Attention-based word embeddings using Artificial Bee Colony algorithm for aspect-level sentiment classification

Zhang, M, Palade, V, Wang, Y & Ji, Z

DOI 10.1016/j.ins.2020.09.038
ISSN 0020-0255

Publisher: Elsevier

Attention-based Word Embeddings using Artificial Bee Colony Algorithm for Aspect-level Sentiment Classification

Ming Zhang ZMJIANGNAN@126.COM
Engineering Research Center of Internet of Things Technology Applications Ministry of Education,
School of IoT Engineering
Jiangnan University
1800. Lihu Road, Binhu District
Wuxi City, 214122, PR China

Vasile Palade VASILE.PALADE@COVENTRY.AC.UK
School of Computing, Electronics and Mathematics
Coventry University
Priory Street, Coventry, CV1 5FB, UK

Yan Wang WANGYAN@JIANGNAN.EDU.CN
Engineering Research Center of Internet of Things Technology Applications Ministry of Education,
School of IoT Engineering
Jiangnan University
1800. Lihu Road, Binhu District
Wuxi City, 214122, PR China

Zhicheng Ji ZCJI@JIANGNAN.EDU.CN
Engineering Research Center of Internet of Things Technology Applications Ministry of Education,
School of IoT Engineering
Jiangnan University
1800. Lihu Road, Binhu District
Wuxi City, 214122, PR China

Editor:

Abstract

Considering that most popular models solving aspect-level sentiment classification problems focus mainly on designing complicated neural networks to scale the importance of each word in the sentence, this paper addresses this problem from the view of a semantic space. Motivated by the fact that the senses of a word can be sophisticatedly embedded into the semantic space in terms of a distributed representation, this paper hypothesizes that each sense of a word can be represented by one or more specific dimensions, and thus the target of aspect-level sentiment classification can be simplified to searching the related dimensions for the aspects and sentiments concerned, by employing an attention mechanism. An attention vector (ATV) is designed for each aspect in terms of a specific task, which involves two sub-vectors, i.e., a dimension attention vector (DATV) and a sentiment attention vector (SATV). Specifically, the DATV determines the significances of different dimensions based on their correlations with an aspect; and the SATV allocates weights for the attributes of words, which are decided by sentiment polarities and part-of-speech (PoS) tagging. Given a sub-dataset related to a particular aspect, the ATV will be optimized by an artificial bee colony (ABC) algorithm with a support vector machine (SVD) classifier, the objective of which is to maximize the classification accuracy. Intrinsically, the DATV can reduce the ambiguity existing in polysemy, meanwhile, the SATV is an auxiliary means for the optimization of the DATV,

eliminating the misunderstandings caused by antonyms. Then, the optimized DATV will be applied on a convolutional neural network (CNN) model via simply scaling the pretrained word embeddings as inputs (named as ATV-CNN model). Experimental results show that the ATV-CNN model can have substantial advantages when compared with the state-of-the-art models.
Keywords: Aspect-level Sentiment Classification, Attention Mechanism, Word Embeddings, Artificial Bee Colony Algorithm, Support Vector Machine

1 Introduction

Sentiment analysis, also known as opinion mining, aims to analyze opinions and deals with notions like evaluations, attitudes, sentiments, and emotions towards individuals, products, social events, markets, and politics affairs (Liu, 2012), as expressed in the form of online comments, such as tweets, microblogs, reviews, and social networks. Sentiment classification is probably the most broadly studied branch of sentiment analysis, which aims to identify sentiment polarities, typically positive, neutral or negative, expressed in short texts or documents. Although previous research on sentiment classification has reported many achievements (Pang et al., 2002; Mullen & Collier, 2004; Pang & Lillian, 2008), it remains challenging to detect human emotions or sentiments from raw texts. Deep learning models have been considered as efficient tools for big data analysis, and has achieved remarkable performance in computer vision, pattern recognition and recommendation systems (Liu et al., 2017). The successful implementation of convolutional neural network (CNN) in document classification (Kim, 2014) triggered an upsurge in deep neural networks for sentiment analysis (Zhang et al., 2018), although there is considerable room for improvements. The other widely used models include recursive neural network (RNNs) (Socher et al., 2011), gated recurrent unit (GRU) (Tang et al., 2015), and long short-term memory (LSTM) (Wang et al., 2015).

Recently, aspect-level sentiment classification has attracted much attention, which is a fine-grained task that takes into consideration not only the overall contents, but also aspect information when determining the sentiment polarities of a sentence. The main reason behind is that the polarities regarding to different aspects may be contrary in the same sentence. For example, in the sentence “This apple looks nice, but the taste is bland.”, the polarity is positive in terms of the appearance aspect, whereas it turns to negative for the taste aspect. The primary concerns of aspect-level sentiment analysis can be thought of extracting an aspect from a sentence and then assigning polarity to it (Liu, 2012). Owing to the complexity of exploring and matching the sentiments for the corresponding aspects, simple implementation of deep neural networks models is inadequate. Currently, integrating attention mechanisms into deep learning models has become a good choice, which can benefit from enforcing the model towards the significant parts of a sentence, i.e. focusing on the contexts that are highly related with the desirable aspects (Wang et al., 2016a). The intuitions underlying most models rely on assigning an attention score to every word in a sentence via adopting outer attention networks to measure the relationships between contexts and aspects. It is unavoidable for these models to be faced with three problems: 1) creating suitable representations for the aspects; 2) designing sophisticated attention networks for aspect extraction; 3) determining the sentiment polarities for each aspect. As in (Ma et al., 2017), before performing sentiment classification, an interactive attention network (IAN) was introduced to learn attentions in contexts and targets, in order to generate separate representations for them. Unlike previous efforts that concentrate on regulating extra attention networks and assistive representations, the attention mechanism proposed in this paper puts emphasis on extracting aspect-specific and sentiment-specific dimensions in the semantic space.

