Integrating Extractive and Abstractive Models for Long Text Summarization

Shuai Wang[§] Xiang Zhao[§†] Bo Li[§] Bin Ge[§†] Daquan Tang[§†]

[§]*National University of Defense Technology, China*
[†]*Collaborative Innovation Center of Geospatial Technology, China*
{*wangshuai11a, xiangzhao, boli, binge, dqtang*}*@nudt.edu.cn*

***Abstract*—With the explosive growth of information on the Internet, it becomes more and more important to improve the efficiency of information acquisition. Automatic text summarization provides a good means for quick acquisition of information through compression and refinement. While existing methods for automatic text summarization achieve elegant performance on short sequences, however, they are facing the challenges of low efficiency and accuracy when dealing with long text. In this paper, we present a two-phase approach towards long text summarization, namely, EA-LTS. In the extraction phase, it conceives a hybrid sentence similarity measure by combining sentence vector and Levenshtein distance, and integrates it into graph model to extract key sentences. In the abstraction phase, it constructs a recurrent neural network based encoder-decoder, and devises pointer and attention mechanisms to generate summaries. We test our model on a real-life long text corpora, collected from `sina.com`; experimental results verify the accuracy and validity of the proposed method, which is demonstrated to be superior to state-of-the-art methods.**

***Keywords*-automatic text summarization; graph model; recurrent neural network**

I. Introduction

With the rapid development of the Internet, we have witnessed massive content surging to us on the web. While we are increasingly dependent on the Internet to get information, it brings to our attention the issue of "information overload". Thus, enhancing the efficiency of accessing massive information becomes a hot research theme.

Particularly for textual information, automatic text summarization is a powerful tool, which employs computers to process and compress texts so as to produce concise and refined content. In general, automatic summarization can be categorized into two classes [8] - extractive and abstractive methods. The essence of the extractive summarization is a selection problem, which automatically choose important sentences from the original text according to various evaluation measures; while abstractive summarization requires a deep semantic and discourse understanding of the text to produce a bottom-up summary, aspects of which may not appear as part of the original. In general, abstractive methods can condense a text more strongly than extractive ones, but the programs that can do this are harder to develop as they require use of natural language generation technology, which itself is a growing field. As a result, the majority of traditional summarization systems are extractive. Recently, deep learning provides a feasible framework for abstractive summarization, and recurrent neural network (RNN) based sequence-to-sequence learning (seq2seq) has achieved remarkable success in various natural language processing tasks, including but not limited to machine translation [2], voice recognition [3] and dialogue systems [22], etc.

Existing literature [16] shows that seq2seq performs well in summarizing short text; whereas, extrapolating the method to other tasks raises a number of technical concerns.In particular, current natural language processing applications possess usually long articles, and ask for tailored techniques to improve reading efficiency; on the other hand, however, in view of model training, especially the memory capacity of GPU's and the time cost of training process, these articles are usually too long to be processed by RNN. Attributed to this contradicting fact, seq2seq is generally applicable to only short text. For long text, RNN has a limit in the length of input sequence, and long-term dependency problem arises once the length exceeds the capacity, which will result in gradient vanishing or exploding. Among others, additional length and paragraphs of text introduces higher levels of compositionality and richer discourse structure. Nevertheless, all existing models overlook and fail to capture these features. Thus, we try to address the aforementioned challenges for long text summarization in this research.

In this paper, to adapt seq2seq to long text summarization, we present a two-phase approach, namely, EA-LTS (extractive and abstractive long text summarization). The EA-LTS model comprises two major phases, i.e., sentence extraction and summary generation. In the phase of sentence extraction, paragraph-wise summarization [5] is incorporated to extract key sentences from different paragraphs hierarchically through handling multiple levels of compositionality. Specifically, the baseline extraction method is improved by devising an advanced semantic similarity measure, which is constituted of sentence vector [11] and Levenshtein distance [12]. In the phase of summary generation, we empower RNN with both attention [2] and pointer [7] mechanisms. Distinct from the canonical encoder-decoder architecture, the attention based seq2seq model revisits the input sequence in its raw form (array of word representations) and dynamically fetches the relevant piece of information based

978-1-5386-1996-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataCongress.2017.46

mostly on the feedback from the generation of the output sequence; meanwhile, the pointer mechanism serves as a good solution of out-of-vocabulary (OOV) problem, which can locate a certain segment of the input sentence and put the segment into the output sequence. To verify the effectiveness of the model, we built a large-scale real-life dataset in Chinese from the financial sector of Sina News [1] and constructed a 200-word vocabulary of finance from historical data. Comprehensive experiments demonstrate that EA-LTS significantly and consistently improves the summarization performance for long text in ROUGE evaluation metric [13].

