Future Generation Computer Systems 92 (2019) 879–888

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Feature-based Compositing Memory Networks for Aspect-based Sentiment Classification in Social Internet of Things

Ruixin Ma [a], Kai Wang [a], Tie Qiu [a,*], Arun Kumar Sangaiah [b], Dan Lin [a], Hannan Bin Liaqat [c]

[a] School of Software, Dalian University of Technology, Dalian 116620, China
[b] School of Computing Science and Engineering, VIT University, Vellore 632014, India
[c] Department of Information Technology, University of Gujrat, Gujrat, Pakistan

HIGHLIGHTS

- We extract location, part-of-speech and sentiment features to enrich word representations.
- We propose the feature-based compositing memory networks with three compositing strategies.
- We obtain state-of-the-art performance in Laptops and Restaurants datasets of SemEval 2014.

ARTICLE INFO

Article history:
Received 22 June 2017
Received in revised form 2 November 2017
Accepted 20 November 2017
Available online 7 December 2017

Keywords:
Sentiment analysis
Memory networks
Aspect-based Sentiment Classification
Attention mechanism

ABSTRACT

Sentiment analysis is an important research field in natural language processing. Aspect-based sentiment classification can efficiently solve fine-grained sentiment recognition, however, its classification accuracy becomes decreasing for large-scale corpus. To solve this problem, we propose a new memory network model, called Feature-based Compositing Memory Networks (FCMN). Differing from typical memory networks, we extract three kinds of features to enrich the word representation of each context word. We design compositing strategies combining feature representations and word embedding to improve the performance of attention mechanism. Experiments on laptops and restaurants datasets in SemEval 2014 show that our approach outperforms the feature-based SVM, TD-LSTM and Deep Memory Networks. Especially, FCMN gets better results with less hops than Deep Memory Networks. Experiments results demonstrate that FCMN can ignore words without sentiment and pay more attention on correct words in a sentence.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Internet of Things (IoT) covers a diverse range of fields with respect to sensor networks [1,2], embedded systems [3], intelligent control, data processing and fusion, task scheduling and allocation [4,5]. Recently, integrating social networking concepts into IoT solutions, Social Internet of Things (SIoT) has become a hot research topic [6,7]. The SIoT paradigm represents an ecosystem which allows people and smart objects to interact within a social structure of relationships [8].

In order to maintain user-friendliness and bridge human-to-machine perceptions in SIoT, Natural language processing (NLP) along with Machine Learning algorithms are applied for extracting the relevant signals from a users' search query or other natural language interaction with services [9]. With the help of NLP technology, smart objects have the potential to understand the user's point of view from natural language, especially the basic sentiment characteristics, such as positive and negative. However, for machines, it is still a challenge to accurately identify sentiment polarity from different texts, which limits the performance of smart objects when interacting with users.

That leads to the sentiment analysis problem in NLP field, increasingly concerned by scholars and IT enterprises [10]. An early task in sentiment analysis is classifying the overall polarity of a given text, whether the expressed sentiment is positive, negative, or neutral. However, with the development of Internet and E-commerce, many applications need to get more fine-grained sentiment polarity [11]. For example, different words in one commodity comment on the shopping website may have opposing sentiments. Considering the sentence "Great food but the service was dreadful", customer is satisfied with the 'food' while the sentiment for 'service' is negative. Therefore, an effective approach is researched to estimate sentiment polarity towards a particular aspect in the text, called Aspect-based Sentiment Classification (ABSC).

* Corresponding author.
E-mail addresses: maruixin@dlut.edu.cn (R. Ma), kai_wang@mail.dlut.edu.cn (K. Wang), qiutie@ieee.org (T. Qiu).

https://doi.org/10.1016/j.future.2017.11.036
0167-739X/© 2017 Elsevier B.V. All rights reserved.

To solve ABSC tasks, many machine learning algorithms are used to build a sentiment classifier in supervise or unsupervised manner [12]. One of representative methods is feature-based Support Vector Machine (SVM) [13], which utilizes engineering of features covering surface features, lexicon features and parse features. Although those approaches do work, the sparseness and discreteness of features restrict their performance. Instead of feature engineering, neural network models based on Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) generate continuous text representation to capture the relation between aspect terms and context words [14]. However, sequential operation of RNNs has the drawback that they cannot take different operations by the importance of each context word.

In recent years, attention mechanism and memory networks take considerable effect in question answering (QA) and other natural language processing tasks [15,16]. Inspired by memory networks, Tang et al. develop Deep Memory Network for aspect-based sentiment classification [17]. This approach converts the ABSC tasks into question answering directly and each memory unit stores only one word in sentence. However, there is still improvement room to focus on the peculiarity of ABSC tasks and exert the ability of memory networks better.

In our opinion, typical memory networks model cannot utilize its ability for ABSC tasks. The success of memory networks in QA tasks is partly determined by its facts reasoning ability which locates mentions of question in sentences and chooses the related sentence as facts to generate new question in single hop. The facts reasoning mechanism depends on long-text inputs [18] or short-text inputs with limited vocabulary [19]. However, the situation of ABSC tasks is opposite. For example, laptops dataset from SemEval 2014 [20] have more than 3000 vocabularies, but each review about laptops only has 21.5 words on average.

Because of the special situation, only word embedding used as external memory in memory networks is not enough. Short sentence and diverse vocabulary make the embedding representation only contains scarce and scattered information. It is difficult to fit an effective function to gather sentiment evidences from those memory units. Therefore, an intuitive thought is that enriching input resources to generate a more effective representation for each context word. This thought is inspired by the cognitive characteristic of humans. When humans are asked to do this task, they can utilize related memory to judge sentiment polarity. For one word, humans can recall its general meaning and usual usage, which are helpful to understand the whole sentence. Machines also need those word information instead of just computing similarity for different words through low-dimensional vectors.

Based on the above ideas, we propose a new neural network model, called Feature-based Compositing Memory Networks (FCMN). While iterating in multiple computational layers (hops), our approach computes context representation for each word consisting of embedding vectors and multi-angle features by using compositing strategies. The features of words include location features, part of speech (POS) features and sentiment features. Each layer in FCMN computes attention scores between aspect representation and every context word. Then attention scores are used to capture the evidences from memory to build new aspect representation. The aspect representation of the last layer inputs into a linear layer classifier for final sentiment classification. Except the pre-processing of feature extracting, the FCMN model is trained end-to-end with Adaptive Gradient (AdaGrad) algorithm [21] and the loss function is based on cross-entropy error.

Our approach has been verified in laptops and restaurants datasets from SemEval 2014, and precedes the existing state-of-art approaches including feature-based SVM and DMN model proposed by Tang et al. Compared with DMN, FCMN can get higher accuracy using less hops. According to the analysis of attention mechanism in DMN and FCMN, our approach has better performance in distributing attention scores and the ability of ignoring words without sentiment. The main contributions of the paper are as follows:

- We choose three kinds of word features to enrich word representations, including location features, part-of-speech features and sentiment features. And we design method to extract them and compute a feature representation used in memory networks.
- We propose the feature-based compositing memory networks with three different compositing strategies. The performance of the memory network in solving aspect-based sentiment classification has been improved.
- We obtain state-of-the-art performance for aspect-based sentiment classification in Laptops and Restaurants datasets of SemEval 2014.