Since distributed representations (also called word embeddings) proposed by (Mikolov et al., 2013a) can project words from a sparse bag-of-words (BoW) encoding onto a low-dimensional vector space, the deduced dense vectors, encoding both semantic and syntactic regulations (Mikolov et al., 2013b), have become baseline representations for words in sentiment analysis and achieved substantial improvements (Tang et al., 2016a; Giatsoglou et al., 2017). On the basis of comprehensively analyzing the characteristics of word embeddings, this work hypothesizes that

each dimension in the word space represents a specific sense for words, and conversely, the meanings of a word can be embedded into specific dimensions in the semantic space. Therefore, the aspect-level sentiment classification problem has been refined to determine the significances of dimensions regarding to an aspect rather than individual words in the sentences. Particularly, for each aspect in a sentence, a simple attention vector (ATV) is designed to detect the related dimensions, which can be optimized by an artificial bee colony (ABC) algorithm with a support vector machine (SVM) classifier, whose accuracy is used as fitness function. The ATV is composed of two sub-vectors. One is the dimension attention (DATV) sub-vector, with the purpose of adjusting the weights for spatial dimensions in the semantic space, and the other one is the sentiment attention (SATV) sub-vector, which can provide complementary help for guiding the search direction of the ABC algorithm, aiming at compensating the deficiencies existing in pretrained word embeddings. Finally, in terms of a specific aspect, the pretrained word embeddings will be scaled by the optimized DATV, which will be further used as the inputs of the CNN model proposed in (Kim, 2014), termed as ATV-CNN. Compared with the conventional models discovering the relations between aspects and contexts in the way of treating them as individual words, the horizons of the ATV-CNN model have been expanded to reveal the essence of aspects and sentiments in the sense of semantic space, which is more universal and stable for a given task. The experimental results show that the ATV-CNN model has superiorities and can achieve the state-of-the-art performance.

The rest of paper is structured as follows. Section 2 describes the related work. Section 3 gives some preliminaries and basic knowledge on the problem under study. Section 4 introduces the motivations and presents the framework of the proposed approach. The experimental studies are reported and discussed in Section 5, and some conclusions are drawn in Section 6.

2 Related Work

This section will briefly review related works on general and aspect-level sentiment classification as well as the applications of EAs for sentiment analysis.

2.1 Traditional Methods for Aspect-level Sentiment Classification

Basically, the processes of aspect-level sentiment analysis include three steps: identifying aspect-sentiment pairs in the text, classifying the pairs, and aggregating the sentiment values of all pairs in order to make an overall judgement (Schouten & Frasincar, 2015; Tsytsarau & Palpanas, 2012). Most of the traditional approaches lay emphasis on detecting salient features for aspects through statistical information or syntactic rules. For example, Hai et al. (2011) first generated sustainable association rules from a co-occurrence matrix created by the bipartite of sentiment words and explicit aspects, and then detected implicit aspects by restricting the appearances of sentiments and aspect words as rule antecedents and rule consequents, respectively. Besides, sentiment lexicons are efficient resources for aspect detection, which can straightforwardly assign sentiment scores for the individual words appearing around the aspects. Zhu et al. (2009) determined the polarities for the aspect related segments of a sentence using a sentiment lexicon; and Mullen & Collier (2004) incorporated a variety of diverse information sources and allocated values to targeted words and phrases. It is noticeable that feature extraction is a time-consuming process and the quality of features can significantly influence the classification performance.

2.2 Neural Network Models for Aspect-level Sentiment Classification

As for neural network models, CNN models are able to capture local information, whereas they are incapable of retaining long-term dependencies existing in texts. This drawback can be addressed by LSTM based models, which have sequential architectures that can model long texts with multiple sentences and flexibly capture the semantic relations between a target and its contexts. Hybrid models composed of CNN and LSTM have attracted attention in recent years, in order to

effectively exploit the advantages of both model types. For example, Wang et al. (2016b) proposed a regional CNN-LSTM model, which used the CNN to estimate regional affections with each sentence viewed as a region and then LSTM was used to sequentially integrate the regional information for the sake of predicting the valence-arousal (VA) ratings of texts. In recent years, LSTM based models have been rapidly developed. By taking into account the target information, Tang et al. (2016b) introduced target-dependent LSTM (TD-LSTM) and target-connection LSTM (TC-LSTM) models, where TD-LSTM employed the preceding and following contexts surrounding a target as feature representations; and TC-LSTM further extended the TD-LSTM model by making use of the connections between each target-context pair for the generation of sentence representation. A hierarchical LSTM model was leveraged in (Ruder et al., 2016) to exploit both intra- and inter-sentence relations under the hypothesis that individual sentences in a review could elaborate upon each other, and the classification of each sentence should depend on the knowledge of the whole review structure and sentential context.

Considering the efficiency of attention mechanism, Liu and Zhang (2017) presented an attention-based LSTM model to measure the contribution of each word to a targeted sentiment polarity and subsequently induce the overall attention values for sentences. Chen et al. (2017) introduced a multiple-attention framework to synthesize sentiment features scattered in a long sentence, which weighted the hidden states in an LSTM model in accordance with their relative positions to the target; then, various attentions would be paid onto the position-weighted memory, with the results combined by GRUs. Besides, the memory network has become an alternative for aspect-level sentiment analysis (Sainbayar et al., 2015). Li et al. (2017) used a deep memory network to imitate attitude identification as an end-to-end process, building a feedback architecture between target detection and polarity classification using a deep memory network. Scenarios in (Majumder et al., 2018) showed that the sentiment of one aspect in a sentence was highly influenced by the presence of others due to the existence of conjunctions, thus, after independently generating aspect-aware representations for sentences using an attention-based GRU, Majumder et al. (2018) designed a repeated matching mechanism based on a memory network to discover the relations between the targeted aspect and other aspects.

2.3 Applications of EAs on Sentiment Classification

EAs, which is an important branch of derivative-free techniques, are regarded as efficient tools for solving difficult optimization issues, such as non-convex, multi-modal, non-differentiable and other challenging problems. Generally, EAs were mainly used to perform feature extraction for sentiment classification in the previous research. Abbasi et al. (2008) designed an entropy weighted genetic algorithm (EWGA) to obtain better assessment of stylistic and syntactic features for sentiment classification of English and Arabic content. Motivated by the observation that a word's polarity can be determined by calculating its relative co-occurrence counts with paradigm words (e.g. "good" and "bad"), Carvalho et al. (2014) used genetic algorithms (GA) to select paradigm words from a set of candidate words in order to establish a statistical model for the classification of tweets.