Contribution. In short, the main contributions of this paper can be summarized by the following three ingredients:

- Aiming at the challenge of long text summarization, we propose a two-phase model EA-LTS, which combines the advantages of extractive and abstractive methods. To the best of our knowledge, it is among the first attempts to take a two-phase approach to solve summarization problem.
- In the phase of sentence extraction, a hybrid sentence similarity measure is conceived, which better captures the importance of key sentences. In summary generation, attention and pointer mechanism are united with seq2seq to improve the summarization quality.
- A large-scale long text dataset is collected from the finance as benchmark. On the dataset, experiments are carried out with a comparative analysis of 6 mainstream models. The results indicate that EA-LTS is better than other models in terms of ROUGE.

Organization. Section II briefs related work on automatic text summarization. Section III describes the proposed model in detail, and Section IV provides the experimental results. Finally, we conclude the paper in Section V.

II. Related Work

Automatic text summarization has been extensively investigated, due to its important applications in large-scale text analytics. Existing methods for automatic text summarization generally fall into two categories [8] - *extractive* and *abstractive*.

The essence of the extractive summarization is a selection problem, while abstractive summarization is a novel text generation process which requires a deep semantic and discourse understanding of the text. Most of the conventional summarization systems use extractive approaches based on human-engineered features. These include surface features such as sentence position and length [19], the words in the title, the presence of proper nouns [17]. Sentences are typically assigned a score indicating the strength of presence of these features. Then a summary is created by identifying and subsequently concatenating the most salient sentences in a document. There are also several methods used to select the summary sentences in graph-based algorithms such as LexRank [5] and TextRank [14]. In both LexRank and TextRank, a graph is constructed by creating a vertex for each sentence in the document. The edges between sentences are based on some form of semantic similarity or content overlap. In both algorithms, the sentences are ranked by applying PageRank [18] to the resulting graph. A summary is formed by combining the top ranking sentences, using a threshold or length cutoff to limit the size of the summary.

Lately, the flourish of deep learning techniques suggests a feasible framework for abstractive summarization, with RNN based model seq2seq as a representative achieving remarkable success. seq2seq is essentially constructed on the encoder-decoder framework, which combines a representation learning encoder and a language modeling decoder to perform mappings between two sequences. Such seq2seq approaches offer a fully data-driven solution to both semantic and discourse understanding. Recent work such as Rush et al. [20]; Nallapati et al. [16] have shown that seq2seq performs remarkably well in summarizing short text. In the meanwhile, such RNN method [10] proposed by Hu et al. provided a baseline for the Chinese LCSTS dataset. Copynet method [6] proposed by Gu et al. successfully applied the Copying Mechanism, which has effectively solved the OOV problem by selectively replicating the certain segments in the input sequence to the output sequence while MRT [1] method proposed by Ayana et al. has statistically led to significant improvements for headline generation through directly optimizing model parameters with respect to evaluation metrics. The MRT method is employed in minimum risk training strategy other than maximum likelihood estimation.

III. Proposed Model

In this section, we formally define the summarization task, and outline the framework of the proposed model, and the details of the model structures involved in the two phases are followed thereafter.

A. Model Overview

For automatic summarization of long text, we develop a two-phase model by combining and leveraging the advantages of extractive and abstractive methods. By doing this, the two categories of methods serve complementarily to each other, and the overall summarization performance is enhanced.

The overall framework of the proposed model is shown in Figure 1, which comprises two major phases:

- **Phase I: Sentence Extraction** - Given a document D consisting of a sequence of sentences $\{s_1, s_2, \ldots, s_L\}$ with length L, we are interested at how to hierarchically extract key sentences so as to reduce the text length. That is, how to extract K key sentences from each paragraph in the document D to get X, where $K < L$.
- **Phase II: Summary Generation** - Assume that the output X from Phase I is arranged in the original

[1] http://finance.sina.com.cn/

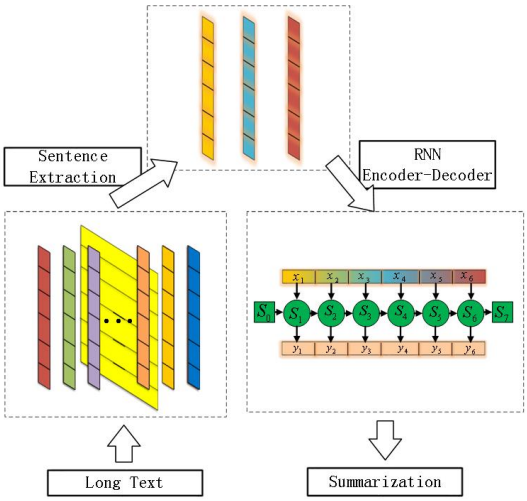

Figure 1. Overall Framework of Proposed Method

order as a sequence of words $\{x_1, x_2, \ldots, x_M\}$ with M words, in which x_i comes from a fixed word vocabulary V of size $|V|$. The summary generation phase aims to take X as input, and generates a target sequence $Y = \{y_1, y_2, \ldots, y_N\}$ of N words, where $N < M$. Typically, the conditional probability is modeled by a parametric function with parameters $\theta : P(Y|X) = P(Y|X : \theta)$. Training involves finding the which maximizes the conditional probability of "sentence-summary" pairs in the training corpus. If the model is trained to generate the next word of the summary, given the previous words, the above conditional probability can be factorized into a product of individual conditional probabilities,

