Hierarchical Bidirectional Long Short-Term Memory Networks for Chinese Messaging Spam Filtering

Wenliang Shao*, Chunhong Zhang*, Tingting Sun*, Hang Li*, Yang Ji*, Xiaofeng Qiu†
*Key Laboratory of Universal Wireless Communications, Ministry of Education
†Beijing Laboratory of Advanced Information Network
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications, Beijing, China

***Abstract*—Messaging spam filtering is an important research area in the field of natural language processing (NLP). In this paper, we propose a hierarchical bidirectional long short-term memory network based approach for Chinese messaging spam filtering. Considering that a message consists of sentences and a sentence consists of words, we design a hierarchical architecture to generate the representation of a message that aggregates the information of each word in each sentence. Besides, we notice that the errors produced by Chinese segment may affect the performance of our model. So we use the unsegmented characters as input rather than the segmented words like most of the Chinese NLP models. The experimental results demonstrate that our method outperforms most of the state-of-the-art methods on the dataset that is tagged manually by a online medical company. Meanwhile, we also show that the unsegmented character has better performance than segmented word in this task.**

I. Introduction

With the development of the Internet, we send and receive message for communication in our daily life. However, the messaging spam is becoming a severe problem in most of the communication systems. Therefore, an efficient messaging spam filtering system is essential to avoid users being beset by useless messages.

Because messaging spam filtering can be treated as text classification task, there are already many approaches proposed with respect to it. For example, many feature-based approaches have shown to perform well in this area such as Support Vector Machine (SVM), decision tree, Naive Bayesian classifier, etc [1]. However, these traditional methods need to define a large set of features manually and depend on prior linguistic knowledge heavily. It is very difficult to improve the model performance if the feature set is not very well chosen [2].

In recent years, deep learning have shown its huge potential in lots of NLP areas. As the text classification task, messaging spam filtering also benefit a lot from the development of deep learning which can learn hidden feature representations from the large scale data. What is more, there are many approaches for text classification task that can be applied to messaging spam filtering. For example, a Convolutional Neural Network (CNN) model is suitable to achieves well results on multiple benchmarks for text classification task [3]. On the other hand, [4] show that Recurrent Neural Network (RNN) based generative classification models is more robust to shifts in the data distribution. Moreover, another remarkable work is that [5] treats text as a kind of raw signal at character level, and applies temporal (one-dimensional) convolutional networks to it.

However, The most of previous models for classification simply use the whole text as input. They neglect the hierarchical structure of texts. Considering that a message consists of sentences and a sentence consists of words, we design the hierarchical model to map the hierarchical structure of messages. We find that we can obtain better representation of a message by aggregating the representations of words and sentences level by level. The contributions of this paper are as follows:

- Propose a novel hierarchical neural network for messaging spam filtering. We firstly divide the message into sentences and process each sentence respectively to generate the representations of sentences. Then we aggregate them into the final representation of message for classification. Because the structure of words in a sentence is similar to sentences in message, we apply the same architecture for both sentence and message level. In this way, the final representation is able to fully capture the context information of the whole message.
- Use unsegmented Chinese characters instead of words as the input of our model. As we know, word segment is an essential step in most of Chinese NLP tasks. Because most of single characters are meaningless in the view of linguistic, only if they form a word we can understand what they mean. However, we find that the errors caused by the word segment tools may impact the performance of the model. Fortunately, deep learning is known to capture high level hidden features that human can not understand directly in recent years. By training with a large scale corpus, the neural model have the ability to learn the meaning of character in context automatically. Therefore, we use unsegmented character as the input of embedding

978-1-5386-3349-6/17 $31.00 © 2017 IEEE
DOI 10.1109/BIGCOM.2017.25

layer instead of words directly.
- Experimentally show that our model outperforms the most previous approaches. To evaluate our model, we apply it to a dataset which is provided by a online medical company. The dataset consists of 25,000 messages that have been tagged manually.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes our model in detail. Section 4 shows our experimental results. Section 5 gives a conclusion of the whole paper.