On the other hand, certain works pay attention to the parameter tuning for sentiment classification. For example, Keshavarz and Abadeh (2017) formulated the sentiment classification as an optimization problem, which can be solved by a novel genetic algorithm (ALGA) with the goal set at finding optimum sentiment lexicons. Basari et al. (2013) used partical swarm optimization (PSO) to optimize the kernel factors of an SVM classifier, aiming to solve binary classification problems. To the best of our knowledge, there is limited work employing EAs for aspect-level sentiment classification. Gupta et al. (2015) proposed a PSO-ASent approach based on the principle of the PSO algorithm, which can conduct automatic feature selection for aspect term extraction and sentiment classification within the learning framework of conditional random fields (CRF). Generally, more attentions should be paid on the development of EAs with respect to the aspect-level sentiment classification.

3 Preliminaries

3.1 Aspect-level Sentiment Classification

Sentiment analysis mainly studies opinions which express or indicate positive or negative sentiments. According to the definition in (Liu, 2012), five components are involved in an opinion, including an entity, an aspect of the entity, a sentiment about the aspect, an opinion holder who expressed the opinion, and a time when the opinion was delivered by the opinion holder. In detail, entities could be products, services, persons, organizations, issues, events, etc.; aspects refer to the attributes of an entity, and there may exist a hierarchy of parts or sub-parts relationships between them; a sentiment mainly refers to positive, negative or neutral, or is represented by some intensity ratings, e.g. marked as 1-5 stars on the Web; besides, the opinion holder and the time are necessary for the description of an opinion, since the attitudes of a person towards an entity may be change along with the time. As in the review: "John Green, 15/01/2019: The phone I bought last month is amazing. Its battery life is long and the picture quality is satisfying.", the opinion holder is John Green, who commented on a camera (i.e. entity) with positive opinions towards both the battery life and the picture quality aspects.

In general, the aspect-level sentiment classification contains two main processes. One is aspect extraction, aiming to extract aspects of a concerned entity from a sentence, and the other one is aspect sentiment classification, the task of which is to determine the sentiment polarities of various aspects. In reality, the current benchmark datasets used, such as the restaurant data collected by Pontiki et al. (2014), have clearly annotated the aspects for each sentence and the corresponding positions of words regarding to each aspect. In this case, the aspect extraction process could be omitted, however, the step of automatically detecting the related information for a specific aspect is still necessary since the sentiment polarities for one sentence could be opposite when different aspects are considered.

Suppose that there are m aspects for a given dataset D, $asp_{p\in\{0,1,\dots,m-1\}}$ represents the p_{th} aspect; D_p represents the corresponding sub-dataset with regard to aspect asp_p with $|D_p|$ denoting the number of sentences contained; D_{pq} represents the q_{th} sentence for aspect asp_p. Let w_i be the i_{th} word in the vocabulary $|V|$. Given an aspect-sentence pair (asp_p, D_{pq}), the goal of aspect-level sentiment classification is to automatically find the words associated with asp_p and predict the sentiment polarity of D_{pq} in terms of asp_p.

3.2 Attention Mechanism

Literally, an attention mechanism is viewed as the mind's ability to allocate uneven focuses on an object and bring concerning elements to the fore, while neglecting or decreasing the importance of others. Such idea of focusing on the most pertinent piece of information have been widely applied in computer vision, speech recognition, machine translation and image caption generation.

In terms of aspect-level sentiment classification, the attention mechanism is employed to allow the classifying model to concentrate on the important parts of a sentence in response to a specific aspect, and thus enhance the capability of extracting the most relevant information for determining the polarities. In most works, attention is simply a vector generated by the output of a dense layer using softmax function. For example, in (Wang et al., 2016), an attention weight vector was proposed to determine the importance of words in a sentence given an aspect, the weights of which can be further used to scale the hidden states of a LSTM model and thus obtain the weighted representation for a sentence. More specifically, in the sentence "The *price* is *reasonable*, although the *service* is *poor*", two aspects are reviewed (as italicized), i.e. the positive attitude towards price, and the negative attitude towards service. However, when it merely takes account of the price aspect, the attention mechanism should allocate larger weights to underlined words like {"price", "reasonable"}, while paying less attention to the other words.

3.3 Artificial Bee Colony Algorithm

An evolutionary algorithm runs on a population of individuals with each one represented by a candidate solution for the optimization problem. Each individual has a fitness value that can determine the quality of the corresponding candidate solution. Typically, EAs gradually improves the population towards a potential sub-space through evolution operators, such as crossover, mutation and greedy selection. Considering that the ABC algorithm has excellent exploration ability as validated in (Karaboga and Basturk, 2007), it is employed here to optimize the proposed method.

The mechanism of the ABC algorithm is inspired by the foraging behaviors existing in bees. According to the division of labors, the colony contains employed bees, onlooker bees and scouts. The number of employed and onlooker bees are equal to the population size, both of which account for half of the colony. Note that one candidate solution is assigned to only one employed bee. Assuming that the initial population, consisting of SN individuals with each e-dimensional solution $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,e})$ randomly created by Eq. (1), where $\mathrm{i} \in \{0,1,..,\mathrm{SN}-1\}$ and $\mathrm{j} \in \{0,1,\ldots \mathrm{e}-1\}$; x_j^{min} and x_j^{max} respectively denotes the lower and upper boundaries of the j_{th} dimension; and $rand$ represents a random value uniformly distributed in $(0,1)$. Each iteration involves three phases:

a) In the employed bee phase, a new candidate solution V_i is generated for X_i by Eq. (2), where $j \in \{0,1,\ldots,e-1\}$ and $k \in \{0,1,\ldots SN-1\}$ are randomly chosen indexes; $\varphi_{i,j}$ is a random value in $[-1,1]$. Then, a greedy selection based on the quality of the fitness is adopted to select the better one between V_i and X_i. After all employed bees update the solutions, they share the new position information with onlooker bees;

b) In the onlooker bee phase, the updating process is similar to that of the employed bees, and the main difference relies on that the onlookers only focus on exploiting potential solutions selected according to their probabilities calculated by fitness values. Since a classification task is a maximum problem, the fitness value of X_i (termed as f_i) is defined in Eq. (3), i.e. the ratio of the number of correct predictions of sentences to the total number of sentences, and the selection probability Pr_i of X_i is calculated by Eq. (4).