$$P(Y|X;\theta) = \sum_{t=1}^{N} P(y_t|Y_{<t}, X;\theta), \qquad (1)$$

where $Y_{<t} = \{y_1, y_2, \ldots, y_{t-1}\}$; that is, the t-th word y_t in the target sequence is generated according to all $y_{<t}$ generated in past as well as the input sequence X.

B. Phase I: Sentence Extraction

In this phase, we incorporate a graph model to handle the task of key sentence extraction, where a hybrid sentence similarity measure is conceived by combining sentence vector similarity and Levenshtein distance. That is, the sentences are modeled into a complete graph with every edge weighted to the similarity of corresponding two sentences, and then ranked by PageRank [18]. An extraction of the document is formed by taking the top-ranked sentences, using a threshold or length cutoff to limit the size of the result. The model structure is shown in Figure 2.

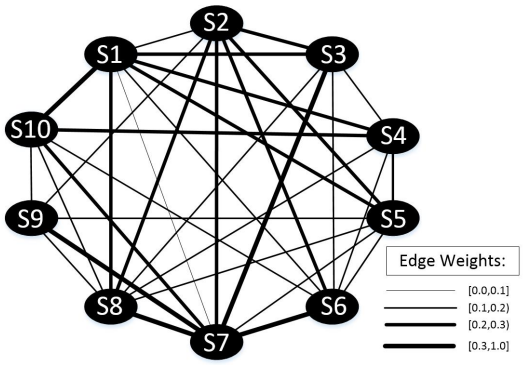

Figure 2. Structure of Graph Model for Extraction

Specifically, the text is first transformed into a topology structure using a complete graph model. Each node S_i in the graph represents a sentence from the input text, and for each pair of sentences, the weight of the connecting edge depicts the semantic similarity between the two corresponding sentences, which is expressed by the thickness of edges in the figure. Then, using PageRank algorithm, we can determine the importance of each node, or sentence. Finally, the extraction from the document is obtained by outputting the key sentences based on their importance from high to low, and arranged in the same order as the original text.

1) Similarity Calculation: The accuracy of the similarity calculation between sentences directly affects the choices of key sentences. Classic TF-IDF-based method [5], i.e., LexRank, uses TF-IDF to estimate sentence importance; however, TF-IDF is unable to capture the semantic similarity between words, but only a simple calculation of the coincidence degree. We argue that sematic similarity is more vital to determine sentence importance, rather than based on frequency information as TF-IDF. To this end, we propose a sematic similarity measure that combines *Levenshtein distance* and *sentence vector similarity*, so as to capture similarity both semantically and literally.

Levenshtein distance, also referred to as edit distance, is a string metric for measuring the difference between two sequences. Informally, Levenshtein distance between two words is the minimum number of single-character edits - insertions, deletions or substitutions - required to change one word into the other.

The incorporation of sentence vector is inspired by recent work on learning vector representations of words using neural networks [15]. In particular, every sentence is mapped to a unique vector, represented by a column in matrix, and every word is also mapped to a unique vector. The sentence vector and word vectors are averaged or concatenated to predict the next word in a context. After being trained, the sentence vectors can be used as features for the sentence, and consequently, we can obtain the semantic similarity of the

two sentences simply by calculating the Cosine function of the sentence vectors. Moreover, the larger the Cosine value, the greater the similarity.

On one hand, Levenshtein distance describes the collinearity of sentences; on the other hand, sentence vector similarity tries to capture the semantic similarity between sentences. Thus, our proposal for similarity calculation is derived by combining with the two similarity/distance. Formally, given two sentences S_i and S_j, with each sentence vector being represented by $V(S_i)$ and $V(S_j)$, the similarity $sim(S_i, S_j)$ of S_i and S_j is defined as

$$\alpha \cdot \cos(V(S_i), V(S_j)) + (1 - \alpha) \cdot Lev(S_i, S_j), \quad (2)$$

where α represents the weight in semantic similarity, and hence, the weight of literal similarity is $1 - \alpha$, $Lev(S_i, S_j)$ is the Levenshtein distance, and $\cos(V(S_i), V(S_j))$ denotes the Cosine value of $V(S_i)$ and $V(S_j)$.