II. Related Work

The messaging spams have puzzled people for a long time, so lots of algorithms have been proposed to solve this problem. In feature-based approaches, different sets of features are extracted and fed to a chosen classifier. [6] tried 14 methods to classify text. He found that the Widrow-Hoir, k-nearest neighbor classifier, neural network and linear least squares fit are the top performers among the learning methods whose results were empirically validated in his study. [7] compared some proposed content based filtering algorithms that rely on text classification to decide whether an email is spam or not. In this work the features are fed to some popular algorithms, including Artificial Neural Networks, SVMs, Local Mixture Support Vector Machines and Decision Trees. To go a step further for feature selection, [8] found that bringing some spam-specific features to their work could improve the performance of the model.

However, these feature-based methods need to define a large set of features manually and depend on prior linguistic knowledge heavily. The downside of them is that designing features manually is time-consuming, and performing poor on generalization due to the low coverage of different training datasets [9]. Recently, Deep learning has shown its great power in the most NLP areas. With deep learning, we can learn the context features of a text automatically, which is different from traditional classifiers and reduce the number of handcrafted features [10]. So it become a popular choice in text classification task. For example, [3] find that a simple CNN performs well to classify text. [11] use recursive neural network for text classification and show it performs well. Meanwhile, RNN is suitable for most NLP tasks, especially it performs well in text classification. [12] introduce the Tree-LSTM, a generalization of LSTMs to tree-structured network topologies. They utilized the structure of a text and applied a tree-structured LSTMs to classification and got good results. However, RNN is a biased model, where later words are more dominant than earlier words in a sentence. As CNNs need a fixed window to capture the semantic of a text [13], it is difficult to determine the window size. To address the limitation of these models, [14] propose a Recurrent Convolutional Neural Network (RCNN) and apply it to the task of text classification. To utilize the hierarchical structure of a document, [15] proposed a model that construct a document representation by learning the representations of sentences and then aggregating them into a document representation.

These methods of text classification can be applied to Chinese corpus directly. Typical approaches for Chinese text categorization, such as Naive Bayes (NB) [16], Vector Space Model (VSM) [17, 18] and Linear List Square Fit (LLSF) [19, 20], have been well studied theoretical basis derived from the information retrieval research [6, 21, 22]. But a key difference between Chinese and English is that the words in Chinese are not split by space. In many cases, NLP researchers working with Chinese use an initial segmentation module that is intended to break a text into words [23]. However, due to the absence of a set of standard segmentation performance metrics, character-based approaches have been reported to outperform the word-based approaches by some researchers [24]. In this work, we try to explore neural network based messaging spam filtering model with and without Chinese segment.

III. Model

In this section, we describe our neural network model in detail. The whole architecture of the Hierarchical Bidirectional Long Short-Term Memory Networks(HBLSTM) is shown in Figure 1. More specifically, the words in the message are firstly mapped into low-dimensional word embeddings in embedding layer. Then we utilize BLSTM to get high level features from the word embeddings in each sentence of the message. After that we use a max-pooling layer to merge the features of words and get the sentence-level representation. After obtaining the series of sentence-level representations, we utilize the BLSTM and max-pooling operations again to get the final message-level representation. Finally, we use a Logistic Regression to classify the message. The details of the model will be described in the following sections.

A. Word Embedding

The message is firstly divided into sentences. For the i^{th} sentence with n words $S_i = (w_{i1}, w_{i2}, \ldots, w_{in})$, we transfer each word w_{ij} in the sentence into a low-dimensional vector by looking up the embedding matrix $W_e \in R^{|d_w| \times |V|}$, where d_w is a hyper-parameter that represent the dimensionality of the word embedding and $|V|$ is the fixed size of input vocabulary. So the representation of the j^{th} word in the i^{th} sentence is computed by

$$x_{ij} = W_e w_{ij} \tag{1}$$

We use the word embeddings that are trained by word2vec [25] with a large-scale corpus in general domains to initialize the embedding matrix W_e. In this way, we get a sequence of real-valued vectors as the input to next layer.

B. Sentence-level Representation

LSTM units In recent years, RNNs are widely used in many NLP tasks. However, one problem of RNN is known as gradient vanishing. If the sequence is too long, the gradient may decay exponentially and make training difficult. Besides, RNN may become unable to learn to connect the information as the distance between two words grows, which is called long-term dependencies problem. As a special kind of RNN,