c) In the scout phase, a solution would be replaced by a newly generated one using Eq. (1) if it had not been improved after consecutive $Limit$ iterations.

$$x_{i,j} = x_j^{min} + rand * (x_j^{max} - x_j^{min}) \quad (1)$$

$$v_{i,j} = x_{i,j} + \varphi_{i,j} * (x_{i,j} - x_{k,j}) \quad (2)$$

$$f_i = \frac{The\ number\ of\ correctly\ predicted\ sentences}{Total\ number\ of\ sentences} \quad (3)$$

$$Pr_i = f_i / \sum_{i=0}^{SN-1} f_i \quad (4)$$

The Evolution of the population occurs during the repeated implementation of the above operators until certain convergence criteria on the fitness values is satisfied, or simply after a predefined number of iterations. **Algorithm1** presents the pseudocode of the ABC algorithm. After completing the evolutionary process, the global best solution recorded will be used as the optimized solution for a particular problem.

Algorithm1: The pseudocode of ABC algorithm

1: Initialize the parameters, i.e. *SN*, *d*, *Limit*;
2: Generate the initial population;
3: Evaluate the fitness values for the population;

4: **Repeat**
7: The employed bee phase:
8: **for** $i = 0: SN - 1$ **do**
12 generate a candidate solution V_i by Eq. (3);
13: **if** $f(X_i) < f(V_i)$, set $X_i = V_i$, $trial_i = 0$; **otherwise**, set $trial_i = trial_i + 1$ **end if**;
14: **end**
15: Calculate probabilities according to Eq. (5), and set $t = 0$, $i = 0$;
17: The onlooker bee phase:
18: **while** $t < SN$ **do**
19: **if** $rand < Pr_i$:
23: generate a candidate solution V_i by Eq. (3);
24: **if** $f(X_i) < f(V_i)$, set $X_i = V_i$, $trial_i = 0$; **otherwise**, set $trial_i = trial_i + 1$ **end if**;
25: $t = t + 1$; $i = i + 1$, **if** $i >= SN$, set $i = 0$ **end if**;
26: **end if**
27: **end**
28: The scout phase:
29: **if** $max(trial_i) > Limit$, replace X_i with a randomly generated solution by Eq. (2) **end if**;
30: **Until** termination condition is met.

3.4 The Architecture of the CNN Model for Sentiment Classification Problem

Considering the efficiency and simplicity of the one-layer CNN model proposed by Kim (2014), which employs pre-trained word vectors as inputs, it will be used here as the baseline model for the proposed attention-based word embeddings, with the aim to improve the performance of aspect-level sentiment classification. On the whole, the architecture of this CNN model is composed of four parts:

a) **Generating Sentence Matrix:** Beginning with a sentence tokenized with z tokens, let $wv_i \in R^d$ be the d-dimensional word vector for the i_{th} word in the sentence, then a sentence matrix, termed as SentMat $\in R^{z \times d}$, can be made by vertically concatenating the word vectors of tokens as $wv_{1:z} = wv_1 \oplus wv_2 \oplus \ldots \oplus wv_z$, where $\oplus$ is the concatenation operator.

b) **Performing Convolution Operation:** Let $\text{SentMat}_{i:i+h}$ refers to a window of words, representing the concatenation of words from wv_i to wv_{i+h-1}. Viewing the sentence matrix as an image, a convolution operation, involving a filter wf $\in R^{h \times d}$, can be used to extract a new feature from $\text{SentMat}_{i:i+h}$ by Eq. (5), where f(.) is a non-linear activation function and b $\in$ R is a bias term. This filter can be repeatedly applied to the possible windows of SentMat, i.e. $\{SentMat_{0:h}, SentMat_{1:h+1}, \ldots, SentMat_{z-h:z}\}$, to obtain a feature map $c = [c_0, c_1, \ldots, c_{z-h}]$ with c $\in R^{z-h+1}$. Multiple filters can be used for the same windows in order to learn comprehensive features. Besides, the heights of the filters, i.e. the number of words contained in the window, can also be varied to exploit more regional information.

$$c_i = f(wf \cdot SentMat_{i:i+h-1} + b) \quad (5)$$

c) **Performing Pooling Operation:** Intending to have a better generalization, a pooling operation, normally a max-over-time type, will be applied onto the feature map, which takes the maximum value $\hat{c} = [c_0, c_1, \ldots, c_{z-h}]$ to represent the salient feature for the corresponding filter. This pooling scheme extracts one feature for one filter.

d) **Performing Softmax Function:** The features extracted from various filters can be concatenated into a fixed-length feature vector, which is then input into a fully connected softmax layer, the output of which is the probability distribution over labels. In order to

avoid co-adaptation of hidden units during forward backpropagation, this softmax layer can be regularized by dropout with a L2-norm constraint on the weight vectors, where dropout randomly set feature values to 0 (Hinton et al., 2012).

Categorical cross-entropy loss is used as the training objective, and the optimization is conducted by stochastic gradient descent (SGD) and back-propagation (Rumelhart et al., 1988). Note that this architecture contains two channels, i.e. "static" and "non-static". Word embeddings are kept static throughout the training in the former channel, while they are fined-tuned by backpropagation in the latter one. Fig. 1. shows an illustration of this model for sentence classification.