2. Related work

2.1. Attention mechanism and memory networks

Recently, computational models based attention mechanism and explicit memory have shown great success in many NLP tasks, such as machine translation [22], question answering and sentiment classification [23]. Attention mechanism computes an attention store/weight to each lower position and combines those weighted positions to build one upper level representation, which can improve the performance of machine learning models [24].

Neural networks with a memory capacity provide a promising approach to natural language processing. Gated recurrent units (GRU) and LSTM-based models can be regarded as internal memory-based architectures, addressing memory through local memory cells which lock in the network state from the past [25].

Recent researchers focus on extending deep neural networks with external memory, such as the Neural Turing Machine which uses a continuous memory representation with both content and address-based access [26]. Weston et al. propose a neural networks based framework called Memory Networks (MemNNs) [27], which is designed with non-writable memories, and construct layered memory representations. Based on MemNNs, end-to-end memory networks (MemN2N) [28] can be trained end-to-end and possess a capacity of facts reasoning to solve complex question answering problems. Another model using external memory is Dynamic Memory Network [29] which is equipped with an episodic memory and shows promising results on both question answering and sentiment analysis tasks.

Inspired by the recent success of MemNNs, our model enriches the typical word embedding by integrating context features and improves the ability of memory networks for sentiment analysis.

2.2. Aspect-based sentiment classification

The goal of aspect-based sentiment classification is to detect sentiment expressed towards a given aspect term [20]. Some of the earlier approaches build sentence representations using features extracted from syntactic parser or external sentiment lexicons. For instance, feature-based SVM [13] trains a classifier with effective feature templates. In recent years, more researchers utilize the capability of neural model, like RNNs and LSTM, in learning continuous text representation [30]. AdaRNN uses the dependency parsing results to find words syntactically connected with the interested target [31]. Nguyen et al. propose PhraseRNN to identify sentiment of the aspect in sentences taking both dependency and constituent trees into account [32]. Tang et al. put forward target-dependent LSTM (TD-LSTM) approach, which selects the relevant parts of context to infer sentiment polarity towards the target [33].

The latest state-of-the-art approach is deep memory networks by Tang et al. [17], which exerts the reasoning ability of memory networks partly and gets promising performance in ABSC tasks. To Improve this approach, our approach integrates three kinds of context features into memory network model and improves the architecture of attention component by using different compositing strategies.

3. Models

In this section, we describe the feature-based compositing memory networks in detail. We first work out a definition of aspect-based sentiment classification tasks. Then we briefly introduce the end-to-end memory networks approach, which is the basis framework of our approach. After the overview of FCMN, we emphatically introduce the feature choosing and compositing strategy.

3.1. Task definition and notation

Aspect-based sentiment classification is vital in sentiment analysis. For sentence $S = \{w_1, w_2, \ldots, w_n\}$ and an aspect term $A = \{a_1, a_2, \ldots, a_m\}(0 < m < n)$, words in aspect term are contained in S ($A \subseteq S$). The aim of ABSC tasks is to classify the sentiment polarity ('positive', 'negative' and 'neutral') for every (S, A) pair. For example, the polarity of ("Did not enjoy the new Windows 8 and touchscreen functions.", "Windows 8") is 'negative', and that of ("I am still in the process of learning about its features", 'features') is 'neutral'.

3.2. End-to-end memory networks

We use QA task as example to explain the architecture of end-to-end memory networks (MemN2N) [28]. Different from ABSC task, the QA task aims to get answer from a list of sentences (story) to solve the question. Having great performance in QA tasks, MemN2N can be divided into two components: facts reasoning and answer predicting. Facts reasoning component finds facts from given sentences and question iteratively in multiple computational layers (hops). External memory used in each hop is comprised of a set of input and output memory representations, in which each memory unit contains one sentence's information. Answer predicting computes final answer by processing the facts from the last hop of facts reasoning.

In each hop of facts reasoning, sentences are encoded into d-dimensional memory vectors which are used as input memory representations. Similarly, a question representation is calculated by question and utilized to compute the relevance score with each input memory unit. Afterwards, the score vector as attention weight computes a weighted sum with output memory representations which are encoded in same way with input memory using different embedding matrix. The sum vector is regarded as the fact representation in each hop. Then new question representation is constructed by the sum of the facts and old question, as the output of this hop. The output of hop k is used as the question input of hop $k+1$, the last output is passed through answer predicting which contains a final weight matrix W and a Softmax to produce the predicted label.

In some follow-up work of MemN2N, the input and output memory are merged into one. To solve ABSC tasks, we make several modifications at the basic architecture of MemN2N following Dynamic Memory Networks and Deep Memory Networks by Tang et al. The detailed architecture of our approach is described in following sections.

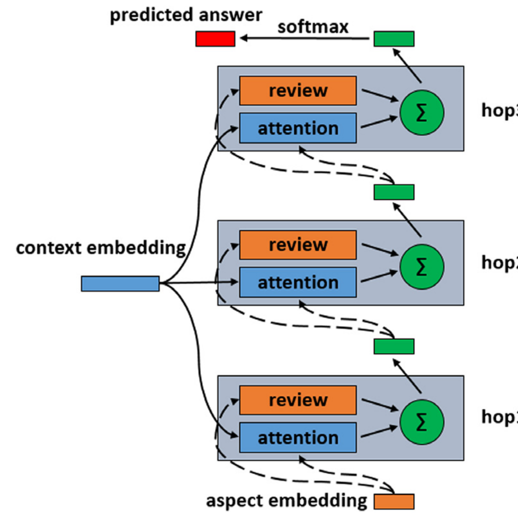


Fig. 1. An illustration of basic model of FCMN without composite strategies. This is a three layers (hops) model in which the context and aspect embedding are encoded from context words and aspect words respectively.

3.3. An overview of FCMN

In this part, we make an overview of feature-based compositing memory networks. As described in Section 1, it is hard in ABSC tasks to find related facts in which every memory unit only contains one word's embedding vector. Thus, our approach aims to enrich the input resources as much as possible by combining with different kinds of word features.

Given a sentence and aspect term for one task, we first separate the sentence into aspect words and context words. Then both of them are embedded into vectors of dimension d by an embedding matrix (of size $d \times V$, V is vocabulary size). Subsequently, we extract three kinds of features from the context words, and convert them into d-dimension vectors. In order to utilize the word features in Memory Networks, three compositing strategies are designed to composite context features and context embedding before, during, after the attention component respectively.

Without compositing strategies, the basic model of FCMN is similar with Deep Memory Networks, as shown in Fig. 1. The FCMN has multiple layers (hops), each layer has an attention component and a review component. The sum of outputs from two component is considered as the input of next layer. Different layers have shared parameters and external memory. In each hop, new facts are extracted from memory according to the aspect representation, which in the first layer is aspect embedding vector. The output vector of last layer contains the whole facts about aspect sentiment polarity, and is used to predict the final answer for aspect-based sentiment classification. The model can be trained end-to-end in supervised way.

3.4. Input representation and feature extracting

The input representation contains context embedding, aspect embedding and context features. We regard the context embedding and context features as external memory in MemNNs, which has a description of multiple perspectives for a single word. Both of them are immovable when model is running, while aspect embedding representation as 'question' is updated in each hop. Fig. 2 shows the constitution of input representation of FCMN.