2) Scoring Strategy: Denote $G = (V, E)$ as an undirected graph with the set of vertices V and set of edges E. For a given vertex V_i, $In(v_i)$ is the set of vertices that point to it (predecessors), and $Out(V_i)$ is the set of vertices that vertex V_i points to (successors). In addition, w_ij represents the weight of the edge between the V_i and V_j node, where the value of w_ij is calculated using the similarity $sim(S_i, S_j)$ between the two sentences S_i and S_j. The score of the vertex V_i is defined as follows:

$$S(V_i) = (1 - d) + d \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} S(V_j), \quad (3)$$

where d is a damping factor between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. In practice, d is usually set to 0.85.

C. Phase II: Summary Generation

In the phase of summary generation, the key sentences extracted are fed into a RNN based encoder-decoder model in order to get the ultimate summary. We sketch the model structure in Figure 3.

In the framework of RNN based encoder-decoder, the source sequence $X = [x_1, x_2, \cdots, x_M]$ is extracted in the first phase, and then converted to a fixed length vector c by the encoder RNN, i.e.,

$$h_t = f(x_t, h_{t-1}); \quad c = \phi(\{h_1, h_2, \ldots, h_M\}), \quad (4)$$

where h_t is the RNN state, c is the so-called *context vector*, f is the dynamics function, and ϕ summarizes the hidden states, e.g., choosing the last state h_M. In practice, it is found that gated RNN alternatives such as LSTM [9] or GRU [4] often performs better than basic RNN.

RNN typically deals with text sequence from start to end, and builds the hidden state of each word by only considering its preceding words. It has been shown that the hidden state

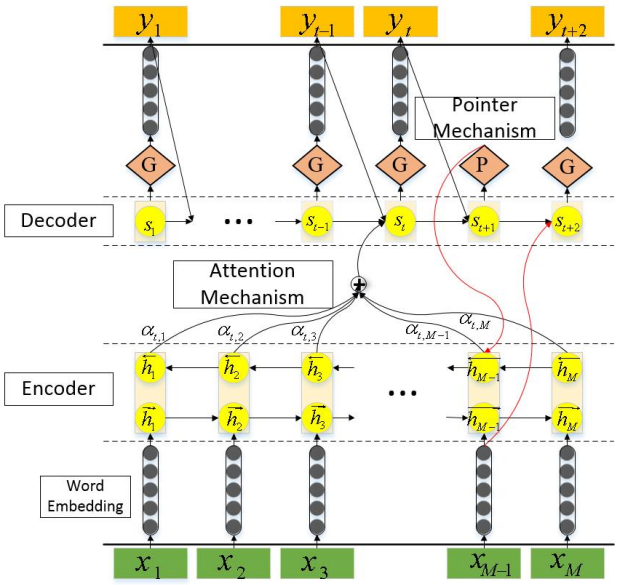

Figure 3. Structure of RNN based Encoder-Decoder

should also consider its following words. Thus, we apply bi-directional RNN (BRNN) [21] to learn hidden states using both preceding and following words. BRNN processes the input sentences in both forward direction and backward direction with two separate hidden layers calculated with RNNs. For each position i, we concatenate its forward hidden states $\overrightarrow{h_1}, \overrightarrow{h_2}, \cdots, \overrightarrow{h_i}$ and backward hidden states $\overleftarrow{h_1}, \overleftarrow{h_2}, \cdots, \overleftarrow{h_i}$ into the final hidden state,

$$h_i = \overrightarrow{h_i} \oplus \overleftarrow{h_i}, \quad (5)$$

where $\overrightarrow{h_i}$ is calculated by following Equation (4), and $\overleftarrow{h_i}$ is calculated by $\overleftarrow{h_i} = f(x_i, \overleftarrow{h_{i+1}})$, in which operator $\oplus$ indicates concatenation.

The decoder RNN is to unfold the context vector c into the target sequence, through the following dynamic model,

$$s_t = f(y_{t-1}, s_{t-1}, c) \quad p(y_t | Y_{<t}, X) = g(y_{t-1}, s_t, c), \quad (6)$$

where s_t is the RNN state at time t, y_t is the predicted target word at t (through function $g(\cdot)$) with $y_{<t}$ denoting the history $\{y_1, y_2, \cdots, y_{t-1}\}$. The prediction process is typically a classifier over the vocabulary. It is can be seen from Equation (6) that the probability of generating target word is related to the current RNN state, the history of the target sequence and the context c. The essence of the decoder is to classify the vocabularies by optimizing the loss function in order to generate the vector representing the feature of the target word y_t. After the vector passes through a softmax function, the word corresponding to the highest probability is the result to be output.

1) Attention Mechanism: On the basis of the basic RNN encoder-decoder model, *attention* mechanism is introduced

to release the burden of summarizing the entire source into a fixed-length vector as context. The attention uses a dynamically changing context c_t in the process of decoding by replacing the uniform distribution in bag-of-words with a learned soft alignment between the input and the summary.

