A Question Answering Approach to Emotion Cause Extraction

Lin Gui[a,b], **Jiannan Hu**[a], **Yulan He**[c], **Ruifeng Xu**[a,d†], **Qin Lu**[e], **Jiachen Du**[a]
[a]Shenzhen Graduate School, Harbin Institute of Technology, China
[b]College of Mathematics and Computer Science, Fuzhou University, China
[c]School of Engineering and Applied Science, Aston University, United Kingdom
[d]Guangdong Provincial Engineering Technology Research Center for Data Science, China
[e]Department of Computing, the Hong Kong Polytechnic University, Hong Kong
guilin.nlp@gmail.com, hujiannan0526@gmail.com,
y.he9@aston.ac.uk, xuruifeng@hit.edu.cn,
csluqin@comp.polyu.edu.hk, dujiachen@stmail.hitsz.edu.cn

Abstract

Emotion cause extraction aims to identify the reasons behind a certain emotion expressed in text. It is a much more difficult task compared to emotion classification. Inspired by recent advances in using deep memory networks for question answering (QA), we propose a new approach which considers emotion cause identification as a reading comprehension task in QA. Inspired by convolutional neural networks, we propose a new mechanism to store relevant context in different memory slots to model context information. Our proposed approach can extract both word level sequence features and lexical features. Performance evaluation shows that our method achieves the state-of-the-art performance on a recently released emotion cause dataset, outperforming a number of competitive baselines by at least 3.01% in F-measure.

1 Introduction

With the rapid growth of social network platforms, more and more people tend to share their experiences and emotions online. Emotion analysis of online text becomes a new challenge in Natural Language Processing (NLP). In recent years, studies in emotion analysis largely focus on emotion classification including detection of writers' emotions (Gao et al., 2013) as well as readers' emotions (Chang et al., 2015). There are also some information extraction tasks defined in emotion analysis (Chen et al., 2016; Balahur et al., 2011), such as extracting the feeler of an emotion (Das and Bandyopadhyay, 2010). These methods assume that emotion expressions are already observed. Sometimes, however, we care more about the stimuli, or the cause of an emotion. For instance, Samsung wants to know why people love or hate Note 7 rather than the distribution of different emotions.

Ex.1 我的手机昨天丢了，我现在很难过。

Ex.1 *Because I lost my phone yesterday, I feel sad now.*

In an example shown above, "sad" is an emotion word, and the cause of "sad" is "I lost my phone". The emotion cause extraction task aims to identify the reason behind an emotion expression. It is a more difficult task compared to emotion classification since it requires a deep understanding of the text that conveys an emotions.

Existing approaches to emotion cause extraction mostly rely on methods typically used in information extraction, such as rule based template matching, sequence labeling and classification based methods. Most of them use linguistic rules or lexicon features, but do not consider the semantic information and ignore the relation between the emotion word and emotion cause.

In this paper, we present a new method for emotion cause extraction. We consider emotion cause extraction as a question answering (QA) task. Given a text containing the description of an event which may or may not cause a certain emotion, we take an emotion word in context, such as "sad", as a query. The question to the QA system is: "Does the described event cause the emotion of sadness?". The expected answer is either "yes" or "no". (see Figure 1). We build our QA system based on a deep memory network. The memory network has two inputs: a piece of text, referred to as a story in QA systems, and a query. The story is represented using a sequence of word embeddings.

A recurrent structure is implemented to mine the deep relation between a query and a text. It

†Corresponding Author: xuruifeng@hit.edu.cn

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1593–1602
Copenhagen, Denmark, September 7–11, 2017.

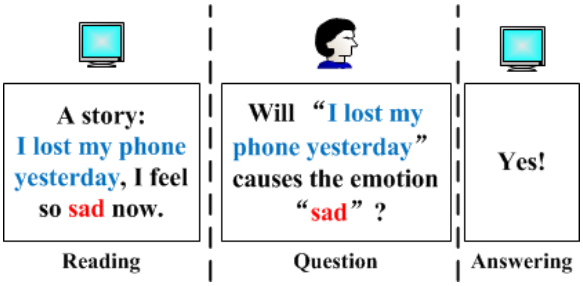

Figure 1: An example of emotion cause extraction based on the QA framework.

measures the importance of each word in the text by an attention mechanism. Based on the learned attention result, the network maps the text into a low dimensional vector space. This vector is then used to generate an answer. Existing memory network based approaches to QA use weighted sum of attentions to jointly consider short text segments stored in memory. However, they do not explicitly model sequential information in the context. In this paper, we propose a new deep memory network architecture to model the context of each word simultaneously by multiple memory slots which capture sequential information using convolutional operations (Kim, 2014), and achieves the state-of-the-art performance compared to existing methods which use manual rules, common sense knowledge bases or other machine learning models.