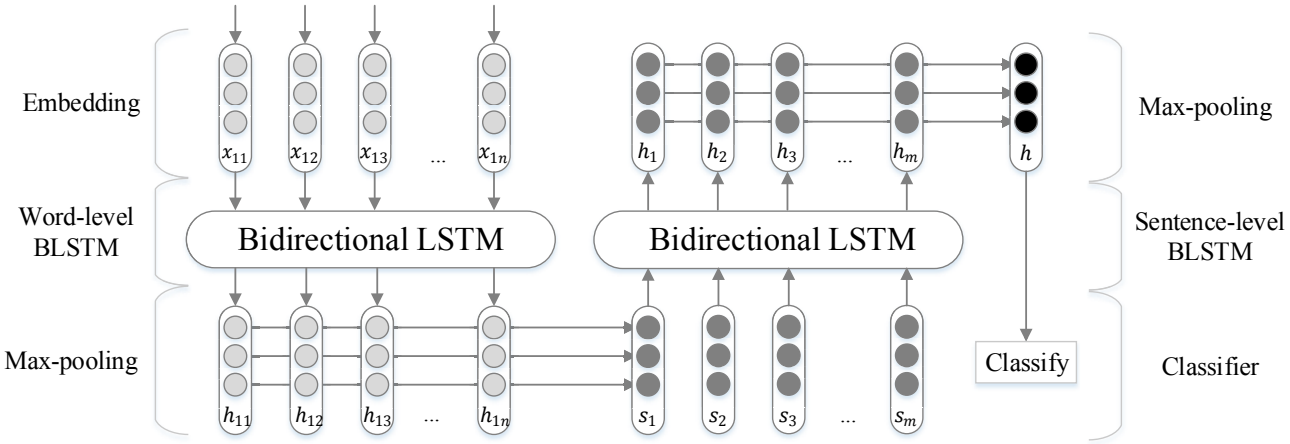


Fig. 1. Hierarchical Bidirectional LSTM

LSTMs are proposed by [26] to solve these problems that RNNs could not handle.

In order to learn long-term dependencies, [26] introduce an adaptive gating mechanism. The LSTM units use three designed gates to control cell state: an input gate i_t to control what information we are going to store in the cell state, a forget gate f_t to control what information we are going to drop from the previous cell and an output gate to control what information we are going to output. These gates use a sigmoid function to control how much information is allowed to go through. The network consists of a series of LSTM units. We define the state and the output of the t^{th} cell as c_t and h_t. The equations are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

$$\tilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \tag{6}$$

$$h_t = o_t \odot tanh(c_t) \tag{7}$$

where $\odot$ means element-wise product. W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_c are the parameters that need to be trained. The current cell accept two inputs: the output h_{t-1} of previous cell and the input x_t that is fed into this cell. Then it generate the current cell state c_t and output h_t.

Bidirectional Network Given a sequence $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ that represent the i^{th} sentence in message, we feed it into Long Short-Term Memory networks to get high level features. In our model, a drawback of one-directional LSTM is that the current cell can not get the information from the future cell. And it will hinder the semantic understanding of the whole sentence. Hence we utilize a bidirectional architecture to capture information from both past and future words (Schuster and Paliwal, 1997). With the bidirectional network, we can get high level features by:

$$\overleftarrow{h_{it}} = \overleftarrow{LSTM}(x_{it}), t \in [n, 1] \tag{8}$$

$$\overrightarrow{h_{it}} = \overrightarrow{LSTM}(x_{it}), t \in [1, n] \tag{9}$$

Now that we obtain the outputs from the forward and backward LSTM procedures. Then we combine the information of two networks by:

$$h_{it} = \overleftarrow{h_{it}} \oplus \overrightarrow{h_{it}} \tag{10}$$

where $\oplus$ refers to element-wise sum. It is optional to concatenate $\overleftarrow{h_{it}}$ and $\overrightarrow{h_{it}}$ in this step. But note that the forward and backward RNNs are trained simultaneously, and so the addition is possible even without any parameter sharing between the two RNN structures [27]. The advantage of addition is that it has lower dimensionality than vectors concatenation, which will benefits next process.

Max-pooling Our target is to generate a vector to represent the entire sentence. Now we have already get a sequence of vectors that represent all words in the sentence. A simple solution is to use the output of the last cell as the representation of the sentence. Because it has aggregated information of all words. But in practice we find that it will lose much long-term information, and the supervision at the end of the sentence is hard to be propagated to early steps in model training [2]. Therefore, we use a max-pooling layer to aggregate the information from each cell. The max-pooling operation is widely used in most of CNN models. With the development of neural networks, it is also applied to some models such as RNN and LSTM. In this work, we find that a max-pooling layer is capable of capturing the semantic information of the sentence. So we process the sequence by:

$$(s_i)_k = \max_t\{(h_{it})_k\}, k \in [1, M] \tag{11}$$

where M is the size of the hidden layer in LSTM. We extract the high level features of the i^{th} sentence as the representation of this sentence s_i to feed into the next layer.

