Knowledge-Based Systems 148 (2018) 167–176

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Learning multi-grained aspect target sequence for Chinese sentiment analysis

Haiyun Peng[a], Yukun Ma[a], Yang Li[b], Erik Cambria[a,*]

[a] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b] School of Automation, Northwestern Polytechnical University, China

ARTICLE INFO

Article history:
Received 14 November 2017
Revised 21 January 2018
Accepted 22 February 2018

Keywords:
Aspect-based sentiment analysis
Chinese NLP
Multi-grained Chinese text representation

ABSTRACT

Aspect-based sentiment analysis aims at identifying sentiment polarity towards aspect targets in a sentence. Previously, the task was modeled as a sentence-level sentiment classification problem that treated aspect targets as a hint. Such approaches oversimplify the problem by averaging word embeddings when the aspect target is a multi-word sequence. In this paper, we formalize the problem from a different perspective, i.e., that sentiment at aspect target level should be the main focus. Due to the fact that written Chinese is very rich and complex, Chinese aspect targets can be studied at three different levels of granularity: radical, character and word. Thus, we propose to explicitly model the aspect target and conduct sentiment classification directly at the aspect target level via three granularities. Moreover, we study two fusion methods for such granularities in the task of Chinese aspect-level sentiment analysis. Experimental results on a multi-word aspect target subset from SemEval2014 and four Chinese review datasets validate our claims and show the improved performance of our model over the state of the art.

1. Introduction

In recent years, sentiment analysis has become increasingly popular for processing social media data on online communities, blogs, wikis, microblogging platforms, and other online collaborative media [1]. Sentiment analysis is a branch of affective computing research [2] that aims to classify text – but sometimes also audio and video [3] – into either positive or negative – but sometimes also neutral [4]. Most of the literature is on English language but recently an increasing number of publications is tackling the multilinguality issue [5].

Sentiment analysis techniques can be broadly categorized into symbolic and sub-symbolic approaches: the former include the use of lexicons [6], ontologies [7], and semantic networks [8] to encode the polarity associated with words and multi-word expressions; the latter consist of supervised [9], semi-supervised [10] and unsupervised [11] machine learning techniques that perform sentiment classification based on word co-occurrence frequencies. There are also a few hybrid approaches [12] that leverage an ensemble of symbolic and sub-symbolic techniques for polarity detection.

Sentiment analysis has raised growing interest both within the scientific community, leading to many exciting open challenges, as well as in the business world, due to the remarkable benefits to be had from financial [13] and political [14] forecasting, e-health [15] and e-tourism [16], user profiling [17] and community detection [18], manufacturing and supply chain applications [19], human communication comprehension [20] and dialogue systems [21], etc.

While most works approach it as a simple categorization problem, sentiment analysis is actually a suitcase research problem [22] that requires tackling many NLP tasks, including named-entity recognition [23], word polarity disambiguation [24], concept extraction [25], subjectivity detection [26], personality recognition [27], sarcasm detection [28], and especially aspect extraction [29]. Aspect-based sentiment analysis (ABSA) proposes a finer-grained polarity detection that extracts aspects first and then classifies them as either positive or negative. For example, in the sentence "*The size of the room was smaller than our expectation but the view from the room would not make you disappointed.*", sentiments expressed towards "*room size*" and "*room view*" are negative and positive, respectively. Those two terms are called aspect terms and ABSA associates a polarity to each aspect term. Another similar yet different sub-task of ABSA is sentiment analysis towards aspect category. For example, both "*room size*" and "*room view*" in the previous example belong to "ROOM_FACILITY". Other aspect categories in this domain are like "PRICE", "SERVICE" and so on. The work of Wang et al. [30] belongs to this sub-task.

In this paper, we focus on aspect term sentiment classification, which is a finer grained study compared to the work of Wang et al.

* Corresponding author.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.knosys.2018.02.034
0950-7051/

Table 1
Comparison between English and Chinese text in composition.

English			Chinese		
Hierarchy	Example	Encodes semantics	Hierarchy	Example	Encodes semantics
Character	a, b, c, ...	N	Radical	丬,忄,宀	Y
Character N-gram	pre, sub	partial Y	Character	雪,林,伐	Y
Word	awake, cheer	Y	Single-character word	好,灯	Y
Phrase	kick off, put on	Y	Multi-character word	风景,大自然	Y
Sentence	Nice to meet you.	Y	Sentence	我很高兴遇见你。	Y

We define aspect term as aspect target. If an aspect term contains multiple words, we call this type of aspect term as aspect target sequence. In aspect target sentiment classification, Tang et al. [31] used target dependent a long short-term memory (LSTM) network. In particular, they use a Bi-LSTM model to encode the sequential information in TC-LSTM. They later appended each word with target embedding to reinforce the extraction of correlation between target and context words in the sentence. In [32], they designed a pure attention-based memory network to explicitly learn the correlation between context words and aspect target. Nevertheless, they simply used average aspect word embedding to represent aspect term, which failed to consider the aspect target sequence information. Wang et al. [30] employed an attention mechanism upon the sequential output from a LSTM layer. Their work treated the sentence sequential information as equal importance to aspect target sequential information.

All the previous work modeled ABSA as a sentence-level sentiment classification problem that treated aspect target/term as a hint. Such design will result in a dilemma when there appear two aspect targets with opposite sentiment polarities in the same sentence. All state-of-the-art works only focused on one aspect target at a time. They cannot process two aspect targets at the same time, due to the assumption that sentiment of a sentence is equivalent to the sentiment of aspect target/term. Moreover, little attention is paid to aspect target itself, especially when aspect target is a sequence of words, namely multi-word aspect. Almost all literature took the average word embeddings to represent aspect target sequence, which ignored aspect target sequential information. In the English language, their models work well in situations where aspect target has a single word, but not in multiple words. In other cases, although they employed a sentence-level sequence encoder, the importance of aspect target sequence is treated with no emphasis compared with non-aspect word sequence. To this end, we propose two versions of an aspect target sequence model (ATSM), namely: ATSM-S, where -S stands for single granularity, and ATSM-F, where -F stands for fusion. The model is available for download at http://github.com/senticnet/aspect-target-sequence-model.

ATSM-S explicitly addresses to the multi-word aspect target case. The model includes two crucial modules: adaptive embedding learning and aspect target sequence learning. The first module aims at appending sentence context meaning to general word embeddings for each of the aspect target words. Thus, an accurate vector representation which encoded sentence context will be obtained for aspect target words. Specifically, we extract sentence context with a LSTM encoder. Each aspect target word was attended by the encoded context to form an adaptive word embedding. The second module links each adaptive aspect word embedding with a sequence learning. In the experimental comparison, our ATSM-S outperforms the state of the art on an English multi-word aspect subset filtered from SemEval 2014 and four Chinese review datasets.

Although ATSM-S only solves part of the problems (multi-word aspect scenario) in English ABSA, it becomes comprehensive in addressing Chinese ABSA if considering the multi-granularity representation of Chinese text.