The rest of the paper is organized as follows. Section 2 gives a review of related works on emotion analysis. Section 3 presents our proposed deep memory network based model for emotion cause extraction. Section 4 discusses evaluation results. Finally, Section 5 concludes the work and outlines the future directions.

2 Related Work

Identifying emotion categories in text is one of the key tasks in NLP (Liu, 2015). Going one step further, emotion cause extraction can reveal important information about what causes a certain emotion and why there is an emotion change. In this section, we introduce related work on emotion analysis including emotion cause extraction.

In emotion analysis, we first need to determine the taxonomy of emotions. Researchers have proposed a list of primary emotions (Plutchik, 1980; Ekman, 1984; Turner, 2000). In this study, we adopt Ekman's emotion classification scheme (Ekman, 1984), which identifies six primary emotions, namely *happiness, sadness, fear, anger, disgust* and *surprise*, known as the "Big6" scheme in the W3C Emotion Markup Language. This emotion classification scheme is agreed upon by most previous works in Chinese emotion analysis.

Existing work in emotion analysis mostly focuses on emotion classification (Li et al., 2013; Zhou et al., 2016) and emotion information extraction (Balahur et al., 2013). Xu et al. (2012) used a coarse to fine method to classify emotions in Chinese blogs. Gao et al. (2013) proposed a joint model to co-train a polarity classifier and an emotion classifier. Beck et al. (2014) proposed a Multi-task Gaussian-process based method for emotion classification. Chang et al. (2015) used linguistic templates to predict reader's emotions. Das and Bandyopadhyay (2010) used an unsupervised method to extract emotion feelers from Bengali blogs. There are other studies which focused on joint learning of sentiments (Luo et al., 2015; Mohtarami et al., 2013) or emotions in tweets or blogs (Quan and Ren, 2009; Liu et al., 2013; Hasegawa et al., 2013; Qadir and Riloff, 2014; Ou et al., 2014), and emotion lexicon construction (Mohammad and Turney, 2013; Yang et al., 2014; Staiano and Guerini, 2014). However, the aforementioned work all focused on analysis of emotion expressions rather than emotion causes.

Lee et al. (2010) first proposed a task on emotion cause extraction. They manually constructed a corpus from the Academia Sinica Balanced Chinese Corpus. Based on this corpus, Chen et al. (2010) proposed a rule based method to detect emotion causes based on manually define linguistic rules. Some studies (Gui et al., 2014; Li and Xu, 2014; Gao et al., 2015) extended the rule based method to informal text in Weibo text (Chinese tweets).

Other than rule based methods, Russo et al. (2011) proposed a crowdsourcing method to construct a common-sense knowledge base which is related to emotion causes. But it is challenging to extend the common-sense knowledge base automatically. Ghazi et al. (2015) used Conditional Random Fields (CRFs) to extract emotion causes. However, it requires emotion cause and emotion keywords to be in the same sentence. More recently, Gui et al. (2016) proposed a multi-kernel based method to extract emotion causes through

learning from a manually annotated emotion cause dataset.

Most existing work does not consider the relation between an emotion word and the cause of such an emotion, or they simply use the emotion word as a feature in their model learning. Since emotion cause extraction requires an understanding of a given piece of text in order to correctly identify the relation between the description of an event which causes an emotion and the expression of that emotion, it can essentially be considered as a QA task. In our work, we choose the memory network, which is designed to model the relation between a story and a query for QA systems (Weston et al., 2014; Sukhbaatar et al., 2015). Apart from its application in QA, memory network has also achieved great successes in other NLP tasks, such as machine translation (Luong et al., 2015), sentiment analysis (Tang et al., 2016) or summarization (M. Rush et al., 2015). To the best of our knowledge, this is the first work which uses memory network for emotion cause extraction.

3 Our Approach

In this section, we will first define our task. Then, a brief introduction of memory network will be given, including its basic learning structure of memory network and deep architecture. Last, our modified deep memory network for emotion cause extraction will be presented.

3.1 Task Definition

The formal definition of emotion cause extraction is given in (Gui et al., 2016). In this task, a given document, which is a passage about an emotion event, contains an emotion word E and the cause of the event. The document is manually segmented in the clause level. For each clause $c = \{w_1, w_2, ...w_k\}$ consisting of k words, the goal is to identify which clause contains the emotion cause. For data representation, we can map each word into a low dimensional embedding space, a.k.a word vector (Mikolov et al., 2013). All the word vectors are stacked in a word embedding matrix $L \in \mathbb{R}^{d \times \|V\|}$, where d is the dimension of word vector and V is the vocabulary size.