Another alternative is using the attention mechanism. Since proposed by [28], attention mechanisms become popular and improve the performance of models in lots of NLP tasks [29, 30]. We also try the attention mechanism instead of max-pooling. But the results show that it could not get better results than max-pooling operation in this task.

C. Message-level Representation

Now that we have obtained the representation of each sentence in a message, we will further get the representation of the entire message. Considering the sequence of sentence representations have the same shape as word embeddings in a sentence, we use the similar bidirectional LSTM architecture again:

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(s_i), i \in [m, 1], \tag{12}$$

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(s_i), i \in [1, m], \tag{13}$$

$$(h)_i = \overleftarrow{h_i} \oplus \overrightarrow{h_i}. \tag{14}$$

Where h_i is the high level feature of the i^{th} sentence and m is the number of sentences.

Afterwards we use a max-pooling layer to merge all the features in sentences likewise. Each feature of message is computed by:

$$h_k = \max_t \{(h_t)_k\}, k \in [1, M] \tag{15}$$

where M is the size of the hidden layer in LSTM and h is the message feature vector and k indexes feature dimensions.

The vector h gathers the information of representation s_i of each sentence. Meanwhile, s_i gathers the information of each word in this sentence. In this way we obtain the final representation that extract the information from entire message for classification.

D. Objective Function

Messaging spam filtering is defined as a binary classification. So we use a logistic regression to classify the message. Given a message M, we compute the label of m from:

$$p(y|M) = \sigma(W_L h + b_L). \tag{16}$$

Where $W_L \in R^{1\times d}$, $b_L \in R^1$ are trained parameters. σ is a sigmoid function. We are going to minimize the objective function:

$$J(\theta) = -\frac{1}{N}[\sum_{i=1}^{N} y_i log p_i + (1-y_i) log(1-p_i)] + \lambda \|\theta\|_F^2. \tag{17}$$

Where N is the batch size of the data. y_i is the real label of the i^{th} message and p_i is the output of equation (17). In this work, we use L2 regularization to alleviate over-fitting and λ is the hyper-parameter of L2 regularization. θ represents the parameters to be trained in the network.

IV. Experiments

A. Dataset

In this work, we use a dataset provided by a online medical company. They create a communication system for doctors to communicate with each other for convenience. However, they need to face the problems that too many useless messages impact the experience of users. The definition of messaging spam in this task is various. Not only traditional harassing messages, but also some kinds of messages may be useless to receivers. For example, many users send messages just to express their gratitude to those who helped them before. But if a specialist receives too many messages like this, the real important information may be neglected. So they define pure thank-you notes as one kind of the messaging spam.

In order to distinguish the messaging spam, the company provides a dataset that have been tagged manually. The dataset consists of 25,000 messages. The statistics of this dataset are shown in Table 1. As shown in Table 1, the dataset consists of 20000 examples for training and 5000 examples for testing. More specifically, the lengths of messages range between 1 and 150 and the average length of messages is 32. As we treat the messaging spam filtering as a binary classification problem, the positive examples which mean the useful message occupy about two thirds of the whole data.

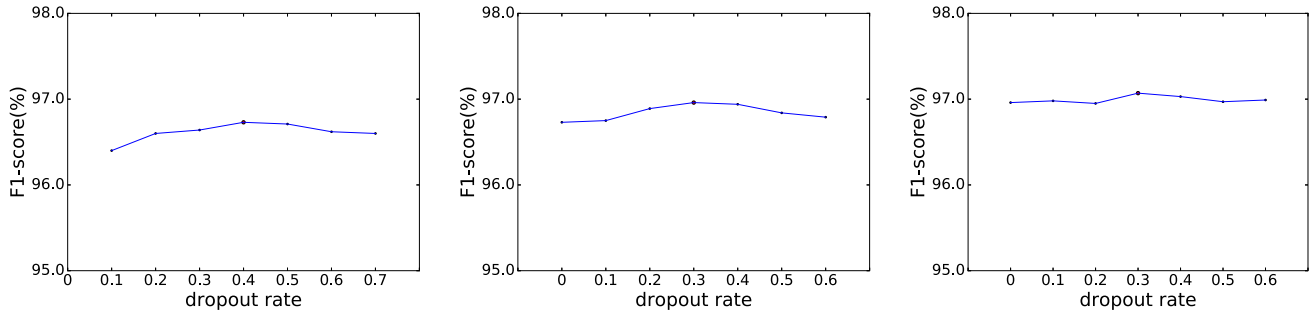
Name	Value
Number of training examples	20000
Number of test examples	5000
Number of classes	2
Percentage of positive examples	67%
Average length of messages	32