Specifically, assuming that the context vector of the input is c_t at the time of generating the t-th target word y_t:

$$c_t = \sum_{i=1}^{M} \alpha_{ti} h_i, \tag{7}$$

where h_i is hidden state from the encoder, α_{ti} indicates how much the i_{th} word x_i from the original sequence contributes to generating the t-th word in the summary. α_{ti} is computed as follows:

$$\alpha_{ti} = softmax(Z_\alpha^T \tanh(W_\alpha s_{t-1} + U_\alpha h_i)), \tag{8}$$

where Z_α^T is the weighting vector, and W_α and U_α are weighting matrices.

2) Pointer Mechanism: It is identified that too many words may cause buffer overflow. To resolve the issue, a feasible solution is to select top-n most frequent words to form the vocabulary, and the other words instead with `UNK`. Since our RNN architecture depends on embeded representations of words, its performance may degrade on such `UNK` due to sparse training data. To address this OOV problem, *pointer* mechanism is applied in a switching generator/pointer architecture.

In this model, the decoder is equipped with a "switch" that decides between using the generator G or a pointer P at every time-step. That is, the decoder produces a word from its target vocabulary with the switch turning generator G on while the decoder instead generates a pointer P to one of the word-positions in the source and then the corresponding word is "copied back" into the summary. The switch is trained based on the hidden state of the decoder in a sigmoid activation function over a linear layer to decide whether P is enabled. Specifically,

$$P(s_i) = \sigma(Z^T(W_h h_i + W_e E[o_{i-1}] + W_c c_i + b)), \tag{9}$$

where $P(s_i)$ is the probability of the switch turning P, h_i is the hidden state at the $i{-}th$ time step of the decoder, $E[o_{i-1}]$ is the embedding vector of the emission from the previous time step, c_i is the attention-weighted context vector and W_h, W_e, W_c, b and Z are model parameters.

With this, we have described our proposed two-phase model EA-LTS for long text summarization. Next, we experimentally verify the effectiveness and efficiency of our model against other state-of-the-art competitors.

IV. Experiments

This section presents the experimental results with analyses.

A. Dataset

The experimental dataset is collected by a self designed topic crawler from real world Chinese data, and it contains 1M articles and the corresponding titles (2.5G in total). The dominant composition of the dataset includes several kinds of news of securities, stocks, funds, listed companies and other series, which is from the financial sector of `Sina News`. The length of each article is between 500 and 1,000 characters, and some typical sample articles are shown in Table I.

A size of 200 words specific vocabulary of common names, company names and the corresponding proper nouns is constructed according to historical data from 2010 to 2016, which significantly improves the accuracy of the word segmentation, and from a certain extent, solves the problem of OOV in text generation process. Then, by removing the data unstructured, the remaining 0.8M article pairs are used as experimental data. The dataset is divided into two situations based on characters and words, and the word segmentation is achieved by tool of Jieba [2]. We set the vocabulary size to 3,000 characters and 30,000 words, respectively, with the artificial specific vocabulary included. According to the method of LCSTS dataset [10], the processed data is divided into three parts, with ratio of 98%, 1.8% and 0.2%, respectively, i.e., training set, verification set and test set. Thus, the test set used for evaluation of the model gets a size of 1,600 pairs.

B. Experiment Setting

In the phase of sentence extraction, PageRank [18] algorithm is chosen as the iterative algorithm on the graph model, with the damping coefficient d is set to 0.85, and the number of key sentences extracted is set to 3. When calculating sentence similarity, the semantic level similarity weight is $\alpha = 0.75$.

In the phase of summary generation, Bi-GRU is selected and the layers of encoder and decoder are 5 and 3, respectively. Using the sentence vector model to pre-train the word vector as the input of the encoder, the dimension of the word vector is 300 degrees, and the batch size is set to 64 during training. The Beam-Search algorithm [2] is used in order to rapidly find the optimal solution through all the generating vectors with the beam-size setting to 10.

C. Evaluation Method

Currently, the most widely used evaluation metric for document summarization is ROUGE [13] which is also the most common evaluation metric in Document Understanding Conference (DUC) [3], a large scale summarization evaluation sponsored by NIST. The basic idea of ROUGE is to count the number of overlapping units between system summaries and the reference summaries, such as overlapped n-grams.