Chinese is a pictogram language whose text originates from images. Chinese text originates from simple symbols. The symbols gradually evolved to fixed types (named radicals). Through a geometric composition, those fixed types build up characters. Then a concatenation of characters creates the word. Unlike English, each Chinese sub-word granular representation still encodes semantics, shown in Table 1. Whereas in English, only partial character N-grams encode semantics. This motivates us to explore each granularity of Chinese text in ABSA. In addition, the surface form of Chinese text is at the character level. This guaranteed that even the smallest aspect target, such as a single Chinese character, can be broken down into a sequence of aspect targets at the radical level. Thus, we proposed ATSM-F as an upgraded version of ATSM-S. Specifically, ATSM-S was conducted at each Chinese granularity and ATSM-F fused their results together. In the design of fusion, we tested both early fusion (hierarchical structure) and late fusion (flat structure). Finally, ATSM-F with late fusion prevails all other methods on three out of four Chinese review datasets. To round up, we made the following contributions:

1. We view aspect-level sentiment analysis from a new perspective, in which aspect target sequence dominates the final result. Whereas in recent literature using deep learning, sentence-level classification is the popular solution [30–32].
2. Starting from our perspective, we propose the adaptive embedding learning to append sentence context to aspect targets. Followed by an explicit modeling of aspect target sequence. Results on English multi-word aspect subset from SemEval2014 and four Chinese review datasets validate the superiority of our model.
3. We take advantage of the multi-grained representation nature of Chinese text and improve the final performance further, which suggests a broader application scenario.

2. Related work

2.1. Aspect-based sentiment analysis

In ABSA, there are three research directions. The first direction is aspect term extraction, such as [33,34]. The second direction aims at categorizing the given aspect term to different categories [35,36]. Wang et al. [30] employed an attention mechanism upon the sequential output from a LSTM layer. Their work aims at predicting sentiment polarity of the category, such as “FOOD” and “PRICE”, rather than any particular aspect terms.

The third branch works on aspect term sentiment classification. Aspect term is usually marked in a sentence and the goal is to determine the sentiment polarity towards the aspect term. Early works used dictionary-based methods [37,38]. Recent works employed machine learning-based feature engineering and classification [39,40].

Most state-of-the-art works used a LSTM network [41] and attention mechanism as the basic modules in their methods [31,32]. Tang et al. used target dependent Bi-LSTM model to encode the sequential information in TC-LSTM. They later appended each word with target embedding to reinforce the extraction of correlation between target and context words in the sentence. In MemNet [32], they designed a pure attention-based memory network to explicitly learn the correlation between context words and aspect words.

Previous works on aspect-term sentiment analysis suffered from two main drawbacks. Firstly, ABSA is modeled as a sentence-level sentiment classification problem that treated aspect target/term as a hint. Such design will result in a dilemma when there appear two aspect targets with opposite sentiment polarities. All state-of-the-art works only focused on one aspect target at a time. They cannot process two aspect targets at the same time, due to the assumption that sentiment of a sentence is equivalent to the sentiment of aspect target/term. Secondly, little attention is paid to aspect target itself, namely aspect target sequence information. In this paper, we aim to address these two drawbacks.

2.2. Chinese text representation

Contemporary Chinese text processing mostly relies on Chinese word embeddings [42,43]. However, Chinese word consists sub-elements like characters that contain semantics. Xu et al. [44] studied characters within a word can enrich semantics for Chinese word and character embeddings. Chinese text has one level below character level, which is radical level. It has been demonstrated that radical level representation also encodes certain extent of semantics [45]. Chen et al. [46] started to decompose Chinese words into characters and proposed a character-enhanced word embedding model (CWE). Sun et al. [47] started to decompose Chinese characters to radicals and developed a radical-enhanced Chinese character embedding. Shi et al. [45] began to train pure radical-based embedding for short-text categorization, Chinese word segmentation, and web search ranking. Li et al. [48] proposed component-enhanced Chinese character embedding models by incorporating internal compositions and the external contexts of Chinese characters. Yin et al. [49] extended the pure radical embedding in [50] by introducing multi-granularity Chinese word embeddings. Peng et al. [51] developed radical-based hierarchical Chinese embeddings specifically for sentiment analysis. However, none of them has exploited the multi-grained characteristic of Chinese text in complex NLP problems, such as ABSA.

3. Method overview

In this section, we first define our task and then present an overview of the proposed method.

3.1. Aspect target sequence

Aspect is a concept that contains various interpretations, such as aspect target/term, aspect word, aspect category, aspect sentiment etc. For instance, a sentence “这菜味道不错。 (This cuisine has a good flavor.)” has an aspect target/term “味道 (flavor)”. The aspect target contains only one aspect word, which is “味道 (flavor)”. The aspect target belongs to an aspect category of “FOOD”. Other aspect categories in the domain of restaurant are like “PRICE”, “SERVICE” and so on. The sentiment of the aspect target “口味 (flavor)”is positive in the sentence.

However, in the context of this paper, we define aspect as aspect target sequence. As Chinese text can be decomposed to three granularities, a single unit of higher level representation can be decomposed to a sequence of units of lower level representation. For instance, the single-word aspect target “味道 (flavor)” in the previous example can be decomposed to a sequence of Chinese characters: “味” and “道”. Moreover, the characters can be further decomposed to a sequence of Chinese radicals: “口”, “未”,“辶” and “首”. As [44,49,51] suggested, various granularities contain exclusive semantics. In the above example, “味道 ” at word level simply means ‘flavor’. “味” and “道” at character level mean ‘thinking of the flavor’. “口”, “未”,“辶” and “首” at radical level mean ‘to taste the unknown and brainstorm the flavor’. It is apparent to see from the example that sub-component semantics provide complementary explanations to the word and, hence, enrich its meaning. We reconstruct an aspect target as three sequences at three granularities. Methods were developed to work on these sequences in order to determine the sentiment polarity of the aspect target.

3.2. Task definition

A sentence s of n units (the unit could be radical, character or word) in the format $s = \{u_1, u_2, \ldots, u_j, u_{j+1}, \ldots, u_{j+L}, \ldots, u_{n-1}, u_n\}$ is marked out with aspect target (comprising multiple units)$\{u_j, u_{j+1}, \ldots, u_{j+L}\}$. The u_{j+L} stands for the $(j+L)$th unit in the sentence and the Lth unit in the aspect target. L indicates that the aspect target contains L consecutive units. The goal is to predict the sentiment polarity of the aspect target.

3.3. Overview of the algorithm

In all previous works of ABSA [30–32], if an aspect target contains multiple words, they treated the multi-word aspect as one unified target by averaging their word embeddings. This is disadvantageous in two ways. Firstly, word embeddings of aspect target that were trained from general corpus might mislead the meaning of aspect target in the sentence. Secondly, sequential information within the aspect target is lost. For instance, a sentence “*The **red apple**released in California last year was a disappointment.*” contains an aspect target “***red apple***”. Based on the occurrence of “released in California”, we can understand that “***red apple***” stands for iPhone. If general word embeddings of “***red***” and “***apple***” were used in the task, it will deviate from the symbolic meaning of iPhone to fruit apple. To make it worse, by averaging the word embeddings of “***red***” and “***apple***”, sequential information is lost and the averaged word embedding will result in a new/irrelevant meaning in the word vector space.