TABLE I
STATISTICS OF THE DATASET.

It is worthwhile to note that there are many medical nomenclatures in messages of the dataset, which makes it more difficult to capture the meaning of messages and impact the performance of most of the typical classifiers.

B. Preprocess

In most of the Chinese natural language processing works, a essential step of preprocessing is word segment. In this work, we compare the performance on segmented data with those without being segmented. Finally we find that using the data without being segmented achieve better performance. By checking up the data, we find that the errors may be generated by Chinese segment. The dataset is based on a special area and contains many medical nomenclature in messages. We have tried several segment toolkits, but they could hardly segment the medical nomenclature correctly. Once we segment the message incorrectly, the results may have different meanings or even become meaningless. So we use characters as input instead of words, which avoid the propagation of errors in segment.



(a) Dropout embedding layer (b) Dropout output of word-level LSTM (c) Dropout output of sentence-level LSTM

Fig. 2. F1-scores versus dropout rates.

C. Training Details

We pretrain the word vectors by word2vec tool with a large-scale corpus in general domains to initialize W_e. The dimension of word embedding is 200. The rest of matrices are initialized with random values. Because there is no official development dataset, we use 8-fold cross validation to tune the hyper-parameters. Then we use the best hyper-parameters we get to train the model on all the training data. For updating parameters, we use Adam Optimizer with a learning rate of 0.001 to minimize the object function. Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, which is based on the adaptive estimates of lower-order moments [31]. The dimensions of the hidden layer of two bidirectional LSTM are both 300. And the mini-batch size of training data is set to 10.

Moreover, we use dropout and L2 regularization to alleviate the over-fitting. To be specific, we set the hyper-parameter λ of L2 regularization to 10^{-5}. To evaluate the effect of dropout, we tune the hyper-parameters of dropout layer by layer. We firstly tune the dropout rate of word embedding layer and show that the model achieves top F1-score when dropout rate is set to 0.4. Then we tune the dropout rate of the output of word-level LSTM with word embedding layer being dropped out by 0.4. And the best rate we get is 0.3. Similarly, we evaluate the dropout rate of output of sentence-level LSTM with dropout rate of embedding layer and word-level LSTM fixed. Then we get the best F1-score when we dropout output of sentence-level LSTM by 0.3.

D. Results

As shown in Table 2, in order to compare the performance of our model with other methods, we choose some algorithms and apply them to the dataset. By comparing the F1-score of these methods, we show that our model is more efficient than most of other classifiers. We also present the F1-score of different setting in our model.

We firstly tried several feature-based classifiers [32]. We adopt CHI statistics to select the keyword features for classification. The scores of these methods are as following:

SVM: SVM offers a principled approach for machine learning (ML) problems because of their mathematical foundation in statistical learning theory [7]. It has been used for email filtering, which is similar to our task. So we apply it to the dataset and it achieves a F1-score of 94.26%.

Model	F1-score
SVM	94.62
Naive Bayesian	95.12
ANN	94.24
CNN	95.46
att-BLSTM	95.84
HBLSTM	**97.07**
+segment	96.12
+attention	96.56
+one-directional	95.82

TABLE II
COMPARING F1 WITH DIFFERENT MODELS.

Naive Bayesian: Naïve Bayesian classifier is based on Bayes' theorem and the theorem of total probability [32]. It achieves a F1-score of 95.12%.

ANN: the Artificial Neural Networks applied consists of an input, output and hidden layers for our task. We use the features of text as the input of input layer and a logistic regression for classifying. This architecture achieves a F1-score of 94.24%.

While the feature-based methods have performed not bad, they neglect the context of a message. Besides, we have to spend much time on feature selecting. To solve these problems, we try some neural network based methods to get better performance.

