A Question Answering Approach to Emotion Cause Extraction

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.

Introduction

With the rapid growth of social network platforms, more and more people tend to share their experiences and emotions online.[2]Corresponding Author: xuruifeng@hit.edu.cn 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 BIBREF0 as well as readers' emotions BIBREF1 . There are also some information extraction tasks defined in emotion analysis BIBREF2 , BIBREF3 , such as extracting the feeler of an emotion BIBREF4 . 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 [id=lq]may or may not cause a certain emotion, we take [id=lq]an emotion word [id=lq]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 [id=lq]expected answer [id=lq]is either “yes” or “no”. (see Figure FIGREF1). We build our QA system based on a deep memory network. The memory network has two inputs: a piece of text, [id=lq]referred to as a story in QA systems, and a query. The [id=lq]story is represented using a sequence of word embeddings.

[id=lq]A recurrent structure is implemented to mine the deep relation between a query and a text. It measure[id=lq]s the [id=lq]importance of each word in the text by [id=lq]an attention mechanism. Based on the [id=lq]learned attention result, the network maps the text into a low dimensional vector space. This vector is [id=lq]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 [id=lq]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 BIBREF5 , 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 SECREF2 gives a review of related works on emotion analysis. Section SECREF3 presents our proposed deep memory network based model for emotion cause extraction. Section SECREF4 discusses evaluation results. Finally, Section SECREF5 concludes the work and outlines the future directions.

Related Work

Identifying emotion categories in text is one of the key tasks in NLP BIBREF6 . 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 BIBREF7 , BIBREF8 , BIBREF9 . In this study, we adopt Ekman's emotion classification scheme BIBREF8 , which identifies six primary emotions, namely happiness, sadness, fear, anger, disgust and surprise, as 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 BIBREF10 , BIBREF11 and emotion information extraction BIBREF12 . xu2012coarse used a coarse to fine method to classify emotions in Chinese blogs. gao2013joint proposed a joint model to co-train a polarity classifier and an emotion classifier. beck2014joint proposed a Multi-task Gaussian-process based method for emotion classification. chang2015linguistic used linguistic templates to predict reader's emotions. das2010finding used an unsupervised method to extract emotion feelers from Bengali blogs. There are other studies which focused on joint learning of sentiments BIBREF13 , BIBREF14 or emotions in tweets or blogs BIBREF15 , BIBREF16 , BIBREF17 , BIBREF18 , BIBREF19 , and emotion lexicon construction BIBREF20 , BIBREF21 , BIBREF22 . However, the aforementioned work all focused on analysis of emotion expressions rather than emotion causes.

lee2010text first proposed a task on emotion cause extraction. They manually constructed a corpus from the Academia Sinica Balanced Chinese Corpus. Based on this corpus, chen2010emotion proposed a rule based method to detect emotion causes based on manually define linguistic rules. Some studies BIBREF23 , BIBREF24 , BIBREF25 extended the rule based method to informal text in Weibo text (Chinese tweets).

Other than rule based methods, russo2011emocause 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. ghazi2015detecting 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, gui2016event proposed a multi-kernel based method to extract emotion causes through learning from a manually annotated emotion cause dataset.

[id=lq]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 BIBREF26 , BIBREF27 . Apart from its application in QA, memory network has also achieved great successes in other NLP tasks, such as machine translation BIBREF28 , sentiment analysis BIBREF29 or summarization BIBREF30 . To the best of our knowledge, this is the first work which uses memory network for emotion cause extraction.

Our Approach

In this section, we will first define our task. [id=lq]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.

Task Definition

The formal definition of emotion cause extraction is given in BIBREF31 . In this task, a given document, which [id=lq]is a passage about an emotion event, contains an emotion word INLINEFORM0 and the cause of the event. The document is manually segmented in the clause level. For each clause INLINEFORM1 consisting of INLINEFORM2 words, the goal [id=lq]is to identify which clause contains the emotion cause. [id=lq]For data representation, we can map each word into a low dimensional embedding space, a.k.a word vector BIBREF32 . All the word vectors are stacked in a word embedding matrix INLINEFORM3 , where INLINEFORM4 is the dimension of word vector and INLINEFORM5 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 [id=lq]expresses the emotion of sadness. [id=lq]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 [id=lq]better model the relation between [id=lq]a emotion word and [id=lq]its emotion causes in such complex sentence structures. In our approach, we only select the clause with the highest probability to be [id=lq] thean emotion cause in each document.

Memory Network

We first present a basic memory network model for emotion cause extraction (shown in Figure 2). Given a clause INLINEFORM0 , and an emotion word, we [id=lq]first obtain the emotion word's representation in an embedding space[id=lq], denoted by INLINEFORM1 . For the clause, [id=lq]let the embedding representations of the words be denoted by INLINEFORM2 . Here, both INLINEFORM3 and INLINEFORM4 [id=lq]are defined in INLINEFORM5 . Then, we use the inner product to evaluate the correlation between each word [id=lq] INLINEFORM6 in a clause and the emotion word, denoted as INLINEFORM7 : DISPLAYFORM0

We then normalize the value of INLINEFORM0 to INLINEFORM1 using a softmax function, denoted by INLINEFORM2 [id=lq]as: DISPLAYFORM0

where INLINEFORM0 is the length of the clause. [id=lq] INLINEFORM1 also serves as the size of the memory. Obviously, INLINEFORM2 and INLINEFORM3 . [id=lq] INLINEFORM4 can serve as an attention weight to measure the importance of each word in our model.