In order to address the above two issues, we propose a three-step model. The first step is adaptive embedding learning, which essentially aims at learning intra-sentence context for each unit in the aspect target sequence. It was designed to embed intra-sentence contexts to the general embeddings of units in aspect target sequence, which will resolve the first issue aforementioned. The second step is simply a sequence learning process of aspect target, which has never been addressed before. Last but not the least, Chinese text has three granularities of representation (radical, character and word) so that we apply the first two steps at each of the three granularities and glue them together with fusion mechanisms. This is particular for Chinese text, as even single word aspect target can be decomposed to up to three sequences of representation. Fig. 1 presents a graphical illustration of ATSM-F in late fusion. In English, however, our model only applies to cases where aspect target contains multiple words. We will illustrate each of the three steps below.

4. Adaptive embedding learning

4.1. Sentence sequence learning

Sequential information is crucial in determining aspect term sentiment polarity. For example, there are two sentences: “*The movie supposed to be amazing but I find it just so-so.*” and “*The movie supposed to be just so so but I find it amazing.*” These two sentences have exactly same words but arranged in a different

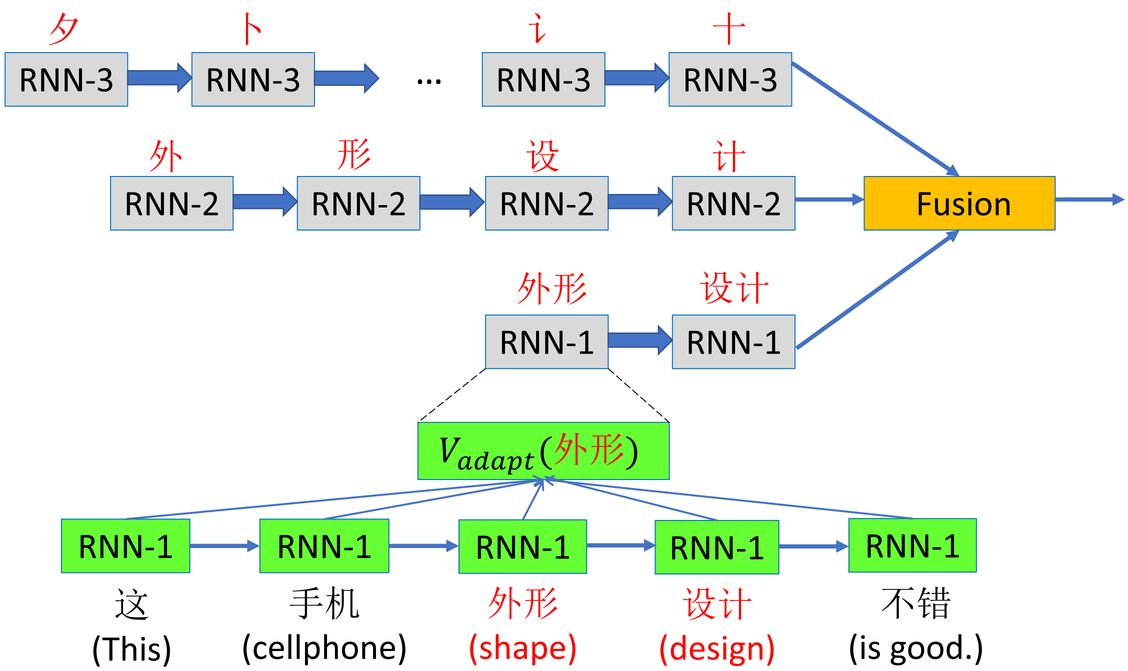


Fig. 1. ATSM-F late fusion framework. RNN-1,-2,-3 are at word, character and radical level, respectively. Green RNNs are for adaptive embedding learning. Grey RNNs are sequence learning of aspect target. Aspect target is highlighted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

order. This results in the opposite sentiment polarity of the aspect target "*The movie*". In order to extract sentence sequential information like the above, we use a LSTM to encode the sentence sequential information. The output of the LSTM is a sequence of cell hidden outputs which has the same length of the sentence. Mathematically, a sentence and its corresponding LSTM sequence output are denoted as $\{w_1, w_2, \ldots, w_j, \ldots, w_{n-1}, w_n\}$ and $\{h_1, h_2, \ldots, h_j, \ldots, h_{n-1}, h_n\}$, respectively, where $w_n \in \mathbb{R}^{1\times d}$ and $h_n \in \mathbb{R}^{1\times d_{lstm}}$.

4.2. Aspect target unit learning

As we discussed before, the meaning of aspect target word may be shifted due to the sentence context, such as the example of "*red apple*". Thus, we decide to embed the intra-sentence contexts to each unit in the aspect target. To this end, we employ an attention mechanism to realize the learning. As we know from Bahdanau et al. [52], the attention mechanism can be understood as a weighted memory of lower-level elements. Conceptually, the output attention vector extracts the correlation between query (in our case which is the unit in aspect target) with each element. In our model, we compute an attention for each unit in aspect target with LSTM sequence hidden output from sentence sequence learning and name it adaptive vector. Thus, the adaptive vector extracts the most relevant correlation with intra-sentence context.

Specifically, for an aspect target unit u_i and its word embedding $v_i \in \mathbb{R}^{1\times d}$ in a sentence of length n, its adaptive vector $V_{adapt} \in \mathbb{R}^{1\times(d+d_{lstm})}$ is given as below:

$$V_{adapt} = \sum_{j=1}^{n} \alpha_j \cdot \begin{bmatrix} v_i \\ h_j \end{bmatrix} \quad (1)$$

where $h_j \in \mathbb{R}^{1\times d_{lstm}}$ is the jth output from LSTM hidden output sequence. α_j is the weight for the jth memory in the sentence and $\sum_{j=1}^{n} \alpha_j = 1$. It depicts how much the semantic influence of the jth unit imposed on the aspect target unit u_i. It is a weight computed from *softmax* below:

$$\alpha_j = \frac{e^{g_j}}{\sum_{m=1}^{n} e^{g_m}} \quad (2)$$

where g_j is a score obtained from a feed-forward neural network attention model and its formula is given as:

$$g_j = tanh\left(W_j \cdot \begin{bmatrix} v_i \\ h_j \end{bmatrix} + b\right) \quad (3)$$

where $W \in \mathbb{R}^{(d+d_{lstm})\times 1}$ and $b \in \mathbb{R}^{1\times 1}$.

We compute the adaptive vector for each unit in aspect target. In the end, we will get as many adaptive vectors as the number of aspect target units. Each of these adaptive vectors concentrates the influence that sentence context imposed on aspect target, namely it enriches the semantic meaning of aspect target by extracting correlation from intra-sentence context. For instance, the meaning of "*apple*" in our previous example.

5. Sequence learning of aspect target

Having obtained the adaptive vector of each aspect target unit, we will extract the sequential information in aspect target sequence. We find that sequential information within aspect target sequence is crucial in representing the meaning of an aspect target. Recall the previous example of "*red apple*". Only by connecting "*red*" and "*apple*" will we have a complete impression of the new iPhone 7 in red coating. Isolating the two aspect words will be harmful. Therefore, we employ the second LSTM neural network [41] to encode such sequential information.