[3] shows that a simple CNN with little hyper-parameters tuning and static vectors achieves good results on multiple benchmarks. We try it for messaging spam filtering and achieve a F1-score of 95.46%. The attention based BLSTM model is proposed to solve the problem of relation classification [9]. Its architecture is suitable for most text classification task. So we use it as a contrast experiment of our model. In this work, it achieves an F1-score of 95.84%.

Finally we compare the performance of our model on different settings. Firstly we segment the sentence and feed the segmented words into model. But we can't get a apparent promotion. By checking the results of segment we find that there are many errors produced in the step of segment. So

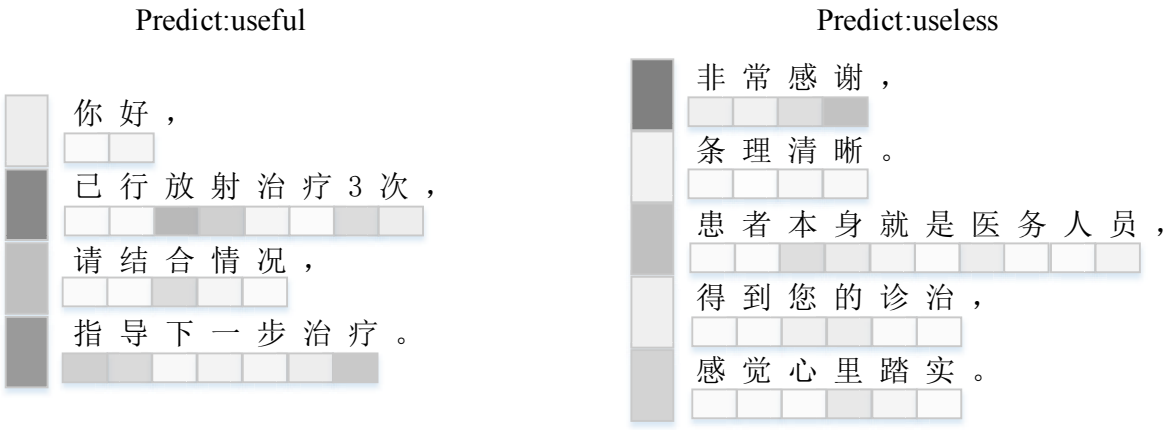


Fig. 3. Two visualization examples of contribution to max-pooling. The example on the left is positive while the one on the right is a spam. In each example, the vertical colors on the left represent the importance of different sentences, while the horizontal colors below each sentence represent the importance of different words. The brightness of color represents the contribution of word or sentence to the whole message. The darker color means the corresponding sentence or word contribute more to the message, while the brighter color means less.

we have a try with unsegmented word and achieve a F1-score of 97.07%. We also test the model that using one-directional LSTM rather than bi-directional LSTM. The F1-score of it is 95.82%, which is much lower than the model using bidirectional LSTM. Besides, in order to compare the effect of attention layer with max-pooling, we use a attention layer as a comparison. And the F1-score with attention is 96.56%, which proves that the max-pooling is more efficient than attention mechanism in this work.

E. Visualization of Max-pooling

In order to validate that our model is able to capture the informative sentences and words, we visualize sentence-level max-pooling and word-level max-pooling. To measure the contribution of each sentence to the semantic meaning of a message, for each message, we count the number of dimensions that the local feature at each sentence step contributes to the output of the max-pooling [27]. Then we compute the proportion p_s of each sentence by dividing this number by the number of total dimensions of the feature vector. After that we compute the proportion p_w of each word in a sentence similarly. The contribution of a word highly depends on the corresponding sentence, which means a word will not contribute much to the whole message if the corresponding sentence is not important, even though the word contributes a lot to this sentence. So we use $p_s p_w$ to represent the contribution of a word to the whole message.

As shown in Figure 3, we present two examples to show the effect of our model. And we use the shade of color to represent the proportion of a sentence or a word. The brightness of color represents the contribution of word or sentence to the whole message. The darker color means the corresponding sentence or word contribute more to the message, while the brighter color means less.

The first example is predicted to be a useful message. It means “Hello. We have tried radiation 3 times. Based on the real condition, please guide for the next cure.” In this message, the model focuses on the second and forth sentences. As we can see, the second sentence states what they have done and the forth sentence asks for help. They both contain the main information of the message. On the level of word, ‘放射’(radiation), ‘指导’(guide) and ‘疗’(cure) are more informative. We find that they are just the key words of the sentences that we focus on. Based on the selected sentences and words, we predict that it is a useful message.