[2]https://pypi.python.org/pypi/jieba
[3]http://duc.nist.gov/

Table I
SAMPLE LONG TEXT DATA

摘要：下周沪深两市限售股解禁市值约1163亿元。
Summary: Next week, Shanghai and Shenzhen restricted shares lifted the market value of about 116.3 billion.
文章：根据沪深交易所安排，下个交易周9 月7 日至11日两市将有53家公司共计5374亿限售股解禁上市，解禁市值约1163 亿元。此次解禁后，沪市将有绿庭投资金 西藏药业京运通渤海轮渡成为新增全流通公司深市没有新增全流通公司。...统计显示，下周解禁的53家公司中，有10 家公司限售股在9 月10日解禁，解禁市值60212亿元，占到全周解禁市值的5177%。(512字)
Article: According to the Shanghai and Shenzhen stock exchange arrangements, the next trading week from September 7th to 11, the city will have a total of 53 companies a total of 537.4 billion restricted shares lifted the ban, lifting the market value of about 116.3 billion. After the lifting of the ban, the Shanghai stock market will have Green Investment, Tibetan Medicine, Beijing Express, Bohai Ferry to become the new full circulation company, while Shenzhen did not add the full circulation company. Statistics show that 10 in 53 companies that lifted next week, will lift the ban in the September 10, lifting the market value of 60212 billion, accounting for 5177% of the market through the whole week. (512 characters)

摘要：人民币中间价调降107点创逾一个月最大单日降幅。
Summary: RMB mid-price cut 107 points to make the largest single-day decline in more than one month。
文章：网易财经2月17日讯今日人民币兑美元中间价报65237，较上个发布日2 月16日65130，调降107个基点调降幅度创1月7日以来最大。...我相信，今年开局主导市场的悲观情绪，有些夸大了，另外得益于日本当局展现出的遏制日元升势的立场，我预计美元或将回升，但我们还需要看到预示美国经济复苏的其它因素。(712字)
Article: NetEase Finance February 17, today the RMB against the US dollar reported 65,237 points, compared with the previous release date on February 16 65130 points, down 107 basis points to reduce the rate, is the largest decline since January 7.I believe that this year's market-led pessimism is somewhat exaggerated, and thanks to the Japanese authorities to curb the yen's rally, I expect the dollar or will pick up, but we also need to see the other factor of US economic recovery. (712 characters)

摘要：沪指下跌失守3000点板块全线失守。
Summary: Shanghai Composite Index closed below 3,000, with all sectors seeing a decline.
文章：盘面上，两市板块全部飘绿次新股大幅领跌交运设备贸易建筑保险军工石油开采等板块跌幅居前。个股方面，中国天楹匹凸匹朗科智能等19 股涨停,新力金融世龙实业天鹅股份等15股跌停。...另外，随着国庆假日的即将来临，市场的谨慎情绪再度升温，预计节前难有大行情，仓位上仍以小仓位操作为主，弱势市场中一定要把保证资金的安全放在第一位。(632字)
Article: On the surface of the market, the two plates are floating green, the new shares has led the decreases in delivery of equipment, construction, insurance, trade industry, petroleum plate. As for the individual stocks, the Chinese TianYing Pimpang Lengke intelligence and other 19 shares daily limit, Sony Finance Dragon Industrial Swan shares and other 15 shares limit. In addition, with the National Day holiday approaching, the market is expected to rise again before the cautious mood, difficult to have a big market, small positions still in operating positions, the weak market must take to ensure the safety of funds in the first place. (632 characters)

In our evaluation, we consider two types of ROUGE's to evaluate the computer-generated summaries - ROUGE-N and ROUGE-L. ROUGE-N counts n-grams, and ROUGE-L counts longest common sub-sequences. Suppose the reference summary is R, and the ROUGE-N computation formula is as follows.

$$\frac{\sum_{R\in\text{Models}}\sum_{(n\text{-gram})\in R} Count_{match}(n\text{-gram})}{\sum_{R\in\text{Models}}\sum_{(n\text{-gram})\in R} Count(n\text{-gram})}, \tag{10}$$

where n indicates the type of n-gram, e.g., uni-gram and bi-gram, respectively, corresponding to ROUGE-1 and ROUGE-2, $Count_{match}(n\text{-gram})$ is the number of n-grams matched between system and reference summaries, and $Count(n\text{-gram})$ is the total number of n-grams in reference summary.

D. Results and Analyses

Over the real data set, following models and methods are compared:

- **RNN method** [10]: Denoted by RNN, it is directly used in the encoder-decoder to build a standard seq2seq learning framework with the results serving as a benchmark;
- **RNN context method** [10]: Denoted by RNN Context, it is based on the RNN method with attention mechanism joined in the encoder;
- **Copynet method** [6]: Denoted by Copynet, in this method, in order to solve the OOV problem in process of text generation, copying mechanism is proposed on the basis of RNN context;
- **MRT method** [1]: Denoted by MRT, using the minimum risk training rather than the traditional maximum likelihood estimation, it makes the evaluation indicator included in the optimization objectives so as to directly optimize the model;
- **TextRank method** [14]: Denoted by TextRank, it is an extractive approach based on the graph model by ranking and selecting the key sentences as summaries;
- **EA-LTS method**: Denoted by EA-LTS, it is the proposed two-phase model for long text summarization.