Then, a sum over the word embedding INLINEFORM0 , weighted by the attention vector form the output of the memory network for the prediction of INLINEFORM1 : DISPLAYFORM0

The final prediction is an output from a softmax function, denoted as INLINEFORM0 : DISPLAYFORM0

Usually, INLINEFORM0 is a INLINEFORM1 weight matrix and INLINEFORM2 is the transposition. Since the answer in our task is a simple “yes” or “no”, we use a INLINEFORM3 matrix for INLINEFORM4 . As the distance between a clause and an emotion words is a very important feature according to BIBREF31 , we simply add this distance into the softmax function as an additional feature in our work.

The basic model can be extended to deep architecture consisting of multiple layers to handle INLINEFORM0 hop operations. The network is stacked as [id=lq]follows:

For hop 1, the query is INLINEFORM0 and the prediction vector is INLINEFORM1 ;

For hop INLINEFORM0 , the query is the prediction vector of the previous hop and the prediction vector is INLINEFORM1 ;

The output vector is at the top of the network. It is a softmax function on the prediction vector from hop INLINEFORM0 : INLINEFORM1 .

The illustration of a deep memory network with three layers is shown in Figure 3. Since [id=lq]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 [id=lq]to previous approaches [id=lq]in emotion cause extraction which are [id=lq]mostly based [id=lq]on manually defined rules or linguistic features, [id=lq]a memory network is a more principled way to identify the emotion cause from text. However, the basic [id=lq]memory network model [id=lq]does not capture the sequential information in context which is important in emotion cause extraction.

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. [id=lq]Also, negations and emotion transitions are context sensitive. However, the memory network described in Section SECREF3 has only one memory slot with size INLINEFORM0 to represent a clause, where INLINEFORM1 is the dimension of a word embedding and INLINEFORM2 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 [id=lq]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.

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 INLINEFORM0 [id=lq]in the INLINEFORM1 -th position considers both the previous word INLINEFORM2 and the following word INLINEFORM3 by a convolutional operation: DISPLAYFORM0

For the first and the last word in a clause, we use zero padding, INLINEFORM0 , where INLINEFORM1 is the length of a clause. Then, the attention [id=lq]weightsignal for each word position in the clause is [id=lq]now defined as: DISPLAYFORM0

Note that we obtain the attention for each position rather than each word. It means that the corresponding attention for the INLINEFORM0 -th word in the previous convolutional slot should be INLINEFORM1 . Hence, there are three prediction output vectors, namely, INLINEFORM2 , INLINEFORM3 , INLINEFORM4 : DISPLAYFORM0

At last, we concatenate the three vectors as INLINEFORM0 for the prediction by a softmax function:

DISPLAYFORM0

Here, the size of INLINEFORM0 is INLINEFORM1 . 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 [id=lq]respectively in this way 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 [id=lq]when the size of annotated training data is small.

For deep architecture with multiple layer[id=lq]s training, the network is more [id=lq]complex (shown in Figure 5).

For the first layer, the query is an embedding of the emotion word, INLINEFORM0 .

In the next layer, there are three input queries since the previous layer has three outputs: INLINEFORM0 , INLINEFORM1 , INLINEFORM2 . So, for the INLINEFORM3 -th layer (INLINEFORM4), we need to re-define the weight function (5) as:

In the last layer, [id=lq]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.

Experiments and Evaluation

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

Experimental Setup and Dataset

We conduct experiments on a simplified Chinese emotion cause corpus BIBREF31 , 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.

[id=lq]Details of the corpus are shown in Table 1. The metrics we used in evaluation follows lee2010text. 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 INLINEFORM0

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.

Evaluation and Comparison

We compare with the following baseline methods:

RB (Rule based method): The rule based method proposed in BIBREF33 .

CB (Common-sense based method): This is the knowledge based method proposed by BIBREF34 . We use the Chinese Emotion Cognition Lexicon BIBREF35 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 BIBREF36 . 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 BIBREF33 and the Chinese Emotion Cognition Lexicon BIBREF35 .

SVM: This is a SVM classifier using the unigram, bigram and trigram features. It is a baseline previously used in BIBREF24 , BIBREF31

Word2vec: This is a SVM classifier using word representations learned by Word2vec BIBREF32 as features.

Multi-kernel: This is the state-of-the-art method using the multi-kernel method BIBREF31 to identify the emotion cause. We use the best performance reported in their paper.

CNN: The convolutional neural network for sentence classification BIBREF5 .

Memnet: The deep memory network described in Section SECREF3 . 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 SECREF13 . 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 BIBREF31 is the best performer 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 INLINEFORM0 -value less than 0.01 in INLINEFORM1 -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.

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.

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.

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.

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 attention 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 UID45 .

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: INLINEFORM0

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.

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.

Limitations

We have shown in Section UID47 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 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.

Conclusions

In this [id=lq]work, we [id=lq]treat emotion cause extraction as a QA task and propose a new model based on deep memory networks for identifying [id=lq]the emotion causes for an emotion expressed in text. [id=lq]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 [id=lq]memory network architecture is able [id=lq]to store context in different memory slots to capture context information [id=lq]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 [id=lq]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.