For example, the sentence, "I lost my phone yesterday, I feel so sad now." shown in Figure 1, consists of two clauses. The first clause contains the emotion cause while the second clause expresses the emotion of sadness. Current methods to emotion cause extraction cannot handle complex sentence structures where the expression of an emotion and its cause are not adjacent. We envision that the memory network can better model the relation between a emotion word and its emotion causes in such complex sentence structures. In our approach, we only select the clause with the highest probability to be the emotion cause in each document.

3.2 Memory Network

We first present a basic memory network model for emotion cause extraction (shown in Figure 2). Given a clause $c = \{w_1, w_2, ..., w_k\}$, and an emotion word, we first obtain the emotion word's representation in an embedding space, denoted by E. For the clause, let the embedding representations of the words be denoted by $e_1, e_2, ..., e_k$. Here, both e_i and E are defined in $\mathbb{R}^d$. Then, we use the inner product to evaluate the correlation between each word i in a clause and the emotion word, denoted as m_i:

$$m_i = e_i \cdot E. \tag{1}$$

We then normalize the value of m_i to $[0, 1]$ using a softmax function, denoted by α_i as:

$$\alpha_i = \frac{\exp(m_i)}{\sum_{j=1}^{k} \exp(m_j)}, \tag{2}$$

where k is the length of the clause. k also serves as the size of the memory. Obviously, $\alpha_i \in [0, 1]$ and $\sum_{i=1}^{k} \alpha_i = 1$. α_i can serve as an attention weight to measure the importance of each word in our model.

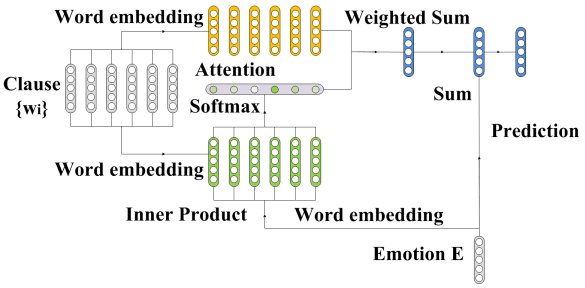

Figure 2: A single layer memory network.

Then, a sum over the word embedding e_i, weighted by the attention vector form the output of the memory network for the prediction of o:

$$o = \sum_{i=1}^{k} e_i \cdot \alpha_i + E. \tag{3}$$

The final prediction is an output from a softmax function, denoted as $\hat{o}$:

$$\hat{o} = \text{softmax}\left(W^T o\right). \quad (4)$$

Usually, W is a $d \times d$ weight matrix and T is the transposition. Since the answer in our task is a simple "yes" or "no", we use a $d \times 1$ matrix for W. As the distance between a clause and an emotion words is a very important feature according to (Gui et al., 2016), we simply add this distance into the softmax function as an additional feature in our work.

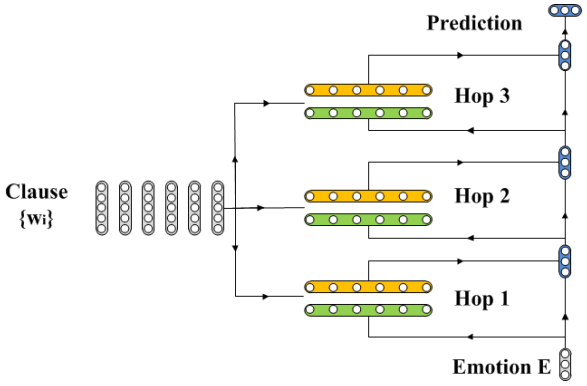

Figure 3: Deep memory network with three computational layers (hops).

The basic model can be extended to deep architecture consisting of multiple layers to handle L hop operations. The network is stacked as follows:

- For hop 1, the query is E and the prediction vector is o_1;
- For hop i, the query is the prediction vector of the previous hop and the prediction vector is o_i;
- The output vector is at the top of the network. It is a softmax function on the prediction vector from hop L: $\hat{o} = \text{softmax}\left(W^T o_L\right)$.

The illustration of a deep memory network with three layers is shown in Figure 3. Since a memory network models the emotion cause at a fine-grained level, each word has a corresponding weight to measure its importance in this task. Comparing to previous approaches in emotion cause extraction which are mostly based on manually defined rules or linguistic features, a memory network is a more principled way to identify the emotion cause from text. However, the basic memory network model does not capture the sequential information in context which is important in emotion cause extraction.

3.3 Convolutional Multiple-Slot Deep Memory Network

It is often the case that the meaning of a word is determined by its context, such as the previous word and the following word. Also, negations and emotion transitions are context sensitive. However, the memory network described in Section 3.2 has only one memory slot with size $d \times k$ to represent a clause, where d is the dimension of a word embedding and k is the length of a clause. It means that when the memory network models a clause, it only considers each word separately.