Before training in the multilayer model of FCMN, each word in input sentence is converted into d-dimensional vector, which is known as word embedding. Given a word set S of input sentence,

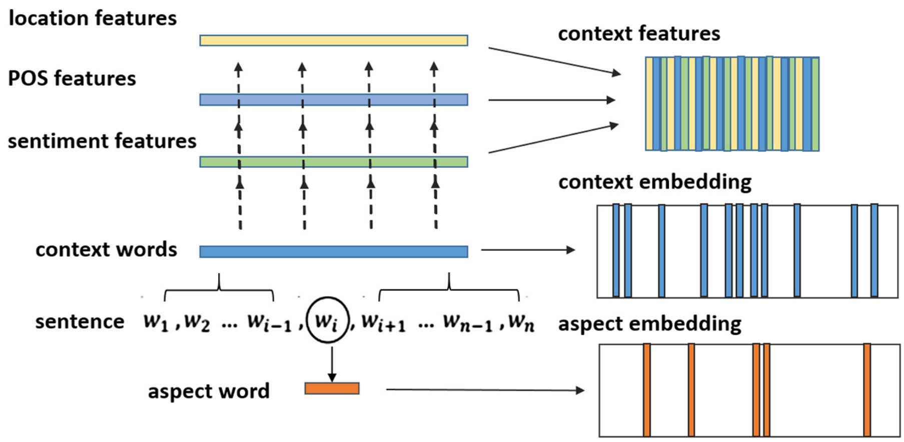


Fig. 2. The constitution of input representation in FCMN.

we first separate S into two sets, context set $C = \{c_1, c_2, \dots, c_{n_c}\}$ and aspect set $A = \{a_1, a_2, \dots, a_{n_a}\}$, where $S = C + A$ and n_c plus n_a equals the sentence length. The context set $\{c_i\}$ is converted into memory vectors $\{m_i\}$ with d-dimensional Glove vectors (of size $d \times V$) [34]. The same embedding operation is used to obtain the aspect vectors $\{q_i\}$ which are averaged to build a single vector $u \in R^{d*1}$ as question input in memory networks.

Besides the embedding representations, we build another representations for context words by extracting context features in three directions.

(1) **Location Feature:** The location feature is used widely in recent attention mechanism research. It is helpful because the closer context words usually have tighter relationship with aspect words. Following Tang et al., we build the location vector $\{l_i\}$ by computing location value for each word with:

$$l_i = 1 - \frac{d_i}{J} \tag{1}$$

where J is the number of context words, d_i is the absolute distance between the single word and aspect term.

(2) **POS Feature:** The part-of-speech is beneficial to the sentiment analysis. To obtain part-of-speech (POS) tags, context words are parsed with the Natural Language Toolkits (NLTK) [35] and we build a POS vector $\{t_i\}$, in which each element t_i is the category Id of single word w_i. The category Id is the index number in candidate categories.

(3) **Sentiment Feature:** The sentiment polarity of single word in usual can help to define it in current sentence. If there are much more positive words than negative words in the sentence, this sentence has a high probability to be positive. We build the sentiment vector $\{s_i\}$ with the support of NRC Hashtag Sentiment Lexicon [36], which is created from 775,310 tweets posted between April and December 2012. In the sentiment vector, s_i is a real number as sentiment score for word w_i. The sentiment polarity is decided by sign symbol of the score whose absolute value is the degree of association with the sentiment. As words not in the lexicon, we set their sentiment score as 0.

Three feature vectors $\{l_i\}$, $\{t_i\}$, $\{s_i\}$ (of size $n_c \times 1$) are calculated in data pre-processing phase and used as complementary inputs in our approach. Then feature vectors are integrated into a feature representation with d dimensions. The $\{l_i\}$, $\{s_i\}$ are continuous real-valued vectors, while the $\{t_i\}$ is a discrete and integer-valued vector. Thus, data transformation for three vectors is necessary. Location vector $\{l_i\}$ and sentiment vector $\{s_i\}$ are converted into d-dimensional vectors using a linear layer with tanh function. POS vector $\{t_i\}$ is embedded with a trainable embedding matrix $W_t \in R^{d \times V_t}$ (V_t is the number of tag categories). The final feature representation vector is produced with:

$$f_i = l_i + s_i + t_i. \tag{2}$$

The feature representations $\{f_i\}$ can provide three kinds of features for context words, while the context embedding $\{m_i\}$ is considered as description of the lexical meaning benefitting by Glove vectors. Composited with two parts of information, the external memory contains multi-angle representations for each word.

3.5. Feature-based compositing memory networks

In this part, we describe Feature-based Compositing Memory Networks model in detail. As basic model shown in Fig. 1, our model has multilayer architecture and external memory. Each layer consists of two parts, attention component and review component. Different from basic model, the external memory in FCMN not only has context embedding, but also contains context features described in Section 3.4. According to compositing strategies, the two parts of representation are utilized in attention component. An illustration of single layer version of FCMN with front-compositing strategy is given in Fig. 3.

As shown in Fig. 3, context features and context embedding are added together directly to obtain the context representations $\{m_i\} \in R^{d \times n_c}$. Then $\{m_i\}$ are used as external memory in attention component. Because compositing operation occurs before attention component, this strategy is called front-compositing strategy.

For each context word with aspect words, the attention component computes an attention score $p \in R^{n_c \times 1}$ is computed by taking a ReLU layer followed by Softmax function:

$$g_i = ReLU(W_{att}[m_i; u] + b_{att}) \tag{3}$$

$$p_i = Softmax(g_i) \tag{4}$$

where $Softmax(z_i) = e^{z_i}/\sum_j e^{z_j}$, $W_{att} \in R^{1 \times 2d}$ and $B_{att} \in R^{1 \times 1}$. In preliminary experiments, we find that the attention score approach has better performance than function in end-to-end memory networks. Attention score determines how much attention that machine pays for each word in external memory units. We calculate the output vector of attention component by a weighted sum using the score vector p:

$$o = \sum p_i \times m_i. \tag{5}$$

Attention component aims to find new facts from memory, while review component is used to process the old facts. This component

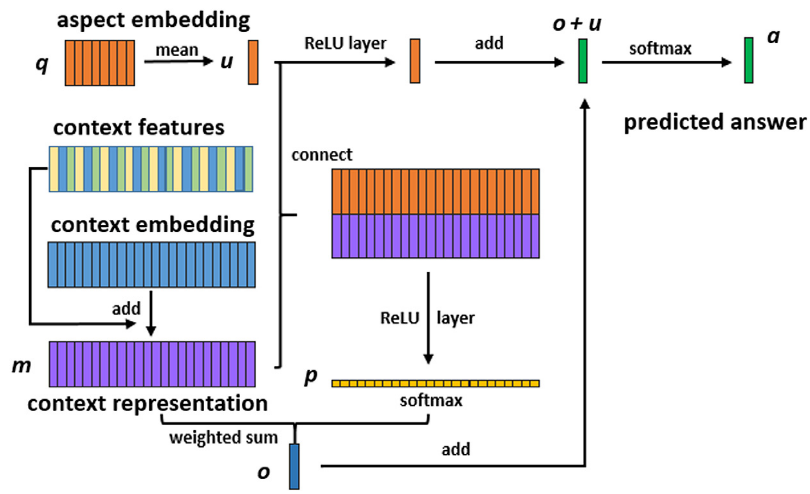


Fig. 3. An illustration of single layer version of FCMN with front-compositing strategy.

is a ReLU layer with aspect representation u as input:

$$\hat{u} = ReLU(W_{rev}u + b_{rev}) \tag{6}$$

where $W_{rev} \in R^{d\times d}$ and $B_{rev} \in R^{d\times 1}$. After obtaining output vectors o and $\hat{u}$ from two components, we finally sum them as the output of single layer which is used as aspect input of next layer:

$$u^{k+1} = \hat{u}^k + o^k. \tag{7}$$

The output of last layer is passed through a final weight matrix W (of size $d \times d$) and a Softmax function to produce the predicted sentiment polarity:

$$\hat{a} = Softmax(W(\hat{u}^n + o^n)). \tag{8}$$

Just like the working memory in psychology, multilayer memory networks aim to build short-term, limited memory according to aspect and context inputs. The context representations used as external memory, can be regarded as knowledge and experience about context words from long-term memory. In each single layer of FCMN, related sentiment information is extracted by attention component from external memory, and combined with review information to build short-term working memory. Human-like facts reasoning makes the FCMN have the ability to solve aspect-based sentiment classification.