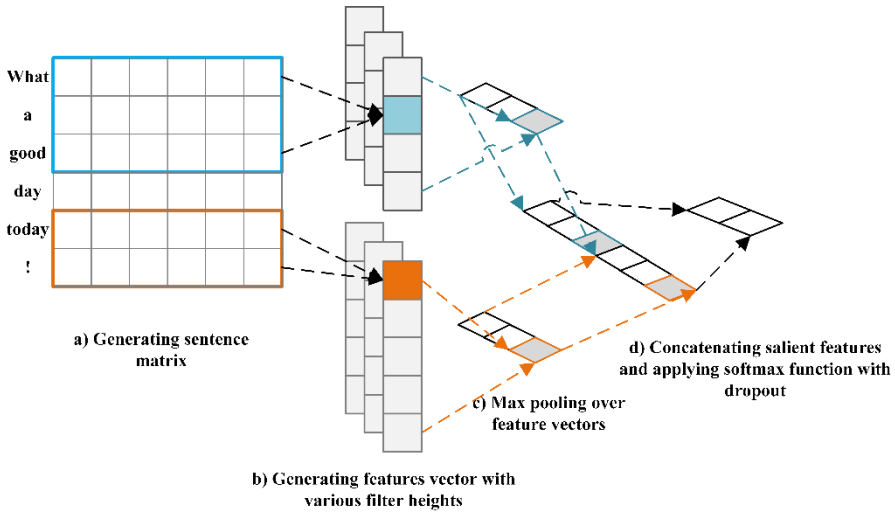


Fig. 1. CNN model architecture for sentiment classification

4 Attention-based Word Embeddings for Aspect-level Sentiment Classification

4.1 Motivations

As illustrated in the above model, the performance of deep neural-network models significantly depends on the quality of word embeddings, which have remarkable capability of capturing semantic relationships between words (Manning et al., 2015). For the sake of revealing the dependencies between aspects and context words, there is a need to comprehensively discuss the properties of word embeddings.

4.1.1 The Characteristics of Word Embeddings

Generally speaking, the superiorities of the word embedding can be summarized into two aspects (Mandelbaum & Shalev, 2016), i.e. quantifying the semantic similarity relationships and regularizing semantic and syntactic relationships as linear analogies. However, according to previous research (Schwartz et al., 2015), there also exist non-trivial deficiencies, which will be explained from the view of creating vectors for antonymous and polysemy words. Furthermore, two derivative doubts are raised and discussed, aiming to deeply grasp the essence of word vectors.

1) Quantification of the Semantic Similarity Relationships

Advantage: Since the words can be projected into a low-dimensional space, the similarity of two words can be simply calculated by the geometrical distances between them, where cosine similarity and Euclidean distance are common metrics. With an eye to the intuition of word embeddings models, it stands behind a hypothesis that the meaning of a word is determined by "the company it keeps" (Firth, 1957). For example, the predictive SGNS-300 model tries to predict a

word from its contexts. Thus, semantically similar or related words, which share a variety of similar contexts, tend to have close vectors in the word space and obtain high similarities, such as synonym pairs like (‘cute’, ‘pretty’) and (‘big’, ‘large’). In other words, similar words tend to cluster towards a specific sub-space, and words related to different themes may gather at diverse points in the semantic space. In Fig. 2, five groups of words regarding to the topics of fruits, cats, birds, clothing and musical instruments are visualized into a 2-dimensional space using a t-distributed stochastic neighbor embedding (t-SNE) technique (Maaten & Hinton, 2008)[1]. It can be seen that the words within different topics marked as particular colors can gather into a specific cluster, and the regional boundaries among different clusters are explicit.

Disadvantage: Under the hypothesis that the meaning of a word is depicted by its contexts, an issue similar to synonyms cannot be avoided; antonyms are also likely to co-occur in the same contexts or fixed patterns, and may achieve high similarity degrees. For instances, the similarity score between (“good”, “bad”) is 0.7190, whereas they have opposite meanings by definition. As examples, the cosine similarity scores of similar and dissimilar pairs are given in Table 1. Comparing with the similar words, the high scores of dissimilar words demonstrate that the contextual word embeddings are incapable of distinguishing opposite words.

Discussion: It needs to clarify that whether the antonyms and synonyms can be clustered into the same sub-space or not, as a simple example, it should make clear that whether the distance between antonyms like (“good”, “bad”) is smaller than that between irrelevant pairs like (“good”, “apple”). For clearness, three sets of words, with the centers being {delightful (Adjective), strictness (Noun), quickly (Adverb)} and the relevant antonyms and synonyms obtained from the online Thesaurus dictionary[2] are visualized in Fig. 3, by the t-SNE with the SGNS-300 word embeddings. Each center word is connected with the corresponding synonyms and antonyms by grey and orange dashed lines, respectively. Apparently, the margins between different groups are large, and the antonyms and synonyms can scatter around an individual center word, forming a relatively isolated cluster. Unfortunately, it is difficult to differentiate the distribution of antonyms from that of the synonyms.

Similar pair	Similarity score	Dissimilar pair	Similarity score
(“good”, “great”)	0.7292	(“good”, “bad”)	0.7190
(“accept”, “acquiesce”)	0.5666	(“accept”, “reject”)	0.6692
(“wide”, “broad”)	0.4729	(“wide”, “narrow”)	0.4576
(“agree”, “concur”)	0.7132	(“agree”, “disagree”)	0.7712
(“forget”, “overlook”)	0.5474	(“forget”, “remember”)	0.7296
(“argument”, “quarrel”)	0.5334	(“argument”, “reasoning”)	0.5321
(“large”, “sizeable”)	0.7341	(“large”, “small”)	0.7331
(“much”, “lot”)	0.5574	(“much”, “little”)	0.6419
(“many”, “numerous”)	0.6569	(“many”, “few”)	0.6052
(“fast”, “quickly”)	0.5394	(“fast”, “slow”)	0.5314

Table 1. Similarity scores for similar and dissimilar words

[1] The word embeddings used in this section are pretrained 300-dimesion word vectors trained on the SGNS model (named as SGNS-300), where the Google News corpus with 6 billion running tokens was used as training source, https://radimrehurek.com/gensim/models/word2vec.html.