In order to capture context information for clauses, we propose a new architecture which contains more memory slot to model the context with a convolutional operation. The basic architecture of Convolutional Multiple-Slot Memory Network (in short: ConvMS-Memnet) is shown in Figure 4.

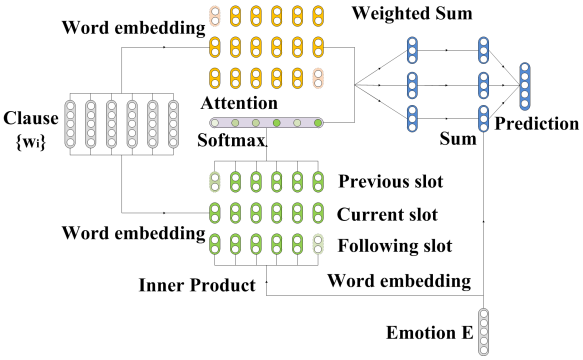

Figure 4: A single layer ConvMS-Memnet.

Considering the text length is usually short in the dataset used here for emotion cause extraction, we set the size of the convolutional kernel to 3. That is, the weight of word w_i in the i-th position considers both the previous word w_{i-1} and the following word w_{i+1} by a convolutional operation:

$$m'_i = \sum_{j=1}^{3} e_{i-2+j} \cdot E \quad (5)$$

For the first and the last word in a clause, we use zero padding, $w_0 = w_{k+1} = \vec{0}$, where k is the

length of a clause. Then, the attention weight for each word position in the clause is now defined as:

$$\alpha'_i = \frac{\exp(m'_i)}{\sum_{j=1}^{k} \exp\left(m'_j\right)} \tag{6}$$

Note that we obtain the attention for each position rather than each word. It means that the corresponding attention for the i-th word in the previous convolutional slot should be α_{i+1}. Hence, there are three prediction output vectors, namely, $o_{previous}$, $o_{current}$, $o_{following}$:

$$o_{previous} = \sum_{i=1}^{k} e_{i-1} \cdot \alpha'_i + E \tag{7}$$

$$o_{current} = \sum_{i=1}^{k} e_i \cdot \alpha'_i + E \tag{8}$$

$$o_{following} = \sum_{i=1}^{k} e_{i+1} \cdot \alpha'_i + E \tag{9}$$

At last, we concatenate the three vectors as $o = o_{previous} \bigoplus o_{current} \bigoplus o_{following}$ for the prediction by a softmax function:

$$\hat{o} = \text{softmax}\left(W_m^T o\right) \tag{10}$$

Here, the size of W_m is $(3 \cdot d) \times d$. Since the prediction vector is a concatenation of three outputs. We implement a concatenation operation rather than averaging or other operations because the parameters in different memory slots can be updated by back propagation. The concatenation of three output vectors forms a sequence-level feature which can be used in the training. Such a feature is important especially when the size of annotated training data is small.

For deep architecture with multiple layer training, the network is more complex (shown in Figure 5).

- For the first layer, the query is an embedding of the emotion word, E.
- In the next layer, there are three input queries since the previous layer has three outputs: $o^1_{previous}$, $o^1_{current}$, $o^1_{following}$. So, for the j-th layer ($j \neq 1$), we need to re-define the weight function (5) as:

$$m'_i = e_{i-1} \cdot o^{j-1}_{previous} + e_i \cdot o^{j-1}_{current} + e_{i+1} \cdot o^{j-1}_{following} \tag{11}$$

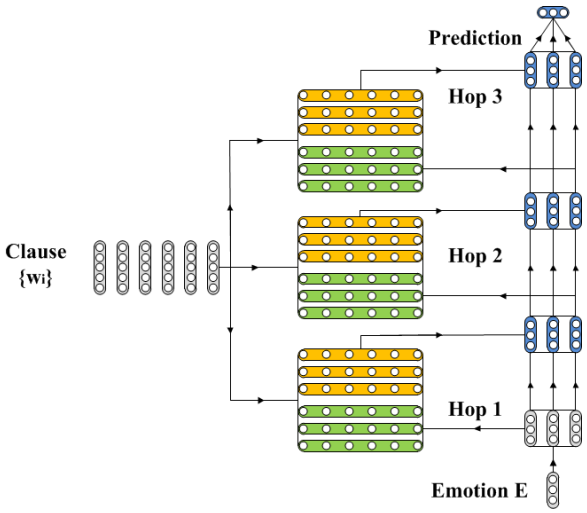

Figure 5: ConvMS-Memnet with three computational layers (hops).

- In the last layer, the concatenation of the three prediction vectors form the final prediction vector to generate the answer.

For model training, we use stochastic gradient descent and back propagation to optimize the loss function. Word embeddings are learned using a skip-gram model. The size of the word embedding is 20 since the vocabulary size in our dataset is small. The dropout is set to 0.4.