3.6. Compositing strategies for FCMN

In previous section, we take the front-compositing strategy as an example to describe the detailed mechanism of FCMN. However, directly summing in that approach is so simple and has no modification with architecture of attention component. To make context features play its role effectively in FCMN, we design three compositing strategies which are described below:

(1) **Front-compositing:** As shown in previous section, this strategy effects before attention component. For given context embedding $\{m_i\}$ and context features $\{f_i\}$, the compositing representations $\hat{m}_i \in R^{d\times n_c}$ is computed with:

$$\hat{m}_i = m_i + f_i. \tag{9}$$

Then the compositing context representations are used as external memory in FCMN.

(2) **Inside-compositing:** In this strategy, context embedding $\{m_i\}$ and context features $\{f_i\}$, are used to compute attention score respectively. Obtaining score vector p_{m_i} for context embedding is the same with Eqs. (3), (4), while score vector p_{f_i} for context features is computing with:

$$\hat{g}_i = ReLU(W_{att2}[f_i; u] + b_{att2}) \tag{10}$$

$$p_{f_i} = softmax(\hat{g}_i) \tag{11}$$

where $W_{att2} \in R^{1\times 2d}$ and $B_{att2} \in R^{1\times 1}$. The final score vector used as weight in Eq. (5) is the sum of two attention scores.

$$p_i = p_{m_i} + p_{f_i}. \tag{12}$$

(3) **Rear-compositing:** Another attention component is inserted into each single layer in rear-compositing strategy. The new component using the context features as external memory, has the same operation with attention component described above. The output vector of new component $\hat{o}_i$ is added with attention output o_i and review output u_i to obtain the final output of single layer.

$$u^{k+1} = \hat{u}^k + o^k + \hat{o}^k. \tag{13}$$

According to the difference of the compositing location, the compositing strategy are divided into three types. An illustration of single layer model with three compositing strategies is shown in Fig. 4. It needs to be emphasized that, the objects we composited are different when compositing location changing. Input representations are composited in front-compositing, corresponding to the attention score in second strategy, and the last composited object is output vector. In next section, we analyze the performance of three compositing strategies in depth on the basis of experiment results.

4. Experiments

In this section, we first describe the experiment settings and training details. Then we compare performance of FCMN with other approaches on Laptops and Restaurants datasets from SemEval 2014. The effectiveness of three compositing strategies is analyzed in depth and finally we discuss the performance of attention mechanism after compositing context features.

4.1. Dataset and data preprocessing

Two datasets from SemEval 2014 [20] have been employed for the experiments, which containing manually annotated reviews

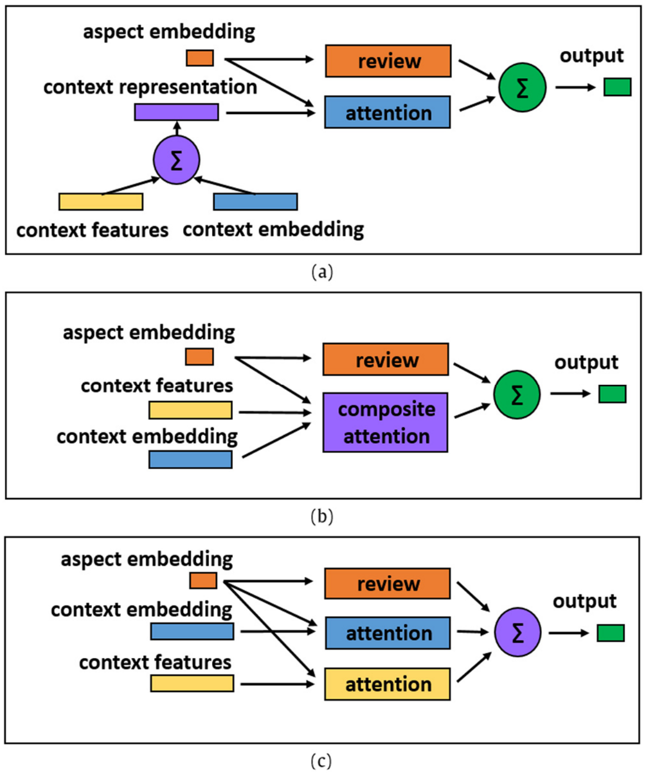


Fig. 4. An illustration of single layer model with three compositing strategies. (a) Front-compositing strategy, (b) Inside-compositing strategy, (c) Rear-compositing strategy. To facilitate understanding, the compositing part in each model is highlighted in purple. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of laptops and restaurants respectively. In two datasets, a given ABSC task consists of a review sentence and an aspect together with a polarity label including four categories: 'positive', 'negative', 'neural' and 'conflict'. We follow the process of Tang et al. removing 'conflict' category because instances in this category are very few and make the dataset extremely unbalanced. The statistics of two datasets are shown in Table 1.

As described in Section 3, we preprocess the datasets and extract word features for each review sentence. After preprocessing, the data inputs of FCMN for each task contains 6 parts: (1) Context words vector, each element is the index of word in vocabulary. (2) Aspect words vector, the same architecture with context words vector. (3) Polarity Index, the index of category which polarity belongs to. (4) Location feature vector, each element is location value for corresponding word. (5) Sentiment feature vector, each element is sentiment value for corresponding word. (6) Part-of-speech feature vector, each element is the index of POS category. Except the aspect words vector and polarity index, the other four vectors have the same size which is equal to the length of context words in sentence. The polarity index is regarded as 'label' to compare with the classification results.

4.2. Training details

We implement our model in Tensorflow [37] and use Python script. Training of FCMN is cast as a supervised classification problem to minimize cross-entropy error of the predicted sentiment polarity sequence. Our models are trained using a learning rate of $\eta = 0.01$. We use back propagation to train all of parameters, and update gradients with AdaGrad optimization algorithm. Besides the gradients with norm greater than 100 are clipped to 100.

Table 1
The statistics of two datasets for three polarity categories.

Dataset	Positive	Negative	Neural
Laptop-Train	994	870	464
Laptop-Test	341	128	169
Restaurant-Train	2164	807	637
Restaurant-Test	728	196	196

Table 2
Classification accuracy of different approaches on two datasets.