[2] https://www.thesaurus.com/

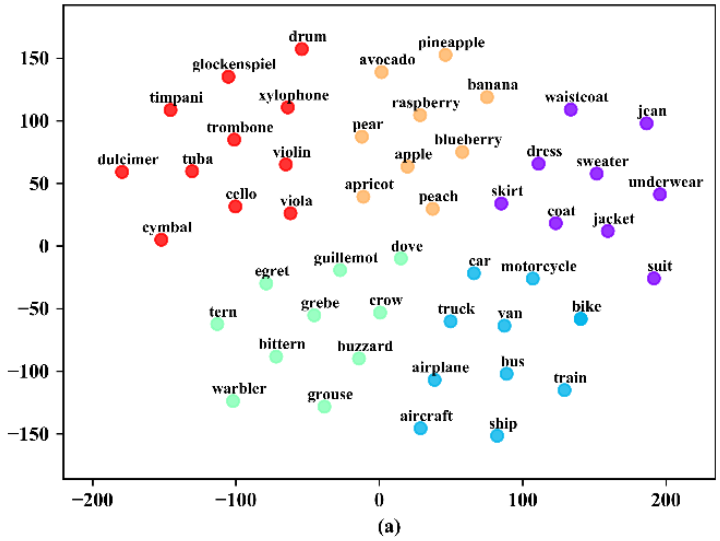


Fig. 2. t-SNE visualization of the five groups of words using SGNS-300 embeddings

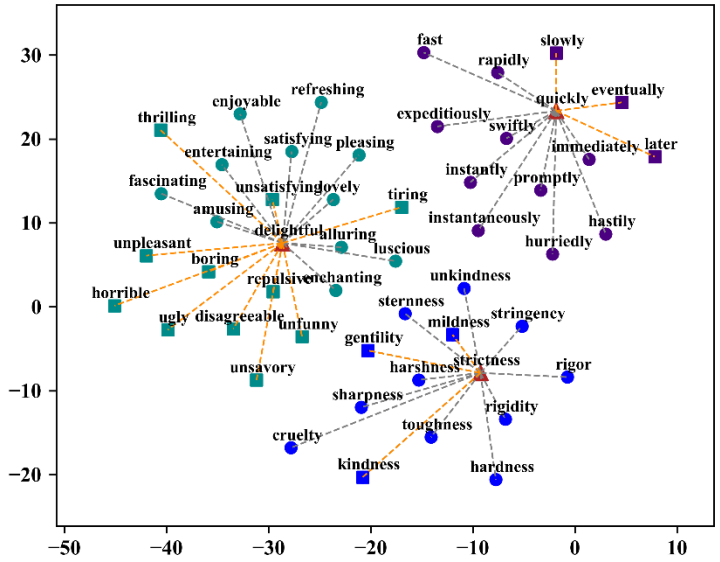


Fig. 3. The distributions of the antonyms and synonyms for given words

2) Regularization of the Semantic and Syntactic Relationships as Linear Analogies

Advantage: Mikolov et al. (2013a; 2013b) claimed that certain meaningful syntactic regularities between words could be revealed by word embeddings via linear algebra. Being more specific, simple analogical reasoning operations could be performed on the word vectors in the form of $wv_{a1} - wv_{b1} = wv_{a2} - wv_{b2}$, where a1, a2, b1 and b2 are four different words. This can be explained in a way that the closest word representation to the resulting vector of $wv_{king} - wv_{man} + wv_{woman}$ is wv_{queen}. Consequently, various types of sophisticated relations can be intrinsically encoded, and several relationship exemplars are depicted in Fig. 4, including the common Capital-Country, Adjective-Comparative, Noun-Plural, and Adjective-Adverb analogies.

Disadvantage: Although the word embeddings can regularize certain relationships as linear analogies for words, they show indiscriminative for homonymy and polysemy words, owing to the

fact that the diverse senses of a word are compressed into only one single vector (Pelevina et al., 2016). Especially for sentiment classification, the single-prototype representation for a word with multiple senses makes it insensitive or even misleading for interpreting the appropriate meaning in a particular context. For example, the word "*sweet*" indicates a pleasant taste characteristic of sugar or honey under the topic of food, while it can also indicate a delightful mood under a topic concerning emotions.

Discussion: It needs to clarify that whether the word embeddings can embed the most comprehensive and generalized senses for a given word or not, namely verifying their generalization ability. With this puzzle, a simple experiment has been conducted, which calculates the similarity scores between a given word and three sets of words that are highly associated to it, including synonyms, hypernyms and hyponyms, and then measure the statistical information, such as the maximum, mean, median and minimum similarity scores for each set. The underlying idea lies in that the word embeddings is supposed to have a good generalization ability, if it shows insignificant differences among the statistical values obtained from the various types of relevant words. Herein, 266 distinct words included in the SimLex-999 dataset (Hill et al., 2015) are used for this experiment, with each word having at least five related words obtained from the WordNet lexical database[3] for each set. The statistical results are shown in Fig. 5. In view of the maximum values, the synonyms have highest values, up to nearly 1.0, nevertheless, the values for hypernyms and hyponyms are both above 0.40, which are not trivial. For the mean, median and minimum scores, the divergences among the three sets are at an insignificant level. As a result, it indicates that the word senses can be roughly generalized by word embeddings to some extent.

4.1.2 Inspirations from the Characteristics of the Word Embeddings

After comprehensively analyzing the characteristics of word embeddings, it can be concluded that the word vectors could be automatically well-organized in the semantic space on the basis of their concepts and senses. In this case, a hypothesis can be naturally put forward as follows:

Hypothesis: *Each sense of the words can be embedded into one or more particular dimensions in the vector space, in other words, each dimension can entail a specific semantic meaning.*

In addition, two extended assumptions can be derived:

Assumption1: *Words with multiple senses (i.e., polysemy) put more emphasis on certain related dimensions or spatial directions, in order to obtain unambiguous distinguishes from other words. For example, "sweet" could have high values in the specific dimensions regarding to both taste and emotion, while losing focuses on others.*

Assumption2[4]*: Antonyms are similar in every dimension of meaning except one referring to a particular sense where they deviate from. For example, considering the aspect of temperature, the difference between "hot" and "cold" only relies on the "heat" dimension, where "hot" has a higher value than "cold".*

Under these assumptions, the difficulties of aspect-level sentiment classification can be considerably alleviated, since it can be briefly simplified into finding the relevant dimensions of the word embeddings for favorable aspects and sentiments in the semantic space.