4 Experiments and Evaluation

We first presents the experimental settings and then report the results in this section.

4.1 Experimental Setup and Dataset

We conduct experiments on a simplified Chinese emotion cause corpus (Gui et al., 2016)*, the only publicly available dataset on this task to the best of our knowledge. The corpus contains 2,105 documents from SINA city news†. Each document has only one emotion word and one or more emotion causes. The documents are segmented into clauses manually. The main task is to identify which clause contains the emotion cause.

Details of the corpus are shown in Table 1. The metrics we used in evaluation follows Lee et al. (2010). It is commonly accepted so that we can compare our results with others. If a proposed emotion cause clause covers the annotated answer, the word sequence is considered correct. The precision, recall, and F-measure are defined by

*Available at: http://hlt.hitsz.edu.cn/?page_id=694
†http://news.sina.com.cn/society/

Item	Number
Documents	2,105
Clauses	11,799
Emotion Causes	2,167
Documents with 1 emotion	2,046
Documents with 2 emotions	56
Documents with 3 emotions	3

Table 1: Details of the dataset.

$$P = \frac{\sum_{\text{correct causes}} 1}{\sum_{\text{proposed causes}} 1},$$
$$R = \frac{\sum_{\text{correct causes}} 1}{\sum_{\text{annotated causes}} 1},$$
$$F = \frac{2 \times P \times R}{P + R}.$$

In the experiments, we randomly select 90% of the dataset as training data and 10% as testing data. In order to obtain statistically credible results, we evaluate our method and baseline methods 25 times with different train/test splits.

4.2 Evaluation and Comparison

We compare with the following baseline methods:

- **RB** (Rule based method): The rule based method proposed in (Lee et al., 2010).
- **CB** (Common-sense based method): This is the knowledge based method proposed by (Russo et al., 2011). We use the Chinese Emotion Cognition Lexicon (Xu et al., 2013) as the common-sense knowledge base. The lexicon contains more than 5,000 kinds of emotion stimulation and their corresponding reflection words.
- **RB+CB+ML** (Machine learning method trained from rule-based features and facts from a common-sense knowledge base): This methods was previously proposed for emotion cause classification in (Chen et al., 2010). It takes rules and facts in a knowledge base as features for classifier training. We train a SVM using features extracted from the rules defined in (Lee et al., 2010) and the Chinese Emotion Cognition Lexicon (Xu et al., 2013).
- **SVM**: This is a SVM classifier using the unigram, bigram and trigram features. It is a baseline previously used in (Li and Xu, 2014; Gui et al., 2016)

Method	P	R	F
RB	**0.6747**	0.4287	0.5243
CB	0.2672	**0.7130**	0.3887
RB+CB	0.5435	0.5307	0.5370
RB+CB+ML	0.5921	0.5307	0.5597
SVM	0.4200	0.4375	0.4285
Word2vec	0.4301	0.4233	0.4136
CNN	0.6215	0.5944	0.6076
Multi-kernel	0.6588	0.6927	**0.6752**
Memnet	0.5922	0.6354	0.6131
ConvMS-Memnet	**0.7076**	**0.6838**	**0.6955**

Table 2: Comparison with existing methods.

- **Word2vec**: This is a SVM classifier using word representations learned by Word2vec (Mikolov et al., 2013) as features.
- **Multi-kernel**: This is the state-of-the-art method using the multi-kernel method (Gui et al., 2016) to identify the emotion cause. We use the best performance reported in their paper.
- **CNN**: The convolutional neural network for sentence classification (Kim, 2014).
- **Memnet**: The deep memory network described in Section 3.2. Word embeddings are pre-trained by skip-grams. The number of hops is set to 3.
- **ConvMS-Memnet**: The convolutional multiple-slot deep memory network we proposed in Section 3.3. Word embeddings are pre-trained by skip-grams. The number of hops is 3 in our experiments.

Table 2 shows the evaluation results. The rule based RB gives fairly high precision but with low recall. CB, the common-sense based method, achieves the highest recall. Yet, its precision is the worst. RB+CB, the combination of RB and CB gives higher the F-measure But, the improvement of 1.27% is only marginal compared to RB.

For machine learning methods, RB+CB+ML uses both rules and common-sense knowledge as features to train a machine learning classifier. It achieves F-measure of 0.5597, outperforming RB+CB. Both SVM and word2vec are word feature based methods and they have similar performance. For word2vec, even though word representations are obtained from the SINA news raw corpus, it still performs worse than SVM trained using n-gram features only. The multi-kernel method (Gui et al., 2016) is the best performer

Word Embedding	P	R	F
Pre-trained	**0.7076**	**0.6838**	**0.6955**
Randomly initialized	0.6786	0.6608	0.6696

Table 3: Comparison of using pre-trained or randomly initialized word embedding.