Specifically, we concatenate the adaptive vector of each aspect target unit to form an aspect target sequence. This sequence will be fed to a LSTM as input. In the end, we take out the hidden output H_L from the last LSTM cell as the representation of aspect target sequence.

6. Fusion of multi-granularity representation

Unlike Latin languages such as English, Chinese written language is a type of pictogram, where its primitive forms symbolize certain meanings, such as characters '日 (sun)' and '月 (moon)'. With time went by, more complex meanings need to be represented in text. Thus, certain simple characters cluster together to form complex characters. For instance, '明 (shining)' composed of two sub-element characters '日 (sun)' and '月 (moon)'. The semantic relation between is both '日 (sun)' and '月 (moon)' emit light and bring brightness. Simple characters like '日' and '月' were thus called 'radical' when they appear in constructing complex characters. In order to represent abstract meanings, certain complex characters were clustered to form words. For instance, word '明星 (celebrity)' composed of character '明 (shining)' and '星 (star)'. Celebrities are shining stars in a sense.

Due to the above reasons, we understand modern Chinese text can be represented at three different granularities: radical, character and word. Inspired by Peng et al. [51], we represent Chinese text at three different granularities in our model and study the possible outcomes by fusion any of them.

In order to fit Chinese text into our deep learning framework, we represent Chinese text with embedding vectors. Particularly, we use the skip-gram model [53] to learn the embedding vectors at different granularities. Our training corpus contains about 8 million Chinese words, which is equal to 38 million Chinese characters or 150 million Chinese radicals. For word embedding vectors, we conduct word segmentation on the corpus using ICTCLAS [54] segmenter and then train with the skip-gram model. For character embedding vectors, we split each word in the corpus to individual characters, in which we still keep the order of characters. For radical embedding vectors, we decompose each character into radicals and concatenate them in the order from left to right. The decomposition was based on a Chinese character-radical look-up table we built using the Chinese character parser 'HanziCraft[1]'.

We design two fusion mechanisms (early and late) to merge three granularities. Early fusion concatenates different granularities of each aspect unit before aspect target sequence learning. Specifically, each aspect target word was represented by a concatenation of its sub-granular representations before it was sent to aspect target sequence learning. The output from aspect target sequence learning step will be fed to a *softmax* classifier.

Late fusion concatenates different granularities after aspect target sequence learning. Thus, for each granularity, an aspect target sequence representation will be obtained first. These representations will be concatenated and fed to a *softmax* classifier.

6.1. Early fusion

We have already proposed the fundamental model ATSM-S for ABSA. However, the performance of the model should largely depend on the representation of text because the embedding vectors are the initial input to the model. To this end, we plan to incorporate the multi-level representation of Chinese text. ATSM-S emphasizes the role of word level of representation. We incorporate multi-level representation for aspect word to further improve the accuracy of aspect words. Instead of using only word level representation in ATSM-S, we explore using either two or three level of representation, namely radical, character and word level.

Specifically for each sentence, we construct three types of sentence strings: a word string, a character string, and a radical string. In each of the string, aspect words are decomposed into the corresponding level. For each unit in the decomposed string of aspect words, it will learn an attention vector between the whole sentence string. For example, given an aspect word '工艺 (craftsmanship)'. One word attention vector will be learned from the word string. Two character attention vectors will be learned from the character string because the aspect word contains two characters, '工' and '艺'. Three radical attention vectors will be learned due to the aspect word can be decomposed into three radicals, '工', '艹' and '乙'. Then, we compute an average attention vector for each representation level. Three resulting average attention vectors will be concatenated and will be treated as a fusion of multi-level representation. As this fusion is a feature level fusion for aspect term, we call this fusion the early fusion. The fusion attention vector will be fed to a LSTM like in ATSM-S. The final output from the LSTM will be fed to a *softmax* classifier. Graphical illustration is in Fig. 2.

6.2. Late fusion

Unlike the early fusion, where the fusion takes place at the feature level, in late fusion, the fusion of multi-level representation happens at classification step.

In late fusion, our ATSM-S is used intact at three different levels independently. As shown in Fig. 2(b), the green break line box stands for ATSM-S working at the word level. While the purple box stands for working at the character level and the blue box stands for working at the radical level. We take out the last LSTM hidden output from each level and concatenate them. The resulting concatenated vector will be fed to a softmax classifier.

Late fusion differs from early fusion in assuming that semantics within a sentence should be unified at representation level. In other words, the semantics of aspect terms at a single level can hardly help extract semantics at other representation levels. Thus, in late fusion, the ATSM-S works merely on one level at a time and combines at final classification step only.

7. Evaluation

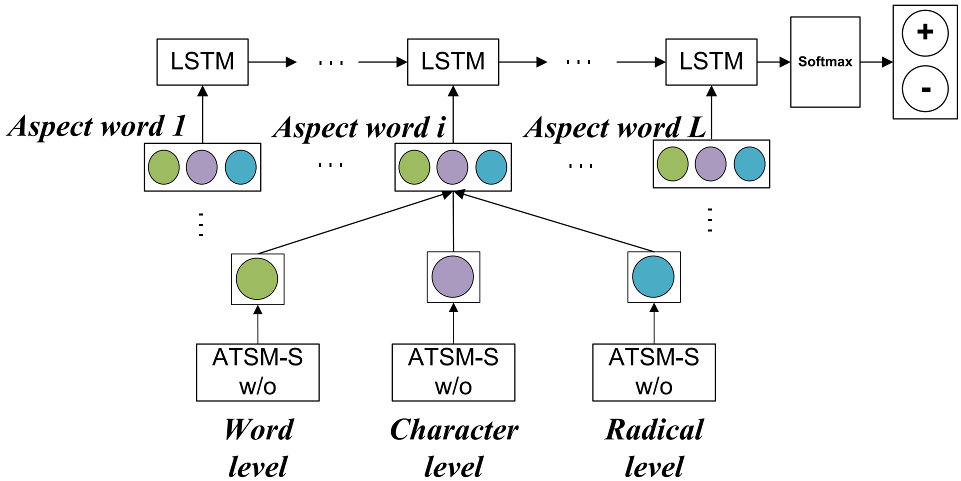
In this section, we present our evaluations in three steps. The first step will conduct experimental evaluations of various methods to model aspect target sequence, as well as adaptive embedding learning. The second step will compare the proposed ATSM-S with the state of the art. The last step will evaluate the improvement by fusions of granularities. We used Tensorflow and Keras to implement our model. All models used AdagradOptimizer with a learning rate of 0.1 and a L2-norm regularizer of 0.01. Each minibatch contains 50 samples. We report the average testing results of each model for 50 epochs in 5-fold cross validation.