[3] https://wordnet.princeton.edu/

[4] Note that the ***Assumption2*** is akin to the *paradox of simultaneous similarity and difference between the antonyms* (Cruse, 1986), which claimed that antonyms and synonyms are similar in every dimension of meaning except a particular one.

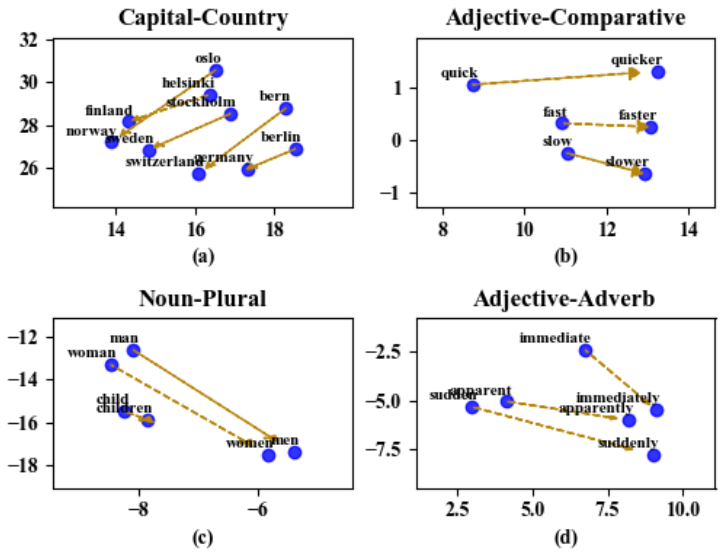


Fig. 4. Linear analogies between word pairs

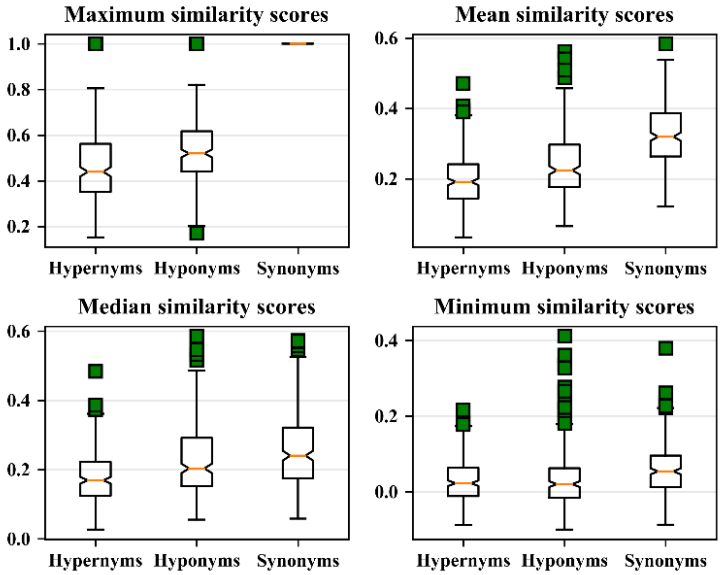


Fig. 5. Statistical distribution of the similarity scores for hypernyms, hyponyms and synonyms

4.2 The Proposed Attention-based Word Embeddings

Based on the aforementioned discussions, aspect-level sentiment classification has been streamlined to concentrate on salient dimensions that highly correlate with the task. Thus, it can take advantages of the attention mechanism to determine the importance of different dimensions based on their contributions to the correct prediction.

4.2.1 Attention Vector Designed for the Word Embeddings

Due to the disadvantages of word embeddings, this work designs an attention vector for each aspect in a task, termed as *ATV*, which comprises two sub-vectors. One is called dimension attention (*DATV*), the goal of which is to measure the importance of dimensions and decrease the noises caused by the multiple senses of polysemy; the other one is called sentiment attention (*SATV*),

aiming to reduce the obscure existing in antonyms for the *DATV*, using the integration of sentiment lexicons and PoS tagging. For the p_{th} aspect asp_p with respect to the sub-dataset D_p, the corresponding attention vector is denoted as $ATV_p = DATV_p \oplus SATV_p$. Let $S \in R^d$ be the word space. The sub-vector for asp_p is illustrated as below:

1) Dimension Attention

$DATV_p$ has the same dimensionality with the word space, i.e. $DATV_p \in R^d$, which scales the significance of each dimension using a weight varying in $[0,1]$. Basically, there is a positive correlation between the significance of a dimension and its corresponding weight, namely, a dimension that is tightly coupled with aspect asp_p could have a large weight. Specifically, the word vector wv_i can be refined by element-wise multiplications with $DATV_p$, resulting in a weighted vector $wd_{ip} = DATV_p * wv_i$, where wd_{ip} denotes the scaled vector of wv_i with $DATV_p$. With d set as 10, the operation of the $DATV_p$ applied on wv_i is shown in Fig. 6, where $D_{j\in\{0,1,\ldots,d-1\}}$ represents the j_{th} dimension, and the red rectangle shows the operation conducted on D_7.

As seen from the heat map of each vector, $DATV_p$ is concerned with partial dimensions ranging from D_5 to D_8, while wv_i pays attention to two parts, including $\{D_0, D_1, D_2\}$ and $\{D_5, D_6, D_7, D_8\}$. Nevertheless, after the attention effect of $DATV_p$, the vision of wv_i will be diverted from $\{D_0, D_1, D_2\}$ and focus on only $\{D_5, D_6, D_7, D_8\}$ as shown in wd_{ip}. In short, $DATV_p$ can fundamentally prevent the distraction of wv_i and switch the attention to meaningful directions for the aspect asp_p. Most importantly, it has advantages of selecting an appropriate sense for a polysemy, so as to avoid making misunderstandings for classification.