Method	P	R	F
Hop 1	0.6597	0.6444	0.6520
Hop 2	0.6877	0.6718	0.6796
Hop 3	**0.7076**	**0.6838**	**0.6955**
Hop 4	0.6882	0.6722	0.6801
Hop 5	0.6763	0.6606	0.6683
Hop 6	0.6664	0.6509	0.6585
Hop 7	0.6483	0.6333	0.6407
Hop 8	0.6261	0.6116	0.6187
Hop 9	0.6161	0.6109	0.6089

Table 4: Performance with different number of hops in ConvMS-Memnet.

among the baselines because it considers context information in a structured way. It models text by its syntactic tree and also considers an emotion lexicon. Their work shows that the structure information is important for the emotion cause extraction task.

Naively applying the original deep memory network or convolutional network for emotion cause extraction outperforms all the baselines except the convolutional multi-kernel method. However, using our proposed ConvMS-Memnet architecture, we manage to boost the performance by 11.54% in precision, 4.84% in recall and 8.24% in F-measure respectively when compared to Memnet. The improvement is very significant with p-value less than 0.01 in t-test. The ConvMS-Memnet also outperforms the previous best-performing method, multi-kernel, by 3.01% in F-measure. It shows that by effectively capturing context information, ConvMS-Memnet is able to identify the emotion cause better compared to other methods.

4.3 More Insights into the ConvMS-Memnet

To gain better insights into our proposed ConvMS-Memnet, we conduct further experiments to understand the impact on performance by using: 1) pre-trained or randomly initialized word embedding; 2) multiple hops; 3) attention visualizations; 4) more training epochs.

4.3.1 Pre-trained Word Embeddings

In our ConvMS-Memnet, we use pre-trained word embedding as the input. The embedding maps each word into a lower dimensional real-value vector as its representation. Words sharing similar meanings should have similar representations. It enables our model to deal with synonyms more effectively.

The question is, "can we train the network without using pre-trained word embeddings?". We initialize word vectors randomly, and use an embedding matrix to update the word vectors in the training of the network simultaneously. Comparison results are shown in Table 3. It can be observed that pre-trained word embedding gives 2.59% higher F-measure compared to random initialization. This is partly due to the limited size of our training data. Hence using word embedding trained from other much larger corpus gives better results.

4.3.2 Multiple Hops

It is widely acknowledged that computational models using deep architecture with multiple layers have better ability to learn data representations with multiple levels of abstractions. In this section, we evaluate the power of multiple hops in this task. We set the number of hops from 1 to 9 with 1 standing for the simplest single layer network shown in Figure 4. The more hops are stacked, the more complicated the model is. Results are shown in Table 4. The single layer network has achieved a competitive performance. With the increasing number of hops, the performance improves. However, when the number of hops is larger than 3, the performance decreases due to overfitting. Since the dataset for this task is small, more parameters will lead to overfitting. As such, we choose 3 hops in our final model since it gives the best performance in our experiments.

4.3.3 Word-Level Attention Weights

Essentially, memory network aims to measure the weight of each word in the clause with respect to the emotion word. The question is, will the model really focus on the words which describe the emotion cause? We choose one example to show the attention results in Table 5:

Ex.2 家人*/family* 的*/'s* 坚持*/insistence* 更*/more* 让*/makes* 人*/people* 感动*/touched*

In this example, the cause of the emotion "touched" is "insistence". We show in Table 5 the distribution of word-level attention weights in different hops of memory network training. We can observe that in the first two hops, the highest attention weights centered on the word "more". However, from the third hop onwards, the highest atten-

previous slot	**current slot**	following slot	**Hop 1**	**Hop 2**	**Hop 3**	**Hop 4**	**Hop 5**
家人/family	的/'s	坚持/insisting	0.1298	0.3165	0.1781	0.2947	0.1472
的/'s	坚持/insistence	更/more	0.1706	0.2619	**0.7346**	**0.6412**	**0.8373**
坚持/insisting	更/more	让/makes	**0.5090**	**0.3070**	0.0720	0.0553	0.0145
更/more	让/makes	人/people	0.0327	0.0139	0.0001	0.0001	0.0000
让/makes	人/people	感动/touched	0.1579	0.0965	0.0145	0.0080	0.0008

Table 5: The distribution of attention in different hops.

Method	**P**	**R**	**F**
Memnet	0.5688	0.5588	0.5635
ConvMS-Memnet	**0.6250**	**0.6140**	**0.6195**

Table 6: Comparison of word level emotion cause extraction.

tion weight moves to the word sub-sequence centred on the word "insistence". This shows that our model is effective in identifying the most important keyword relating to the emotion cause. Also, better results are obtained using deep memory network trained with at least 3 hops. This is consistent with what we observed in Section 4.3.2.