7.1. Datasets

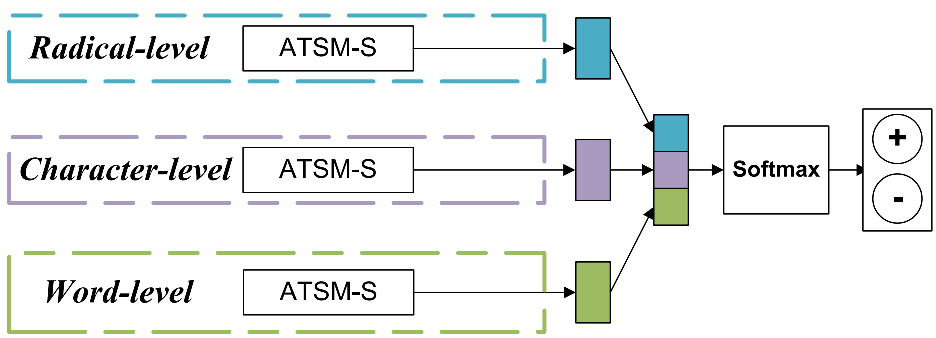
Four Chinese datasets from four domains were used in our experiments[2]. They are Chinese aspect datasets we collect from [55]. They contains reviews in four domains: notebook, car, camera, and phone. Aspect targets were originally tagged by Zhao et al. [56]. Then, we manually labeled the sentiment polarity towards each aspect target as either positive or negative. The metadata of the dataset was displayed in Table 2. English dataset used in our experiments was a subset from the SemEval2014 [57], which contains reviews from two different domains: restaurant and laptop. We select the reviews which contain multi-word aspect target only and results in a subset of 2309 reviews (30% of the original dataset).

[1] http://hanzicraft.com.

[2] Datasets download at: http://sentic.net/chinese-review-datasets.zip.



(a) Early fusion.(ATSM-S w/o stands for ATSM-S without sequence learning of aspect target)



(b) Late fusion.

Fig. 2. Fusion mechanisms.

Table 2
Metadata of Chinese dataset.

	Notebook	Car	Camera	Phone	Overall
Positive	417	886	1558	1713	4574
Negative	206	286	673	843	2008
Percentage of multi-word aspect	38.20	36.95	44.55	40.49	41.02

7.2. Comparison methods

In our experiments, we include three types of baseline comparisons. The first type includes the self-variants of ATSM-S, which examines the validity of each module of our model. The second type is the state-of-the-art methods in ABSA task, which tests the overall performance. The last type explores how the fusion of multi-grained representation of Chinese will affect the ABSA task.

7.2.1. Variants of ATSM-S

As there are two major modules of ATSM-S, namely adaptive embedding learning and sequence learning of aspect target, we design different variants for either of the modules to validate its superiority. **ATSM-v1** and **ATSM-v2** were designed to examine adaptive embedding module. **ATSM-v3, ATSM-v4** and **ATSM-v5** were designed to examine the module of sequence learning of aspect target.

1. **ATSM-v1**: The first variant of ATSM-S. It eliminates sentence sequence learning step in ATSM-S. In the following steps, it replaces sentence level LSTM hidden state outputs with initial word embeddings.
2. **ATSM-v2**: The second variant of ATSM-S. It removes the adaptive embedding learning module. Instead, it takes the sentence level LSTM hidden state outputs of each aspect target word as the input to aspect target sequence learning module.
3. **ATSM-v3**: The third variant of ATSM-S. It replaces the module of aspect target sequence learning with an average of aspect target word embeddings. It does not extract the aspect target sequence information.
4. **ATSM-v4**: The fourth variant of ATSM-S. It opts for different modeling of aspect target sequence. Specifically, it replaces the LSTM at aspect target sequence level in ATSM-S with a CNN (convolutional neural network).

Table 3
Variants of ATSM-S on Chinese datasets at word level.

	Notebook		Car		Camera		Phone		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ATSM-v1	69.98	**62.60**	80.88	55.09	78.27	66.81	80.83	68.51	81.95	77.05
ATSM-v2	66.94	40.58	75.59	42.29	69.94	47.08	67.72	42.58	70.24	45.64
ATSM-v3	74.15	62.04	80.71	57.24	78.09	69.49	81.65	71.59	81.89	76.47
ATSM-v4	74.80	60.00	82.94	59.43	82.34	69.86	84.11	73.24	85.76	**80.84**
ATSM-v5	73.35	58.67	79.61	56.65	78.31	68.33	80.56	70.03	82.42	77.84
ATSM-S	**75.59**	60.09	**82.94**	**64.18**	**82.88**	**72.50**	**84.86**	**75.35**	**85.95**	80.13

Table 4
Accuracy and Macro-F1 results on Chinese datasets at word level.

	Notebook		Car		Camera		Phone		Overall	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SVM	66.92	40.09	75.60	43.04	69.83	41.11	67.02	40.11	69.49	41.00
LSTM	74.63	62.32	81.99	58.83	78.31	68.72	81.38	72.13	82.71	78.28
Bi-LSTM	74.15	**63.09**	81.82	56.42	78.35	69.35	81.45	70.42	82.22	76.93
TD-LSTM	67.10	40.58	76.53	46.47	70.48	51.46	69.17	52.40	70.56	51.72
TC-LSTM	68.39	50.57	76.19	50.99	70.88	54.79	69.88	54.26	70.66	53.60
MemNet	69.10	53.51	75.55	51.01	70.59	55.13	70.29	55.93	72.86	55.99
ATSM-S	**75.59**	60.09	**82.94**	**64.18**	**82.88**	**72.50**	**84.86**	**75.35**	**85.95**	**80.13**

5. **ATSM-v5**: The last variant of ATSM-S. Unlike ATSM-v4 which models the aspect target sequence with a CNN, ATSM-v5 concatenates their embeddings and feeds to a nonlinear neural layer.
6. **ATSM-S**: There are three sub-categories of this type. The first one is ATSM-S working on the word level. The other two are ATSM-S models working on character and radical level, respectively. These three variants do not have any fusion of representation levels and, hence, serve as baselines towards our fusion mechanism.

7.2.2. State of the art

We include several state-of-the-art methods: SVM, LSTM, Bi-LSTM, TD-LSTM, TC-LSTM, MemNets and ATSM-S.

1. **SVM**: SVM classifier trained on the surface and parse features, such as unigram, bigram, POS tags. Aspect target features were concatenated to sentence features.
2. **LSTM**: The typical sequence modeling method that unveils the sequential information from the head to the tail of the sentence. It does not pay special attention to aspect term in the sentence. For long sentences, this method leverages more on the ending words in the sentence than the beginning words. Thus, for the case when aspect term appears in the head of the sentence, it may not work well.
3. **Bi-LSTM**: It adds a reverse sequential learning step to LSTM. Bi-LSTM models both head to tail and tail to head sequential information, however, it does not distinguish the aspect term with context words in ABSA.
4. **TD-LSTM**: Instead of attending to the complete length of a sentence like LSTM, TD-LSTM [31] used a forward and a backward sequence that ends immediately after including aspect term. It extracts sentence semantics prior and after aspect term separately.
5. **TC-LSTM**: In addition to TD-LSTM, TC-LSTM appended the sentence word embedding with aspect target embedding. It hopes this procedure could explicitly capture the interaction between aspect word and sentence context word. Nevertheless, this method treated the sequential information from aspect target sequence and sentence word sequence with equal importance. They did not model the aspect target sequence.
6. **MemNet**: This method took out the aspect word and looked for correlation with sentence context words. The problem of this method is it did not use the sequential information within aspect target sequence. In our experiments on both English and Chinese datasets, we conducted various experiments using hop number from one to nine of this model and reported the best results.