Methods	Laptops	Restaurant
Majority	53.45	65.00
Feature + SVM	72.10	80.89
LSTM	66.45	74.28
TDLSTM	68.13	75.63
MemN2N	63.11	72.17
DMN	72.37	80.95
FCMN(1)	67.35	79.17
FCMN(3)	72.68	80.25
FCMN(5)	73.31	81.15
FCMN(8)	**73.94**	81.60
FCMN(9)	72.68	**82.03**

The glove vectors we use are 300-dimensional, trained with 42B tokens and have a 1.9M vocabulary. And we set the number of POS categories as 37 and randomize other parameters with uniform distribution $U(-0.01; 0.01)$.

4.3. Comparing to other approaches

We compare FCMN with other approaches on both datasets. Those approaches include the basic baseline method Majority, Feature-based SVM [13], basic LSTM, TDLSTM [33], End-to-end Memory Networks (MemN2N) [28] and DMN proposed by Tang et al. [17]. Experimental results are shown in Table 2.

Our approach FCMN with inside-compositing is shown as FCMN (k), where k is the number of hops in model. From Table 2, we can find that both of LSTM and MemN2N get better results than basic Majority approach. To some extent, it proves the effectiveness of neural network approach. Although the basic MemN2N is worse than LSTM, DMN and our FCMN get much better performance than LSTM and its variant TDLSTM. Our approach in shallow hops ($k \leq 3$) can already outperform the basic approach and TDLSTM. In deep hops, we obtain higher accuracy than the state-of-art approaches feature-based SVM and DMN. With the supporting of context features, we outperform DMN which does not use syntactic parser or sentiment lexicon. Compared with feature-based SVM, we can get better results with less features used.

4.4. Comparing to DMN in different hops

To verify the effectiveness of our approach, we compare FCMN with DMN in different hops from 1 to 9 graphically showcased in Fig. 5. The detailed results are shown in Table 3.

In both DMN and FCMN, the performance of memory networks is on the rise with the increasing of hops. After 6 hops the performance appears fluctuating, and the best value is obtained when the model has more than 7 hops. As shown in the line chart, our approach has better results than DMN in almost every state, especially in Laptops dataset. Compared with DMN, FCMN has the ability to get better results with less hops. On both two datasets, the accuracy of FCMN using 2 hops exceeds DMN with hops less than 7.

4.5. Effects of three compositing strategies

As mentioned in Section 3.6, we design three compositing strategies to combine context features and context embedding

Table 3
Detailed results of comparison with FCMN and DMN on two datasets.

Hops	DMN-Laptops	FCMN-Laptops	DMN-Restaurants	FCMN-Restaurants
1	67.66	67.35	76.10	79.17
2	71.14	72.21	78.61	80.25
3	71.74	72.68	79.06	80.25
4	72.21	73.63	79.87	80.52
5	71.89	73.31	80.14	81.15
6	72.21	72.53	80.05	80.70
7	72.37	73.16	80.32	81.06
8	72.05	73.94	80.14	81.60
9	72.21	72.68	80.95	82.03

Table 4
Related score for each context word in attention component of 9 hops basic FCMN model when hops from 1 to 9. The example sentence is "Great food but the service was dreadful!" with 'food' and 'service' as aspect.

(a) **Aspect**: Food **Polarity**: Positive **Result**: Wrong

Hops	**great**	but	the	service	was	**dreadful**	!
1	**0.18**	0.14	0.06	0.09	0.07	**0.36**	0.09
2	**0.18**	0.14	0.06	0.09	0.07	**0.36**	0.09
3	**0.18**	0.14	0.06	0.09	0.07	**0.37**	0.09
4	**0.17**	0.13	0.07	0.09	0.07	**0.37**	0.09
5	**0.16**	0.13	0.07	0.09	0.08	**0.38**	0.09
6	**0.15**	0.13	0.07	0.09	0.08	**0.38**	0.09
7	**0.15**	0.12	0.08	0.1	0.08	**0.38**	0.09
8	**0.14**	0.12	0.08	0.1	0.09	**0.37**	0.1
9	**0.14**	0.12	0.09	0.1	0.09	**0.36**	0.1

(b) **Aspect**: Service **Polarity**: Negative **Result**: Right

Hops	**great**	food	but	the	was	**dreadful**	!
1	**0.17**	0.08	0.14	0.07	0.07	**0.37**	0.09
2	**0.16**	0.08	0.13	0.07	0.08	**0.38**	0.09
3	**0.16**	0.08	0.13	0.07	0.08	**0.38**	0.09
4	**0.16**	0.09	0.13	0.07	0.08	**0.38**	0.09
5	**0.15**	0.09	0.13	0.08	0.08	**0.38**	0.1
6	**0.15**	0.09	0.12	0.08	0.09	**0.38**	0.1
7	**0.14**	0.09	0.12	0.09	0.09	**0.37**	0.1
8	**0.14**	0.1	0.12	0.09	0.09	**0.36**	0.1
9	**0.14**	0.1	0.12	0.09	0.1	**0.35**	0.1

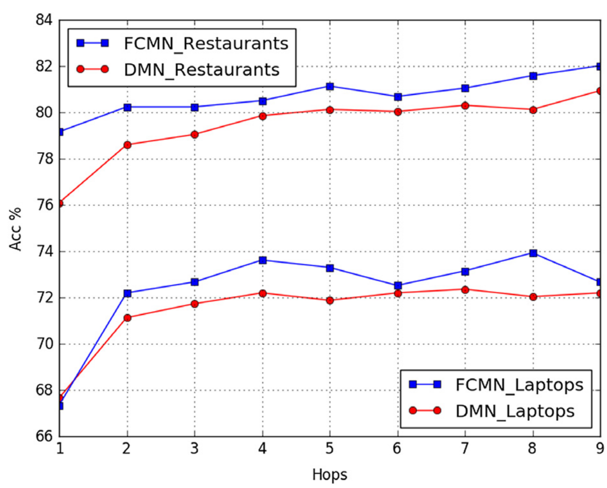


Fig. 5. Classification accuracy of FCMN and DMN in different hops on two datasets.

effectively in FCMN. The intuitive difference of those strategies is compositing location relative to attention component in single layer. We test performance of FCMN with different compositing strategies on two datasets, and experiment results are shown in Fig. 6.