In order to evaluate the quality of keywords extracted by memory networks, we define a new metric on the keyword level of emotion cause extraction. The keyword is defined as the word which obtains the highest attention weight in the identified clause. If the keywords extracted by our algorithm is located within the boundary of annotation, it is treated as correct. Thus, we can obtain the precision, recall, and F-measure by comparing the proposed keywords with the correct keywords by:

$$P = \frac{\sum_{\text{correct keywords}} 1}{\sum_{\text{proposed keywords}} 1},$$
$$R = \frac{\sum_{\text{correct keywords}} 1}{\sum_{\text{annotated keywords}} 1},$$
$$F = \frac{2 \times P \times R}{P + R}.$$

Since the reference methods do not focus on the keywords level, we only compare the performance of Memnet and ConvMS-Memnet in Table 6. It can be observed that our proposed ConvMS-Memnet outperforms Memnet by 5.6% in F-measure. It shows that by capturing context features, ConvMS-Memnet is able to identify the word level emotion cause better compare to Memnet.

4.3.4 Training Epochs

In our model, the training epochs are set to 20. In this section, we examine the testing error using a case study. Due to the page length limit, we only choose one example from the corpus. The text below has four clauses:

Ex.3 *45*天，对于失去儿子的他们是多么的漫长，宝贝回家了，这个春节是多么幸福。

Ex.3 *45 days, it is long time for the parents who lost their baby. If the baby comes back home, they would become so happy in this Spring Festival.*

In this example, the cause of emotion "happy" is described in the third clause.

We show in Table 7 the probability of each clause containing an emotion cause in different training epochs. It is interesting to see that our model is able to detect the correct clause with only 5 epochs. With the increasing number of training epochs, the probability associated with the correct clause increases further while the probabilities of incorrect clauses decrease generally.

4.4 Limitations

We have shown in Section 4.3.4 a simple example consisting of only four clauses from which our model can identify the clause containing the emotion cause correctly. We notice that for some complex text passages which contain long distance dependency relations, negations or emotion transitions, our model may have a difficulty in detecting the correct clause containing the emotion causes. It is a challenging task to properly model the discourse relations among clauses. In the future, we will explore different network architecture with consideration of various discourse relations possibly through transfer learning of larger annotated data available for other tasks.

Another shortcoming of our model is that, the answer generated from our model is simply "yes" or "no". The main reason is that the size of the annotated corpus is too small to train a model which can output natural language answers in full sentences. Ideally, we would like to develop a model which can directly give the cause of an emotion

Clause	5 Epochs	10 Epochs	15 Epochs	20 Epochs
45 Days	0.0018	0.0002	0.0000	0.0000
it is ... baby	0.3546	0.6778	0.5457	0.3254
If the ... back home	**0.7627**	**0.7946**	**0.8092**	**0.9626**
they ... Spring Festival	0.2060	0.0217	0.0004	0.0006

Table 7: The probability of a clause containing the emotion cause in different iterations in the multiple-slot memory network.

expressed in text. However, since the manual annotation of data is too expensive for this task, we need to explore feasible ways to automatically collect annotate data for emotion cause detection. We also need to study effective evaluation mechanisms for such QA systems.

5 Conclusions

In this work, we treat emotion cause extraction as a QA task and propose a new model based on deep memory networks for identifying the emotion causes for an emotion expressed in text. The key property of this approach is the use of context information in the learning process which is ignored in the original memory network. Our new memory network architecture is able to store context in different memory slots to capture context information in proper sequence by convolutional operation. Our model achieves the state-of-the-art performance on a dataset for emotion cause detection when compared to a number of competitive baselines. In the future, we will explore effective ways to model discourse relations among clauses and develop a QA system which can directly output the cause of emotions as answers.

Acknowledgments

This work was supported by the National Natural Science Foundation of China 61370165, U1636103, 61632011, 61528302, National 863 Program of China 2015AA015405, Shenzhen Foundational Research Funding JCYJ20150625142543470, JCYJ20170307150024907 and Guangdong Provincial Engineering Technology Research Center for Data Science 2016KF09.

References

Alexandra Balahur, Jesús M. Hermida, Andrés Montoyo, and Rafael Muñoz. 2011. Emotinet: A knowledge base for emotion detection in text built on the appraisal theories. In *Proceedings of International Conference on Applications of Natural Language to Information Systems*, pages 27–39.

Alexandra Balahur, Jesús M. Hermida, and Hristo Tanev. 2013. Detecting implicit emotion expressions from text using ontological resources and lexical learning. *Theory and Applications of Natural Language Processing*, pages 235–255.

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task gaussian processes. In *EMNLP*, pages 1798–1803.

Yung-Chun Chang, Cen-Chieh Chen, Yu-Lun Hsieh, and WL Hsu. 2015. Linguistic template extraction for recognizing reader-emotion and emotional resonance writing assistance. In *ACL*, pages 775–780.