7.2.3. Fusion comparison

1. **ATSM-F**: Based on ATSM-S, it fuses not only three representation granularities but also any two representation granularities, in both early and late manner. There are 11 different settings in this experiment. It evaluates whether fusion will improve from single granularity and which combination benefits the final result most.

Unlike all the previous methods, the novelties of our model are three folds. Firstly, we adapt the general word embedding to ABSA task by encoding sentence context. Secondly, we explicitly model the aspect target sequence. Finally, we design fusion mechanism on ATSM-S to take advantage of three granularities of Chinese representations. The experimental results are shown in Tables 3, 4 and 6.

7.3. Result analysis

7.3.1. Self comparison

In this section, we compare different variants of ATSM-S with experiments on Chinese datasets. The experimental results were shown in Table 3.

We can observe that ATSM-S achieves the highest accuracy in all the datasets and highest F-score in three datasets. It generally demonstrates the validity of our model design. In order to elaborate more details, we conduct the comparisons with model variants.

The only difference between ATSM-v1 and ATSM-S is that the former omits sentence sequential information. The decrease of performance from ATSM-v1 proves that ATSM-S has successfully encoded sentence sequence. Even if the sentence sequential information is correctly learned, the overall performance cannot be guaranteed. This is illustrated by ATSM-v2, which encodes sentence sequence but does not learn adaptive embedding. Since ATSM-S learns adaptive embedding on top of ATSM-v2, a more accurate aspect target representation is learned and, hence, contributes to the final performance.

Table 5
Accuracy and Macro-F1 results on single-word/multi-word aspect target subset from SemEval2014.

	ATSM-S (word)		MemNet		TC-LSTM		TD-LSTM		Bi-LSTM	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Multi-word aspect English dataset	**65.37**	36.54	58.54	42.16	63.58	43.87	63.48	**47.16**	62.19	45.02
Single-word aspect English dataset	**75.39**	54.12	67.83	52.70	59.33	49.58	68.38	52.95	72.80	**54.35**

ATSM-v3, -v4 and -v5 distinguish with each other in the way they model aspect target sequence. -v3 takes the average of aspect target word embeddings, which ignores aspect target sequential information. -v4 models the aspect target sequence with a CNN. -v5 models the sequence with the middle layer from an MLP. In comparison, ATSM-S models the sequence with a LSTM. From the table, we can conclude that LSTM achieves the best results compared with other variants. It further proves our assumption that the aspect target sequential information plays a significant role in ABSA task.

7.3.2. Peer comparison

From Table 4, we can see that ATSM-S beats other state-of-the-art methods by around 1–4% in all four datasets and in one overall dataset.

We believe the first reason why ATSM-S wins over other methods is that we explicitly learned the adaptive meaning of each aspect target unit. The adaptive embedding of each aspect target unit not only carries semantics from general word embedding but also encodes semantics within the sentence. In comparison, the baseline model ATSM-v2 eliminates sentence sequence learning step and, hence, results in a poor adaptive embedding.

The second reason is that we explicitly modeled the aspect target sequence. Other state-of-the-art works either ignored the aspect target sequence [30,32] or treated aspect target sequence as equal importance as sentence sequence [31]. Both of the approaches did not render enough emphasis on aspect target sequence. To validate its importance, we designed the second baseline variant of ATSM-S, which is ATSM-v3. It differs from ATSM-S only in ignoring target sequence information. The sharp decrease of performance from ATSM-S to ATSM-v3 validated our assumption.

The difference between ATSM-S with the popular attention model where the aspect is embedded by a LSTM layer are twofold. Firstly, ATSM-S specifically encodes aspect target sequential information. Whereas attention model treated aspect target as an averaged embedding vectors. Secondly, aspect target sequential information was given higher importance than sentence sequential information in ATSM-S. Whereas attention model treated the two sequences of equal importance.

Since ATSM-S specializes in modeling aspect target sequence, we conduct further experiments to test whether it is language independent. Thus, we removed reviews from English SemEval2014 dataset that had single-word aspect target only (eg. pasta) and collected the remaining reviews that all had multi-word aspect target (eg. build quality) to form a multi-word aspect target subset. Meanwhile, we also collected the removed reviews to form a single-word aspect target subset.

Experimental results on these two subsets were shown in Table 5 in comparison with the top few state-of-the-art works, namely MemNet, TC-LSTM, TD-LSTM, and Bi-LSTM. In the single-word case, the proposed ATSM-S achieved the highest accuracy. It is beyond our expectation because the module of sequence learning of aspect target from ATSM-S would not work on single-word aspect target. On the other hand, it validates the contribution of adaptive embedding learning module, which learns an accurate presentation of aspect target. In the other case, the table shows ATSM-S is the best at predicting multi-word aspect sentiment polarity in English dataset than the state of the art. The main reason is that our model explicitly learns adaptive embedding and aspect target sequence, where the latter is crucial. A visual analysis is provided in the next section.

7.4. Visual case study

We visualize the difference between ATSM-S and a typical baseline model (MemNet) via a case study from English SemEval 2014 dataset.

As shown in Fig. 3, we plot the heatmap of attention weights. The deeper the color is, the heavier weight the word has. Our ATSM-S has two heatmaps due to we explicitly learn adaptive embedding for each aspect target word (‘Korean’ and ‘dishes’). Whereas MemNet only has one, because it averages the word embeddings of aspect target and learns a sentence-level attention. It is apparent that either of our aspect unit adaptive embedding captures a key opinion word in the sentence, which are ‘affordable’ and ‘yummy’. In the later step of aspect target sequence learning, both of the opinion words will be captured and reflected in our final model output. Nevertheless, the heatmap from MemNet is the final model output, which unfortunately misses a crucial part of the opinionated content. The case study provides an intuitive explanation of why our ATSM-S prevails.

7.5. Granularity and fusion analysis

In the last set of experiments, we evaluated if multiple granularities in Chinese text representation will improve the performance of our model further. As shown in Table 6, we performed ATSM-S at each of the three granularities as baselines. We also applied ATSM-F in both early fusion mode and late fusion mode. The ATSM-F in late fusion of word and character level achieved high results in four out of five datasets. It beat ATSM-S in almost any single granularity situation (except word level on Car dataset. However, it is close to the performance of ATSM-S at word level.), which proved that a fusion of multiple granularities promoted the sentiment inference over single granularity.

Generally, we could see ATSM-S at character level produces the top few results in all single granularity cases. However, we found that word level performed better than character level on notebook and car datasets, a deeper look into those two datasets revealed the possible cause of biased data distribution. After computing variances of experimental results for each dataset, we found that the average variance of Notebook and Car dataset is 1.7 times bigger than the average variance of all five datasets at the word level, and 1.29 times larger than the average at the character level. This indicates that our model performances in these two datasets were less robust than the other three datasets. Furthermore, we found that the number of unique aspect target in these two datasets were relatively higher compared with their dataset size. This explains why our model did not generalize well on these two datasets. Moreover, due to their smaller size compared, we believe all the above caused the character level performed worse than word level. In comparison, in the other three datasets, whose variances were well below average, character level outperformed

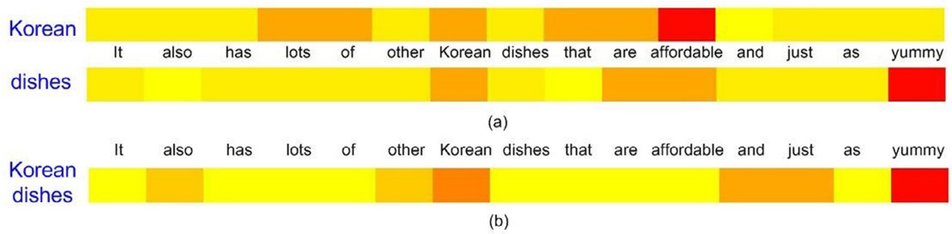


Fig. 3. Visual attention weights of each word in the example. (a) is from ATSM-S. (b) is from baseline model.