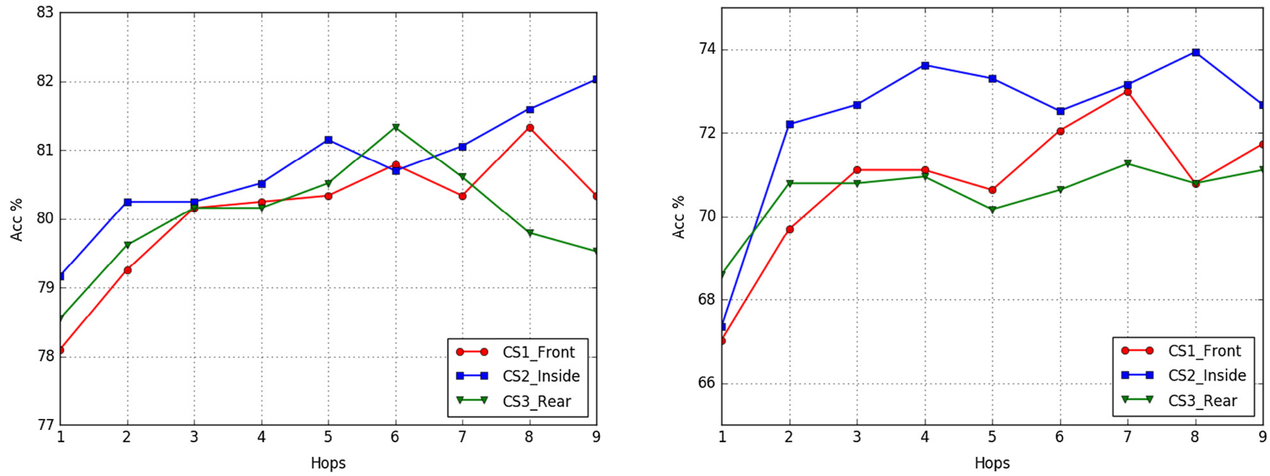
In Fig. 6, each line emerges the changes of classification accuracy using corresponding strategy when the hops number varies from 1 to 9. On the whole, the inside-compositing strategy gets better performance than the other two strategies, while it keeps high accuracy steadily and gets the best accuracy on both two datasets. In Laptops dataset, the inside-compositing strategy keeps more than 72% accuracy when the hops number larger than 1. Meanwhile, it keeps more than 80% accuracy with more than one hop in Restaurants dataset. In comparison, the other two strategies are weaker than the inside-compositing on overall performance. And there are obvious fluctuations in those two strategies with hops increasing. In spite of this, the two strategies still have good performance. The front-compositing strategy gets better accuracy than the best results of DMN on both laptops and restaurants datasets. And the rear-compositing strategy outperforms DMN on restaurants dataset when model contains 8 hops. Compared with compositing the input vectors or output vectors, we find that compositing two score vectors in inside-compositing gets the best performance. The reason probably is that using this way not only retains the characteristics of word embedding and features, but also makes the two parts influence each other in attention mechanism.

4.6. Detailed analysis of attention mechanism in FCMN

We get better performance than DMN based on the same memory networks model. However, why it works when combining the context features into context embedding? To answer this question, we analyze the effects of attention mechanism when running hop by hop. In attention component, how much evidence extracted from each context word is determined by the attention scores in each hop. Because the sum of attention scores for all words is equal to 1, the other scores decrease when one word gets higher score. Thus, we display the attention scores in each hop to understand how machines finding answer in different models. We compare three models including the basic model of FCMN, DMN by Tang et al. and FCMN with inside-compositing strategy. The results are shown from Table 4 to Table 6.

We choose two representative tasks as examples to show experiment results. The two tasks have the same sentence "Great food but the service was dreadful !", but obtain contrary polarity because of different aspect 'food' and 'service'. Humans can understand easily that the sentiment of 'food' is determined by 'great' and 'dreadful' is used to describe 'service'. We denominate the words reflecting sentiment in sentence as 'keyword', such as 'great' and 'dreadful' in this example.

As shown in Table 4, the basic model only using context embedding fails in task 1, while it pays more attention on keyword 'dreadful' than correct keyword 'great' when aspect is 'food'. Despite the basic model passes the task 2, the attention scores in two tasks has no obvious difference. It shows that the basic model is not enough sensitive when aspect changing, one of reasons is that word embedding in Glove just reflects general meaning of single word.



(a) Restaurants.

(b) Laptops.

Fig. 6. Classification accuracy of FCMN with three compositing strategies on two datasets.

Table 5
Related score for each context word in attention component of 9 hops DMN model when hops from 1 to 9. The example sentence is same as Table 4.

(a) **Aspect**: Food **Polarity**: Positive **Result**: Right								(b) **Aspect**: Service **Polarity**: Negative **Result**:Right							
Hops	**great**	but	the	service	was	**dreadful**	!	Hops	**great**	food	but	the	was	**dreadful**	!
1	**0.26**	0.18	0.07	0.09	0.08	**0.21**	0.1	1	**0.17**	0.09	0.15	0.06	0.07	**0.35**	0.1
2	**0.26**	0.18	0.06	0.09	0.08	**0.22**	0.1	2	**0.16**	0.09	0.15	0.07	0.07	**0.36**	0.1
3	**0.26**	0.17	0.08	0.1	0.09	**0.2**	0.11	3	**0.15**	0.09	0.14	0.08	0.08	**0.36**	0.1
4	**0.26**	0.18	0.07	0.09	0.08	**0.21**	0.1	4	**0.15**	0.09	0.14	0.08	0.08	**0.36**	0.1
5	**0.26**	0.17	0.08	0.1	0.09	**0.2**	0.11	5	**0.15**	0.1	0.14	0.08	0.09	**0.35**	0.1
6	**0.26**	0.17	0.08	0.1	0.09	**0.2**	0.11	6	**0.14**	0.1	0.13	0.09	0.09	**0.35**	0.1
7	**0.25**	0.17	0.08	0.1	0.09	**0.2**	0.11	7	**0.14**	0.1	0.13	0.09	0.09	**0.33**	0.11
8	**0.25**	0.17	0.08	0.1	0.09	**0.2**	0.11	8	**0.14**	0.11	0.13	0.1	0.1	**0.32**	0.11
9	**0.25**	0.16	0.09	0.1	0.1	**0.19**	0.11	9	**0.14**	0.11	0.13	0.1	0.1	**0.3**	0.11

Table 6
Related score for each context word in attention component of 9 hops FCMN model with inside-compositing strategy when hops from 1 to 9. The example sentence is same as Table 4.

(a) **Aspect**: Food **Polarity**: Positive **Result**: Right								(b) **Aspect**: Service **Polarity**: Negative **Result**: Right							
Hops	**great**	but	the	service	was	**dreadful**	!	Hops	**great**	food	but	the	was	**dreadful**	!
1	**0.31**	0.14	0.05	0.09	0.05	**0.28**	0.09	1	**0.09**	0.04	0.08	0.05	0.08	**0.61**	0.05
2	**0.37**	0.14	0.05	0.08	0.04	**0.27**	0.05	2	**0.09**	0.04	0.07	0.05	0.09	**0.61**	0.05
3	**0.37**	0.14	0.04	0.08	0.04	**0.26**	0.07	3	**0.08**	0.04	0.06	0.06	0.1	**0.62**	0.04
4	**0.39**	0.15	0.04	0.08	0.04	**0.25**	0.06	4	**0.08**	0.04	0.07	0.06	0.1	**0.62**	0.04
5	**0.39**	0.15	0.04	0.08	0.04	**0.24**	0.06	5	**0.08**	0.04	0.06	0.06	0.1	**0.62**	0.04
6	**0.4**	0.15	0.04	0.08	0.03	**0.23**	0.06	6	**0.08**	0.04	0.06	0.06	0.1	**0.61**	0.04
7	**0.4**	0.15	0.04	0.08	0.03	**0.23**	0.06	7	**0.07**	0.04	0.06	0.06	0.1	**0.61**	0.04
8	**0.41**	0.16	0.04	0.08	0.03	**0.22**	0.06	8	**0.07**	0.04	0.06	0.06	0.1	**0.61**	0.05
9	**0.41**	0.16	0.04	0.09	0.03	**0.21**	0.06	9	**0.07**	0.04	0.06	0.07	0.11	**0.6**	0.05