Wei Fan Chen, Mei Hua Chen, Ming Lung Chen, and Lun Wei Ku. 2016. A computer-assistance learning system for emotional wording. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1–1.

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *COLING*, pages 179–187.

Dipankar Das and Sivaji Bandyopadhyay. 2010. Finding emotion holder from bengali blog texts—an unsupervised syntactic approach. In *Proceedings of Pacific Asia Conference on Language, Information and Computation*, pages 621–628.

Paul Ekman. 1984. Expression and the nature of emotion. *Approaches to Emotion*, 3:19–344.

Kai Gao, Hua Xu, and Jiushuo Wang. 2015. A rule-based approach to emotion cause detection for chinese micro-blogs. *Expert Systems with Applications*, 42(9):4517–4528.

Wei Gao, Shoushan Li, Sophia Yat Mei Lee, Guodong Zhou, and Chu-Ren Huang. 2013. Joint learning on sentiment and emotion classification. In *CIKM*, pages 1505–1508. ACM.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *Computational Linguistics and Intelligent Text Processing*, pages 152–165. Springer.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016. Event-driven emotion cause extraction with corpus construction. In *EMNLP*, pages 1639–1649.

Lin Gui, Li Yuan, Ruifeng Xu, Bin Liu, Qin Lu, and Yu Zhou. 2014. Emotion cause detection with linguistic construction in chinese weibo text. In *Natural Language Processing and Chinese Computing*, pages 457–464. Springer.

Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *ACL*, pages 964–972.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53. Association for Computational Linguistics.

Shoushan Li, Lei Huang, Rong Wang, and Guodong Zhou. 2013. Sentence-level emotion classification with label and context dependence. In *ACL*, pages 1045–1053.

Weiyuan Li and Hua Xu. 2014. Text-based emotion classification using emotion cause extraction. *Expert Systems with Applications*, 41(4):1742–1749.

Bing Liu. 2015. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Huanhuan Liu, Shoushan Li, Guodong Zhou, Chu-Ren Huang, and Peifeng Li. 2013. Joint modeling of news reader's and comment writer's emotions. In *ACL*, pages 511–515.

Kun-Hu Luo, Zhi-Hong Deng, Liang-Chen Wei, and Hongliang Yu. 2015. Jeam: A novel model for cross-domain sentiment classification based on emotion analysis. In *EMNLP*, pages 2503–2508.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Mitra Mohtarami, Man Lan, and Chew Lim Tan. 2013. Probabilistic sense sentiment similarity through hidden emotions. In *ACL*, pages 983–992.

Gaoyan Ou, Wei Chen, Tengjiao Wang, Zhongyu Wei, Binyang Li, Dongqing Yang, and Kam-Fai Wong. 2014. Exploiting community emotion for microblog event detection. In *EMNLP*, pages 1159–1168.

Robert Plutchik. 1980. Emotion: A psychoevolutionary synthesis.

Ashequl Qadir and Ellen Riloff. 2014. Learning emotion indicators from tweets: Hashtags, hashtag patterns, and phrases. In *EMNLP*, pages 1203–1209.

Changqin Quan and Fuji Ren. 2009. Construction of a blog emotion corpus for chinese emotional expression analysis. In *EMNLP*, pages 1446–1454.

Irene Russo, Tommaso Caselli, Francesco Rubino, Ester Boldrini, and Patricio Martínez-Barco. 2011. Emocause: an easy-adaptable approach to emotion cause contexts. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 153–160.

Jacopo Staiano and Marco Guerini. 2014. Depechemood: a lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605*.

Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. 2015. End-to-end memory networks. In *NIPS*, pages 2431–2439.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *EMNLP*, pages 214–225.

Jonathan H Turner. 2000. *On the origins of human emotions: A sociological inquiry into the evolution of human affect*. Stanford University Press Stanford, CA.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Jun Xu, Ruifeng Xu, Qin Lu, and Xiaolong Wang. 2012. Coarse-to-fine sentence-level emotion classification based on the intra-sentence features and sentential context. In *CIKM*, pages 2455–2458. ACM.

Ruifeng Xu, Chengtian Zou, Yanzhen Zheng, Xu Jun, Lin Gui, Bin Liu, and Xiaolong Wang. 2013. A new emotion dictionary based on the distinguish of emotion expression and emotion cognition. *Journal of Chinese Information Processing*, 27(6):82–90.

Min Yang, Dingju Zhu, and Kam-Pui Chow. 2014. A topic model for building fine-grained domain-specific emotion lexicon. In *ACL(2)*, pages 421–426.

Deyu Zhou, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. 2016. Emotion distribution learning from texts. In *EMNLP*, pages 638–647.