Table 6
Accuracy results of multi-granularity with and without fusion mechanisms. (W, C, R stands for word, character and radical level, respectively. + means a fusion operation.)

			Notebook	Car	Camera	Phone	Overall
ATSM-S		W	75.59	**82.94**	82.88	84.86	85.95
		C	74.32	81.56	87.98	88.34	88.50
		R	69.92	75.68	77.19	78.09	79.87
ATSM-F	Early fusion	W+C	**77.52**	82.16	86.55	87.13	**89.38**
		W+R	68.38	76.61	77.73	78.29	83.64
		C+R	69.99	77.81	80.73	80.90	87.41
		W+C+R	69.99	77.55	78.76	78.91	84.94
	Late fusion	W+C	73.67	**82.93**	**88.30**	**88.46**	**89.33**
		W+R	67.26	78.23	80.68	84.94	86.43
		C+R	67.58	79.00	87.63	88.14	88.50
		W+C+R	67.91	78.15	87.98	88.07	89.30

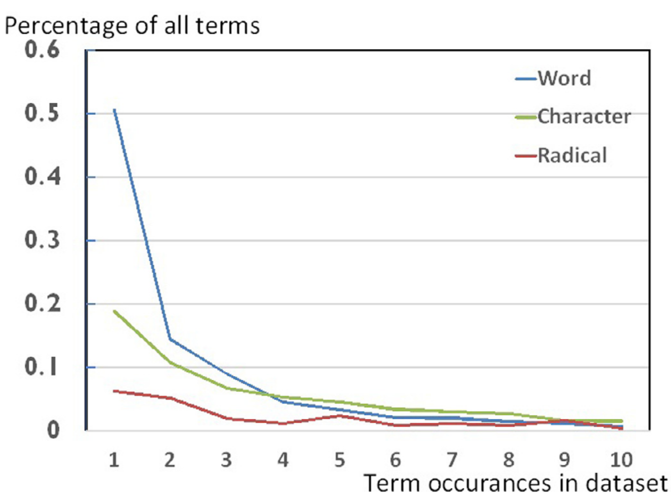


Fig. 4. Percentage of number of terms in corresponding to from 1 to 10 occurrences in three-level representation.

word level with an obvious advantage. This is consistent with our expectation, as working at character level will wipe out the negative effects brought by word segmentation.

It also explains why ‘W+C’ achieved the top few results. Sentiment information from the character level is effectively extracted and properly maintained with the help of effective character embeddings and ATSM-S. Fused with the word level information, the character level sentiment information improved the overall performance. However, working at radical level did not improve the performance much, if not exacerbating the situation. Thus, it drove us to analyze the reason. We studied the aspect target distribution for each of the three representation granularities with our experimental datasets as examples. As shown in Fig. 4, we plot the percentage of token types (i.e, unique tokens) of three different granularities appearing less than 10 times in the whole dataset. It is obvious to find that the representation of character level largely reduces the percentage of token types occurring only once in the word level. That is to say, character level representation significantly reduces the data sparsity of rare words by decomposing the words into characters. This explains why character level representation could greatly improve from word level. Radical level, on the contrary, does not reduce much the percentage from character level. One possible reason could be the ineffectiveness of our radical embedding vectors. In training the radical embeddings, we did not distinguish the radicals by phoneme and morpheme. This may introduce errors to radical embeddings, as phonemes do not carry semantics. These radical embeddings could affect the final results drastically. That being said, the radical level representation is still comparable to other baseline models. It indicates the potential of introducing radical level representation in the task of Chinese ABSA.

We elaborate why ATSM-F in late fusion mode achieves the top few performances as below. Our fusion mechanisms experimented on possible combinations of three granularities extract multi-granular semantics in the task of ABSA. In comparison, late fusion has a flat structure, while early fusion has a hierarchical structure. Using a flat structure means the semantic encoded by each granularity is relatively independent. Whereas using a hierarchical structure breaks down the semantic flow along each granularity. Because it fuses the representations of multi-granularity for each aspect target word, semantics at character and radical level were cut off by the boundary of words.

8. Conclusion

In this paper, we investigated the problem of aspect-level sentiment analysis from a new perspective, in which the aspect target sequence dominates the final result. Accordingly, we proposed ATSM-S, which prevailed the state of the art in multi-word aspect sentiment analysis on SemEval2014. Moreover, our model specifically catered to the multi-granularity representations of Chinese text. By designing a late fusion method, ATSM-F outperformed all state-of-the-art works in three Chinese review datasets. As one of the first attempts to exploiting multi-granularity nature of Chinese ABSA, this work paves the path for potentially wider application scenarios in Chinese natural language processing.

