the event summarization model.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL 2019
[2] Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. NAACL 2019
[3] Quanzhi Li, Sameena Shah, Xiaomo Liu, *et al.*, TweetSift: Tweet Topic Classification Based on Entity Knowledge Base and Topic Enhanced Word Embedding. CIKM 2016
[4] Quanzhi Li, Armineh Nourbakhsh, Sameena Shah, Xiaomo Liu, Real-Time Novel Event Detection from Social Media, In Proceedings of the IEEE 33rd International Conference on Data Engineering (ICDE), 2017.
[5] Xiaomo Liu, Quanzhi Li, Armineh Nourbakhsh, *et al.* 2016. Reuters Tracer: A Large Scale System of Detecting & Verifying Real-Time News Events from Twitter, CIKM 2016
[6] Yang Liu and Mirella Lapata, Text Summarization with Pretrained Encoders, EMNLP 2019.
[7] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, and *et al.* Abstractive text summarization using sequence-to-sequence RNNs and beyond. CoNLL 2016
[8] lya Sutskever, Oriol Vinyals, and Quoc V. Le., Sequence to sequence learning with neural networks. NIPS 2014
[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, and *et al.* Attention is all you need. NIPS 2017