The specific comparison results are shown in Table II. As can be seen from the table that EA-LTS model framework exhibits superior performance in long text processing capability, compared to RNN and MRT on ROUGE-1 (word) upgrade of 20.4% and 19% and on ROUGE-L (word) upgrade of 20.6% and 1.6%. RNN, RNN context, Copynet, MRT methods all belong to the RNN based abstractive summarization model, which is poor in processing long text due to the long-term dependency issue. These factors hinder the performance on accuracy rate, if the length of the input sequence exceeds the capacity of RNN encoder. Even if we reduce the input length by cutting the beginning or the ending of the text as parts of the input sequence, the simple

Table II
COMPARISON OF EXPERIMENT RESULTS

Models	data	ROUGE Scores		
		ROUGE-1	ROUGE-2	ROUGE-L
RNN	Word	16.2	6.5	14.5
	Char	20.1	7.0	17.7
RNN context	Word	23.4	15.1	22.1
	Char	24.7	16.3	23.2
Copynet	Word	33.1	20.3	31.3
	Char	32.6	19.7	30.0
MRT	Word	34.1	23.8	33.5
TextRank	Sentence	30.5	19.4	28.4
EA-LTS	Word	**36.6**	**25.5**	**35.1**
	Char	**33.9**	**21.1**	**32.2**

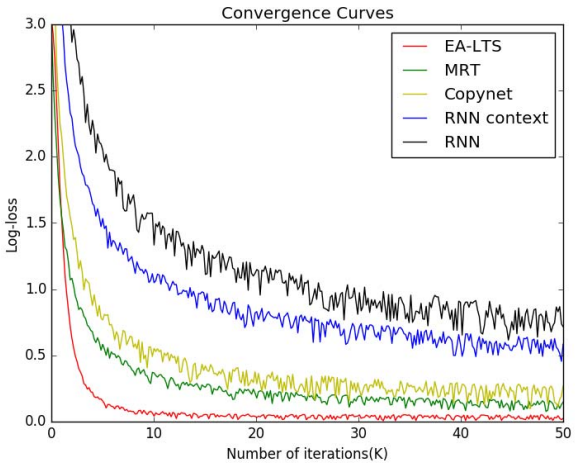

Figure 4. Log Loss Variations

interception of the beginning and ending will lose much useful information in the text. As as a result, it will also seriously affect the accuracy of the summarization system.

Figure 4 examines the time cost with increasing numbers of training times. It can be seen from the figure that EA-LTS model has almost converged at about 10,000 iterations, while the other methods have obvious log loss fluctuations, which leads to slow training speed and poor model convergence ability. As a result of the specific financial vocabulary, we achieve a great improvement in the accuracy of the word segmentation, and the use of pointer mechanism serves as a good solution of avoiding the emergence of OOV problem. With this, our model leads to a experimental fact that the quality of the summaries based on words is significantly higher than that on characters. For illustration, some sample results of our summarization system is given in Table III.

In short, according to the framework of EA-LTS, the first phase is to extract key sentences from the articles on the basis of graph model, and then in the second phase, in order to generate the final summaries, the key sentences is modeled in a RNN encoder-decoder framework. The experimental results show that not only the training speed

Table III
SAMPLE RESULTS OF SUMMARIES

句子抽取： 根据沪深交易所安排，下个交易周9月7日至11日两市将有53家公司共计5374亿限售股解禁上市，解禁市值约1163亿元。其中，沪市12家公司76858亿元，深市41家公司解禁市值53071亿元，为沪市解禁市值最大公司解禁市值排第二三名的公司是京运通和迪马股份，解禁市值分别为69亿元和4649亿元。
Sentence Extraction: According to the Shanghai and Shenzhen stock exchange arrangements, the next trading week from September 7th to 11, the city will have a total of 53 companies a total of 537.4 billion restricted shares lifted the ban, lifting the market value of about 116.3 billion. Among them, the 12 companies in Shanghai accounted for 768.88 billion, 41 companies in Shenzhen lifted the market value of 5307.1 billion yuan. The first and second company of the Shanghai stock market that lifted the market value is Beijing express and Dimma shares, with lifting the market value of 6.9*billionand*464.9 billion respectively.
标准摘要： 下周沪深两市限售股解禁市值约1163亿元。
Reference Summary: Next week, Shanghai and Shenzhen restricted shares lifted the market value of about 116.3 billion.
机器摘要： 沪深交易所限售股解禁上市共1163亿元。
System Summary: Shanghai and Shenzhen restricted shares lifted the market about 116.3 billion.