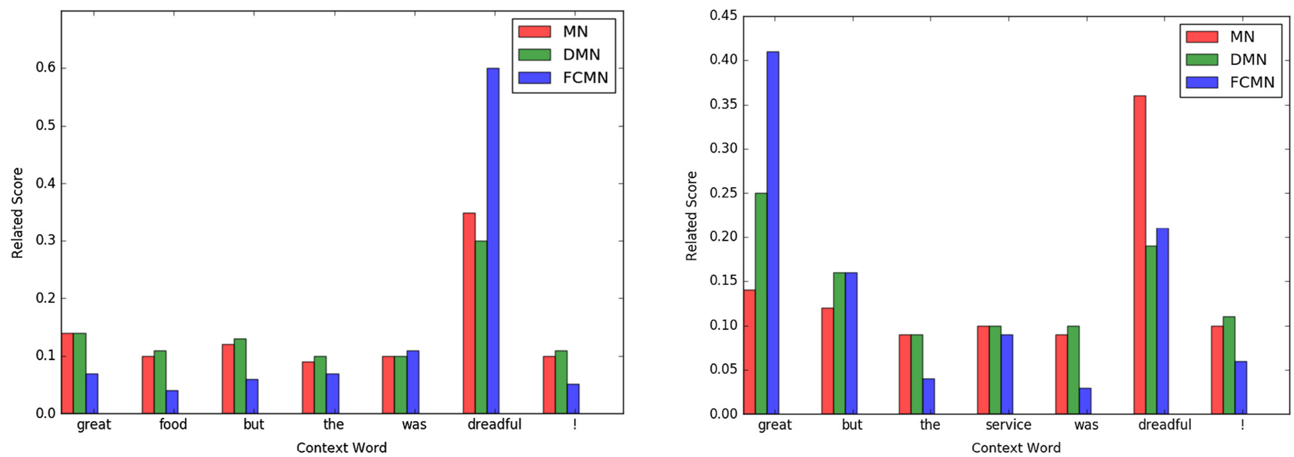
To solve this problem, DMN integrates location information into attention component and obtains good effect in Table 5. The attention scores in DMN are the same as those in basic model for task 2. Nevertheless, DMN passes the first task successfully because it gets a little higher score for correct keyword ‘great’ than ‘dreadful’. This illustrates the benefit of location information in DMN.

Compared with DMN, our approach enriches word representations by three kinds of features and gets improvement in attention component as shown in Table 6.

On the one hand, our approach ignores words without containing sentiment. Because of the part-of-speech features and sentiment features in context representations, the FCMN model can learn to ignore words. Those ignored words have no obvious sentiment polarity or belong to a category in which words usually not used to express feelings. The attention scores for each context word in three different models are shown in Fig. 7. From the bar charts, some words in FCMN, such as ‘but’, ‘was’ and ‘!’, have lower score than those in basic model and DMN.

On the other hand, our approach makes correct keyword more distinct. After ignoring words without sentiment, the model needs to determine which is more important in the remaining keywords. FCMN puts up a good performance in this part. For task 1 with aspect ‘food’, correct keyword ‘great’ obtains higher score with the hops number increasing. The ‘great’ score is almost two times more than ‘dreadful’ in the final hop. Furthermore, keyword ‘dreadful’ in task 2 obtains more than 0.6 score in our approach. With the hops number increasing, the score of ‘dreadful’ remains stable rather than decreasing in two models above.

According to experiment results above, FCMN can capture keyword more effectively than DMN and improve the performance of attention mechanism with the support of context features.



(a) Task with aspect 'service'. (b) Task with aspect 'food'.

Fig. 7. The attention scores of final hop in three different models. The example sentence is "Great food but the service was dreadful!" with 'food' and 'service' as aspect.

5. Conclusion

Identifying sentiment polarity from users' opinion can improve user-friendliness in Social Internet of Things. In this paper, we propose feature-based compositing memory networks (FCMN) to solve aspect-based sentiment classification. The FCMN extracts information not only from word embedding, but also with the support of three kinds of word features including location features, part-of-speech features and sentiment features. Three compositing strategies are designed to combine those information to build context representations, and we find the FCMN with inside-compositing strategy has the best performance. Our approach outperforms the existing state-of-the-art approaches feature-based SVM and deep memory networks in the laptops and restaurants datasets of SemEval 2014. After the analysis of attention mechanism in FCMN, enriching context representations can make the keywords in sentence more distinct, and help FCMN to select effective information from context words.

We will further focus on improving FCMN model and applying it to other NLP problems. On the one hand, there may be more effective way to integrate three features instead of a simple sum. Moreover, besides location, POS, sentiment features, compositing other features into FCMN is worth to study. On the other hand, enriching word representations can provide more information to memory networks, it will be useful in other NLP problems. Thus, our next step will try to apply FCMN to solve those problems, such as question answering, natural language inference, and machine translation.

Acknowledgments

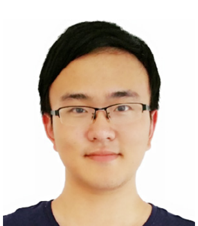
This work is supported by National Natural Science Foundation of China (Grant No. 61672131) and the Fundamental Research Funds for the Central Universities (DUT17ZD216 and DUT16QY27).

References

[1] T. Qiu, R. Qiao, M. Han, A.K. Sangaiah, I. Lee, A lifetime-enhanced data collecting scheme for internet of things, IEEE Commun. Mag. 55 (11) (2017) 132–137.
[2] S. Cheng, Z. Cai, J. Li, H. Gao, Extracting kernel dataset from big sensory data in wireless sensor networks, IEEE Trans. Knowl. Data Eng. 29 (4) (2017) 813–827.
[3] T. Qiu, A. Zhao, F. Xia, W. Si, D. Wu, ROSE: Robustness strategy for scale-free wireless sensor networks, IEEE/ACM Trans. Netw. 25 (5) (2017) 2944–2959.
[4] T. Qiu, R. Qiao, D. Wu, EABS: An event-aware backpressure scheduling scheme for emergency internet of things, IEEE Trans. Mob. Comput. (2017). http://dx.doi.org/10.1109/TMC.2017.2702670.
[5] X. Zheng, Z. Cai, J. Li, H. Gao, A study on application-aware scheduling in wireless networks, IEEE Trans. Mob. Comput. 16 (7) (2017) 1787–1801.
[6] L. Atzori, A. Iera, G. Morabito, M. Nitti, The social internet of things (SIoT) when social networks meet the internet of things: Concept, architecture and network characterization, Comput. Netw. 56 (16) (2012) 3594–3608.
[7] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, Y. Li, Cost-efficient strategies for restraining rumor spreading in mobile social networks, IEEE Trans. Veh. Technol. 66 (3) (2017) 2789–2800.
[8] D. Hussein, S.N. Han, G.M. Lee, N. Crespi, E. Bertin, Towards a dynamic discovery of smart services in the social internet of things, Comput. Electr. Eng. 58 (2017) 429–443.
[9] M. Strohbach, H. Ziekow, V. Gazis, N. Akiva, Towards a big data analytics framework for IoT and smart city applications, in: Modeling and Processing for Next-Generation Big-Data Technologies: With Applications and Case Studies, Springer, 2015, pp. 257–282.
[10] L. Zhang, B. Liu, Sentiment analysis and opinion mining, in: Encyclopedia of Machine Learning and Data Mining, Springer, 2016, pp. 1–10.
[11] M. Hu, B. Liu, Mining opinion features in customer reviews, in: Proceedings of the 19th National Conference on Artifical Intelligence, 2004, pp. 755–760.
[12] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, L. Tounsi, DCU: Aspect-based polarity classification for SemEval task 4, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval 2014, 2014, pp. 223–229.
[13] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, NRC-Canada-2014: Detecting aspects and sentiment in customer reviews, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval 2014, 2014, pp. 437–442.
[14] H. Lakkaraju, R. Socher, C. Manning, Aspect specific sentiment analysis using hierarchical deep learning, in: Proceedings of the NIPS Workshop on Deep Learning and Representation Learning, 2014, pp. 1–9.
[15] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, J. Weston, Evaluating prerequisite qualities for learning end-to-end dialog systems, in: Proceedings of the International Conference on Learning Representations, ICLR, 2016.
[16] A. Bordes, N. Usunier, S. Chopra, J. Weston, Large-scale simple question answering with memory networks, CoRR abs/1506.02075, 2015, pp. 1–10. arXiv:1506.02075.
[17] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016, pp. 214–224.
[18] F. Hill, A. Bordes, S. Chopra, J. Weston, The goldilocks principle: Reading children's books with explicit memory representations, in: Proceedings of the International Conference on Learning Representations, ICLR, 2016.
[19] J. Weston, A. Bordes, S. Chopra, A.M. Rush, B. van Merrinboer, A. Joulin, T. Mikolov, Towards AI-complete question answering: A set of prerequisite toy tasks, in: Proceedings of the International Conference on Learning Representations, ICLR, 2016.
[20] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, SemEval-2014 task 4: Aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval 2014, 2014, pp. 27–35.
[21] J. Duchi, H. Elad, S. Yoram, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (Jul) (2011) 2121–2159.
[22] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of the International Conference on Learning Representations, ICLR, 2015.