References

[1] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, A Practical Guide to Sentiment Analysis, Springer, Cham, Switzerland, 2017.
[2] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A review of affective computing: From unimodal analysis to multimodal fusion, Inf. Fusion 37 (2017) 98–125.
[3] S. Poria, E. Cambria, D. Hazarika, N. Mazumder, A. Zadeh, L.-P. Morency, Context-dependent sentiment analysis in user-generated videos, in: Proceedings of the ACL, 2017, pp. 873–883.
[4] I. Chaturvedi, E. Cambria, R. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: survey and challenges, Inf. Fusion 44 (2018) 65–77.
[5] S.L. Lo, E. Cambria, R. Chiong, D. Cornforth, Multilingual sentiment analysis: From formal to informal and scarce resource languages, Artif. Intell. Rev. 48 (4) (2017) 499–527.
[6] A. Bandhakavi, N. Wiratunga, S. Massie, P. Deepak, Lexicon generation for emotion analysis of text, IEEE Intell. Syst. 32 (1) (2017) 102–108.
[7] M. Dragoni, S. Poria, E. Cambria, OntoSenticNet: a commonsense ontology for sentiment analysis, IEEE Intell. Syst. 33 (2) (2018).
[8] E. Cambria, S. Poria, D. Hazarika, K. Kwok, SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Proceedings of the AAAI, 2018.
[9] L. Oneto, F. Bisio, E. Cambria, D. Anguita, Statistical learning theory and ELM for big social data analysis, IEEE Comput. Intell. Mag. 11 (3) (2016) 45–55.
[10] A. Hussain, E. Cambria, Semi-supervised learning for big social data analysis, Neurocomputing 275 (2018) 1662–1673.
[11] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, E. Cambria, Learning word representations for sentiment analysis, Cognit. Comput. 9 (6) (2017) 843–851.
[12] E. Cambria, A. Hussain, Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis, Springer, Cham, Switzerland, 2015.
[13] F. Xing, E. Cambria, R. Welsch, Natural language based financial forecasting: a survey, Artif. Intell. Rev. doi:10.1007/s10462-017-9588-9 (2018).
[14] M. Ebrahimi, A. Hossein, A. Sheth, Challenges of sentiment analysis for dynamic events, IEEE Intell. Syst. 32 (5) (2017) 70–75.
[15] E. Cambria, A. Hussain, T. Durrani, C. Havasi, C. Eckl, J. Munro, Sentic computing for patient centered application, in: Proceedings of the ICSP, IEEE, 2010, pp. 1279–1282.
[16] A. Valdivia, V. Luzon, F. Herrera, Sentiment analysis in tripadvisor, IEEE Intell. Syst. 32 (4) (2017) 72–77.
[17] R. Mihalcea, A. Garimella, What men say, what women hear: finding gender-specific meaning shades, IEEE Intell. Syst. 31 (4) (2016) 62–67.
[18] S. Cavallari, V. Zheng, H. Cai, K. Chang, E. Cambria, Learning community embedding with community detection and node embedding on graphs, in: Proceedings of the CIKM, 2017, pp. 377–386.
[19] C. Xu, E. Cambria, P.S. Tan, Adaptive two-stage feature selection for sentiment classification, in: Proceedings of the SMC, IEEE, 2017, pp. 1238–1243.
[20] A. Zadeh, P.P. Liang, S. Poria, P. Vij, E. Cambria, L.-P. Morency, Multi-attention recurrent network for human communication comprehension, in: Proceedings of the AAAI, 2018.
[21] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, M. Huang, Augmenting end-to-end dialog systems with commonsense knowledge, in: Proceedings of the AAAI, 2018.
[22] E. Cambria, S. Poria, A. Gelbukh, M. Thelwall, Sentiment analysis is a big suitcase, IEEE Intell. Syst. 32 (6) (2017) 74–80.
[23] Y. Ma, E. Cambria, S. Gao, Label embedding for zero-shot fine-grained named entity typing, in: Proceedings of the COLING, 2016, pp. 171–180.
[24] Y. Xia, E. Cambria, A. Hussain, H. Zhao, Word polarity disambiguation using Bayesian model and opinion-level features, Cognit. Comput. 7 (3) (2015) 369–380.
[25] E. Cambria, S. Poria, R. Bajpai, B. Schuller, SenticNet 4: a semantic resource for sentiment analysis based on conceptual primitives, in: Proceedings of the COLING, 2016, pp. 2666–2677.
[26] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, E. Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Frankl. Inst. (2018) doi:10.1016/j.jfranklin.2017.06.007.
[27] N. Majumder, S. Poria, A. Gelbukh, E. Cambria, Deep learning-based document modeling for personality detection from text, IEEE Intell. Syst. 32 (2) (2017) 74–79.
[28] S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, in: Proceedings of the COLING, 2016, pp. 1601–1612.
[29] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM, in: Proceedings of the AAAI, 2018.
[30] Y. Wang, M. Huang, L. Zhao, X. Zhu, Attention-based LSTM for aspect-level sentiment classification, in: Proceedings of the Conference on EMNLP, 2016, pp. 606–615.
[31] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, arXiv:1512.01100 (2015).
[32] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the EMNLP, 2016, pp. 214–224.
[33] G. Qiu, B. Liu, J. Bu, C. Chen, Opinion word expansion and target extraction through double propagation, Comput. Linguist. 37 (1) (2011) 9–27.
[34] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowl. Based Syst. 108 (2016) 42–49.
[35] J. Yu, Z.-J. Zha, M. Wang, T.-S. Chua, Aspect ranking: identifying important product aspects from online consumer reviews, in: Proceedings of the Forty-Ninth ACL: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 1496–1505.
[36] W. Wang, S.J. Pan, D. Dahlmeier, X. Xiao, Coupled multi-layer attentions for co-extraction of aspect and opinion terms., in: Proceedings of the AAAI, 2017, pp. 3316–3322.
[37] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 813–830.
[38] J. Zhu, H. Wang, B.K. Tsou, M. Zhu, Multi-aspect opinion polling from textual reviews, in: Proceedings of the Eighteenth ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 1799–1802.
[39] T. Mullen, N. Collier, Sentiment analysis using support vector machines with diverse information sources., in: Proceedings of the EMNLP, 4, 2004, pp. 412–418.
[40] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, Nrc-canada-2014: detecting aspects and sentiment in customer reviews, in: Proceedings of the Eighth International Workshop on Semantic Evaluation (SemEval), 2014, pp. 437–442.
[41] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[42] W.Y. Zou, R. Socher, D.M. Cer, C.D. Manning, Bilingual word embeddings for phrase-based machine translation., in: Proceedings of the EMNLP, 2013, pp. 1393–1398.
[43] X. Zheng, H. Chen, T. Xu, Deep learning for chinese word segmentation and pos tagging., in: Proceedings of the EMNLP, 2013, pp. 647–657.
[44] J. Xu, J. Liu, L. Zhang, Z. Li, H. Chen, Improve chinese word embeddings by exploiting internal structure, in: Proceedings of the NAACL-HLT, 2016, pp. 1041–1050.
[45] X. Shi, J. Zhai, X. Yang, Z. Xie, C. Liu, Radical embedding: delving deeper to chinese radicals., in: Proceedings of the ACL2, 2015, pp. 594–598.
[46] X. Chen, L. Xu, Z. Liu, M. Sun, H.-B. Luan, Joint learning of character and word embeddings., in: Proceedings of the IJCAI, 2015, pp. 1236–1242.
[47] Y. Sun, L. Lin, N. Yang, Z. Ji, X. Wang, Radical-enhanced chinese character embedding, in: Proceedings of the ICNIP, Springer, 2014, pp. 279–286.
[48] Y. Li, W. Li, F. Sun, S. Li, Component-enhanced chinese character embeddings, in: Proceedings of the EMNLP, 2015, pp. 829–834.
[49] R. Yin, Q. Wang, R. Li, P. Li, B. Wang, Multi-granularity chinese word embedding, in: Proceedings of the EMNLP, 2016, pp. 981–986.
[50] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[51] H. Peng, E. Cambria, X. Zou, Radical-based hierarchical embeddings for chinese sentiment analysis at sentence level, in: Proceedings of the FLAIRS, 2017, pp. 347–352.
[52] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv:1409.0473 (2014).
[53] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv:1301.3781 (2013).
[54] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, Q. Liu, HHMM-based chinese lexical analyzer ICTCLAS, in: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing-Volume 17, Association for Computational Linguistics, 2003, pp. 184–187.
[55] W. Che, Y. Zhao, H. Guo, Z. Su, T. Liu, Sentence compression for aspect-based sentiment analysis, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (12) (2015) 2111–2124.
[56] Y. Zhao, B. Qin, T. Liu, Creating a fine-grained corpus for chinese sentiment analysis, IEEE Intell. Syst. 30 (1) (2015) 36–43.
[57] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: aspect based sentiment analysis, in: Proceedings of the SemEval, 2014, pp. 27–35.