句子抽取： 网易财经2月17日讯今日人民币兑美元中间价报65237，较上个发布日2月16日65130，调降107个基点调降幅度创1月7日以来最大。中国央行周二公布数据显示，1月新增人民币贷款251万亿元，创单月记录新高。这意味着中国政府维持宽松货币政策以抵御经济放缓。
Sentence Extraction: NetEase Finance February 17, today the RMB against the US dollar reported 65,237 points, compared with the previous release date on February 16 65130 points, down 107 basis points to reduce the rate, is the largest decline since January 7.
标准摘要： 人民币中间价调降107点创逾一个月最大单日降幅。
Reference Summary: RMB mid-price cut 107 points to make the largest single-day decline in more than one month.
机器摘要： 今日人民币兑美元中间价报调降107个基点调降幅度创创单月记录新高。
System Summary: Today's RMB mid-price against the US dollar cut 107 basis points to make the highest rate among the records in a single month.

句子抽取： 盘面上，两市板块全部飘绿次新股大幅领跌交运设备贸易建筑保险军工石油开采等板块跌幅居前。今日股指跳空低开，全天一路震荡走低，个股全线尽墨，沪指再度失守3000点大关，热点纷纷回落，多头持续撤退，国庆小长假临近，操作上应保持谨慎。只是由于短期均线及上方缺口压制，导致沪指举步维艰，使得市场重心下移，这就预示着后市还将是一个缓慢的修复过程。
Sentence Extraction: On the surface of the market, the two plates are floating green, the new shares has led the decreases in delivery of equipment, construction, insurance, trade industry, petroleum plate The stock index opened lower today and closed lower, with shares seeing a decline across the board. Hot spots have dropped and the operation should be cautious with the National Day holiday approaching. Just because of the short-term moving average and the gap suppression, leading to the difficult situation of stock index with the market falling down, which indicates that the market will be in a slow recovery process.
标准摘要： 沪指下跌失守3000点板块全线失守。
Reference Summary: Shanghai Composite Index closed below 3,000, with all sectors seeing a decline.
机器摘要： 两市板块全部飘绿，沪指再度失守3000点。
System Summary: The two plates of the two city are floating green, Shanghai Composite Index closed below 3,000

is faster than the existing mainstream methods, but also the effect of EA-LTS outperforms the state-of-the-art models based on a real world dataset.

V. Conclusion

In this paper, aiming at the challenges of long text summarization, we have proposed EA-LTS through the extension and integration of the state-of-art models, which is constituted of a two-phase architecture - sentence extraction and summary generation. The method leverages the advantages of both extractive and abstractive summarization methods, and achieves significant performance improvements when dealing with long text. In addition, we also propose a new real world dataset from financial domain for long text summarization, and establish benchmark numbers on it. On the benchmark dataset, EA-LTS outperforms competitors significantly both in terms of speed and accuracy.

As future work, we plan to continue to explore from the following two aspects:

(1) In the phase of summary generation, RNN seq2seq is a fully data-driven method which requires a large-scale structured training data. However, this can not be always satisfied in practical natural language processing applications. Thus, how to build an automatic summarization system with a small amount of training data through the combination of traditional natural language processing techniques such as grammar analysis, semantic analysis, syntactic analysis, is an open problem currently.

(2) Relation extraction and event extraction can quickly get trunk information of the text, which can quickly analyze and extract key topics of multiple documents by removing noise information. Consequently, it is possible to consider the method of relational extraction and event extraction on the basis of information extraction in order to combine them into the text extraction phase of EA-LTS. Hence, the model can be extended to multi-document and multi-topic automatic summarization tasks.

Acknowledgment

This work was partially supported by NSFC under grants No. 61402494, 61402498, 71690233, and NSF Hunan under grant No. 2015JJ4009.

References

[1] S. S. Ayana, Z. Liu, and M. Sun. Neural headline generation with minimum risk training. *CoRR abs/1604.01904*, 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR abs/1409.0473*, 2014.

[3] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *NIPS 2015*, pages 577–585, 2015.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR abs/1412.3555*, 2014.

[5] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[6] J. Gu, Z. Lu, H. Li, and V. O. Li. Incorporating copying mechanism in sequence-to-sequence learning. *ACL (1) 2016*, 2016.

[7] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio. Pointing the unknown words. *CoRR abs/1603.08148*, 2016.

[8] U. Hahn and I. Mani. The challenges of automatic summarization. *Computer*, 33(11):29–36, 2000.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] B. Hu, Q. Chen, and F. Zhu. Lcsts: A large scale chinese short text summarization dataset. *EMNLP 2015: 1967-1972*, 2015.

[11] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[12] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[13] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL workshop 2004*, volume 8. Barcelona, Spain, 2004.

[14] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*, pages 3111–3119, 2013.

[16] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016: 280-290*, 2016.

[17] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *ACM 2006*, pages 573–580. ACM, 2006.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[19] D. R. Radev, T. Allison, and S. e. Blair-Goldensohn. Mead-a platform for multidocument multilingual text summarization. In *LREC*, 2004.

[20] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *EMNLP 2015: 379-389*, 2015.

[21] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[22] O. Vinyals and Q. Le. A neural conversational model. *CoRR abs/1506.05869*, 2015.