The second example is predicted to be a spam. It is “Thank you very much. (What you said) is clear. The patient himself is a medical staff. With your diagnosis, he feel peace.” In Section 4.1 we have mentioned that we define the thank-you note as a kind of massaging spam in this dataset. The substance of this message is a thank-you note. So the network focuses on the first sentence and two characters ‘感谢’(thank you). Then the model predicts it to be a spam.

By checking up the contribution of each word and sentence to max-pooling, we confirm that our model is able to extract the informative words and sentences in a message.

V. CONCLUSION

In this paper, we propose a neural network model named Hierarchical Bidirectional Long Short-Term Memory Networks for Chinese messaging spam filtering. We evaluate our model on a dataset provided by a online medical company. And we use the unsegmented character as input. Experimental results demonstrate that our model outperform most of the other classifiers.

ACKNOWLEDGMENTS

This work was supported by NSF Projects 61602048 ,61302077, Wikipedia Content Generation based Collaborative Recognition.

REFERENCES

[1] S. Youn and D. McLeod, "A comparative study for email classification," in *Advances and innovations in systems, computing sciences and software engineering*. Springer, 2007, pp. 387–391.

[2] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths." in *EMNLP*, 2015, pp. 1785–1794.

[3] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[4] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and discriminative text classification with recurrent neural networks," *arXiv preprint arXiv:1703.01898*, 2017.

[5] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[6] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.

[7] S. A. Saab, N. Mitri, and M. Awad, "Ham or spam? a comparative study for some content-based classification algorithms for email filtering," in *Mediterranean Electrotechnical Conference (MELECON), 2014 17th IEEE*. IEEE, 2014, pp. 339–343.

[8] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 workshop*, vol. 62, 1998, pp. 98–105.

[9] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *The 54th Annual Meeting of the Association for Computational Linguistics*, 2016, p. 207.

[10] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao *et al.*, "Relation classification via convolutional deep neural network." in *COLING*, 2014, pp. 2335–2344.

[11] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, p. 1642.

[12] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[14] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *AAAI*, vol. 333, 2015, pp. 2267–2273.

[15] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of NAACL-HLT*, 2016, pp. 1480–1489.

[16] Z. Lanjuan, "The theory and experiments on automatic chinese documents classification," *Journal of the China Society for Scientific and Technical Information*, vol. 6, no. 6, pp. 433–437, 1987.

[17] Z. Tao, W. J. Cheng, Z. H. Yu, J. X. Yu, and Z. F. Yan, "The technology implementation of information mining on www [j]," *Journal of Computer Research and Development*, vol. 8, 1999.

[18] T. Zou, J.-C. Wang, Y. Huang, and F.-Y. Zhang, "The design and implementation of an automatic chinese documents classification system," *Journal for Chinese Information [in Chinese]*, 1998.

[19] C. Suqing, Z. Fuhu, and C. Huanguang, "A mathematical model for automatic chinese text categorization," *Journal of the China Society for Scientific and Technical Information*, vol. 1, pp. 27–32, 1999.

[20] Y. Yang, "Expert network: Effective and efficient learning from human decisions in text categorization and retrieval," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 13–22.

[21] J. He, A.-H. Tan, and C. L. Tan, "A comparative study on chinese text categorization methods." in *PRICAI workshop on text and web mining*, vol. 35, 2000.

[22] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 42–49.

[23] T. Zagibalov and J. Carroll, "Automatic seed word selection for unsupervised sentiment classification of chinese text," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 1073–1080.

[24] S. Foo and H. Li, "Chinese word segmentation and its effect on information retrieval," *Information processing & management*, vol. 40, no. 1, pp. 161–190, 2004.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] D. Zhang and D. Wang, "Relation classification via recurrent neural network," *arXiv preprint arXiv:1508.01006*, 2015.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[29] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention cnns," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*, 2016.

[30] Q. Li, T. Li, and B. Chang, "Discourse parsing with attention-based hierarchical neural networks," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 362–371. [Online]. Available: https://aclweb.org/anthology/D16-1035

[31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2015.

[32] S. K. Tuteja, "A survey on classification algorithms for email spam filtering," *International Journal of Engineering Science*, vol. 5937, 2016.