[23] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, J. Bradbury, R.S. Com, Ask me anything : Dynamic memory networks for natural language processing, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 1378–1387.
[24] K.M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: Advances in Neural Information Processing Systems, 2015, pp. 1693–1701.
[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[26] A. Graves, G. Wayne, I. Danihelka, Neural Turing Machines, CoRR abs/1410.5401, 2014, pp. 1–26. arXiv:1410.5401.
[27] J. Weston, S. Chopra, A. Bordes, Memory networks, in: Proceedings of the International Conference on Learning Representations, ICLR, 2015.
[28] S. Sukhbaatar, J. Weston, R. Fergus, et al., End-to-end memory networks, in: Advances in Neural Information Processing Systems, 2015, pp. 2440–2448.
[29] C. Xiong, S. Merity, R. Socher, Dynamic memory networks for visual and textual question answering, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 2397–2406.
[30] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1724–1734.
[31] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the Meeting of the Association for Computational Linguistics, 2014, pp. 49–54.
[32] T.H. Nguyen, K. Shirai, Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2509–2514.
[33] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, in: Proceedings of the International Conference on Computational Linguistics, 2016, pp. 3298–3307.
[34] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Vol. 14, 2014, pp. 1532–1543.
[35] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009.
[36] S.M. Mohammad, S. Kiritchenko, X. Zhu, NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets, in: Joint Conference on Lexical and Computational Semantics, 2013, pp. 321–327.
[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I.J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D.G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P.A. Tucker, V. Vanhoucke, V. Vasudevan, F.B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. Software available from tensorflow.org.

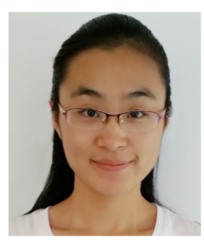
Ruixin Ma received M.Sc. and Ph.D. from Dalian University of Technology (DUT), in 2003 and 2012, respectively. He is currently Associate Professor at School of Software, Dalian University of Technology, China. He has authored/coauthored 7 books, over 10 scientific papers international journals and conference proceedings. He has contributed to the development of 10 copyrighted software systems.



Kai Wang received the B.E. from Dalian University of Technology, China, in 2015. He is currently Master Student in School of Software, Dalian University of Technology (DUT), China. He used to participate in "Microsoft Imagine Cup Global Student Contest 2015" and won the National Third Prize. He is an excellent graduate student of DUT and has been awarded several scholarships in academic excellence and technology innovation. His research interests cover neural language processing, deep learning and knowledge base construction.

Tie Qiu received B.Sc. from Inner Mongolia University of Technology, M.Sc. and Ph.D. from Dalian University of Technology (DUT), China, in 2003, 2005 and 2012, respectively. He is currently Associate Professor at School of Software, Dalian University of Technology. He was a visiting professor at electrical and computer engineering at Iowa State University in U.S. (Jan. 2014–Jan. 2015). He serves as an Associate Editor of IEEE Access Journal, Computers & Electrical Engineering (Elsevier journal) and Human-centric Computing and Information Sciences (Springer Journal), an Editorial Board Member of Ad Hoc Networks (Elsevier journal) and International Journal on AdHoc Networking Systems, a Gest Editor of Future Generation Computer Systems (Elsevier journal). He serves as General Chair, PC Chair, Workshop Chair, Publicity Chair, Publication Chair or TPC Member of a number of conferences. He has authored/co-authored 8 books, over 60 scientific papers in international journals and conference proceedings. He has contributed to the development of 4 copyrighted software systems and invented 15 patents. He is a senior member of China Computer Federation (CCF) and a Senior Member of IEEE and ACM.

Arun Kumar Sangaiah received his Ph.D. from VIT University and Master of Engineering from Anna University, in 2007 and 2014, respectively. He is currently Associate Professor at School of Computing Science and Engineering, VIT University, Vellore, India. He was a visiting professor at School of computer engineering at Nanhai Dongruan Information Technology Institute in China (September. 2016–Jan. 2017). He has published more than 100 scientific papers in high standard SCI journals like IEEE-TII, IEEE-Communication Magazine, IEEE systems, IEEE-IoT, IEEE TSC, IEEE ETC and etc. In addition he has authored/edited over 8 books (Elsevier, Springer, Wiley, Taylor and Francis) and 50 journal special issues such as IEEE-Communication Magazine, IEEE-IoT, IEEE consumer electronic magazine etc. His area of interest includes software engineering, computational intelligence, wireless networks, bio-informatics, and embedded systems. Also, he was registered a one Indian patent in the area of Computational Intelligence. Besides, Prof. Sangaiah is responsible for Editorial Board Member/Associate Editor of various international SCI journals.



Dan Lin received the B.E. from Shandong Normal University, China, in 2015. She is currently Master Student in School of Software, Dalian University of Technology (DUT), China. She used to participate in "the 13th Huawei Cup National Post-Graduate Mathematical Contest in Modeling" and won the National Second Prize. She is an excellent graduate student of DUT and has been awarded several scholarships in academic excellence and technology innovation. Her research interests cover data mining, deep learning and internet of things.

Hannan Bin Liaqat received the B.S. degree in information technology and the M.S. degree in computer networks from COMSATS Institute of Information Technology, Lahore, Pakistan, in 2006 and 2009, respectively. From 2009 to 2011, he was a Lecturer in the Computer Science Department with the University of Gujrat, Gujrat, Pakistan. He received the Ph.D., degree from Dalian University of Technology, Dalian, China, in 2016. From 2016 to now, he worked for Department of Information Technology of University of Gujrat as an Assistant Professor, Ph.D., and M.Phil. supervisor. Dr. Hannan has a number of publications to his credits in international journals and conferences. He is the reviewer of several international journals. His research interests include ad hoc social networks, IoT, cloud computing, mobile computing, and social computing.