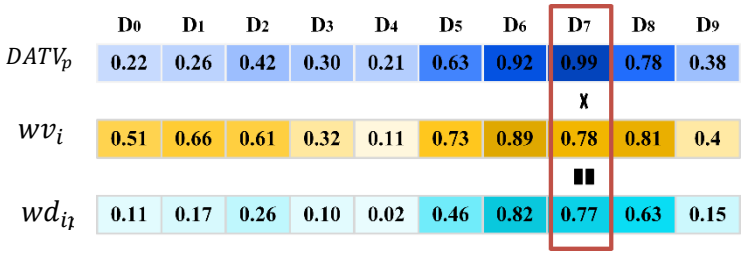


Fig. 6. The operation of the dimension attention applied on a word vector

2) Sentiment Attention

Considering the discussions under the Assumption1, a problem has aroused that the sentiments of antonyms cannot be finely distinguished by dimensions. In order to get the words' polarities, sentiment lexicons are leveraged to allocate sentiment scores to individual words. Moreover, as claimed in (Asghar et al., 2014), PoS tagging is beneficial for explicit feature extraction as proven by the good performance achieved in (Wang, 2017). Therefore, the sentiment lexicons and PoS tagging are integrated to determine the attributes of words.

Concretely speaking, the main types of PoS tags adopted here include nouns, verbs, adjectives and adverbs, as they can deliver crucial implications of opinions or sentiments. Taking into account of the special roles of indicating the emotions, punctuations, like "?" and "!", and conjunctions (e.g. "so", "or" and "but") are also involved. The PoS tags are simplified into coarse-grained categories for the sake of reducing computational complexity. For example, the PoS tags related to various nouns, such as plural nouns (NNS), proper nouns (NNP) and proper/plural nouns (NNPS), are equally treated and abbreviated as "N". Table 2 shows six distinct PoS categories used in this paper, while those unconsidered ones are marked as "UNK".

Moreover, the sentiment types extracted from lexicons contain "positive" (Pst), "negative" (Neg), "neural" (Neu), "intensifier" (Int), "negation" (No), "transition" (Trs) and "none" (Non),

where “none” represents that the current word cannot be found in all lexicons used. The priorities are ordered as No > Trs> Int > Pst = Neg = Neu > Non. These polarities are automatically marked according to the rating valences given by the lexicons. That is to say, a particular word would be annotated as “positive” if its valence is greater than zero, otherwise it would be labelled as “negative” or “neural” when the valence is less than or equal to zero. As declared in (Kiritchenko & Mohammad, 2016), negators, modals, and degree adverbs can dramatically affect the sentiments of words or phrases they modify. Hereby, negations (e.g., “not”, “yet”, and “no”) are implemented to avoid misclassification, considering that a negation key appearing around a word may alter the sentiment into the opposite side; degree adverbs (e.g. “more”, “very”, and “much”) are leveraged as “intensifier” to precisely detect the sentiment intensities of words; and transition words (e.g., “but” and “whereas”) are used to detect the changes in sentiments.

The attribute of a word can be determined by its sentiment polarity and PoS tag. More specifically, a word would be labelled as “N_Pst” when it is viewed as a noun with a positive polarity in a given sentence. Theoretically, each PoS category could have seven or less sentiment polarities. Define the number of distinct PoS categories as np, the maximum number of distinct attributes, termed as ns, is equal to $np * 7$. On this basis, the $SATV_p$ is an ordered vector, the element of which is a weight that can represent the importance of a particular word attribute. Apparently, the ranges of weights defined for the positive and non-positive attributes should be different. Herein, the weights of the “negative” and “negation” attributes vary in [-1, 1], while that of other attributes range from 0 to 1. The depiction of $SATV_p$ is presented in Fig. 7, where $S_{i\in\{0,1,\dots,ns-1\}}$ means the i_{th} component; $Bound_{lower}$ and $Bound_{upper}$ denote the lower and upper boundaries, respectively. The contents in the red rectangular gives an example of S_6, which represents the “A_Pst” attribute fluctuating in [0,1] with the weight currently tuned as 0.85.

PoS category	PoS tags	Abbreviation
Nouns	NN, NNS, NNP, NNPS	“N”
Adjectives	Adjective (JJ); Comparative Adjective (JJR), Superlative Adjective (JJS)	“A”
Verbs	Verb (VB), Past-Tense Verb (VBD), Gerund/Present Participle (VBG), Past-Participle Verb (VBN), Non-3rd Person/Single/Present Verb (VBP), 3rd Person/Single/Present Verb (VBZ)	“V”
Adverbs	Adverb (RB), Comparative Adverb (RBR), Superlative Adverb (RBS)	“R”
Punctuations	“?”, “!”	“P”
Conjunctions	Coordinating Conjunction (CC), Preposition/Subordinating Conjunction (IN)	“C”
Others		“UNK”

Table 2. PoS categories and their abbreviations

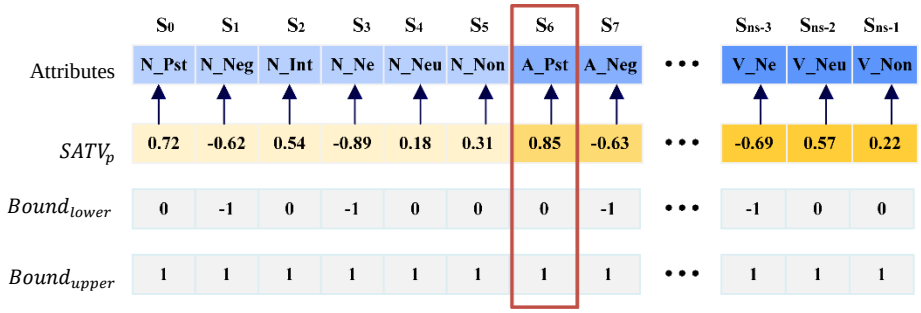
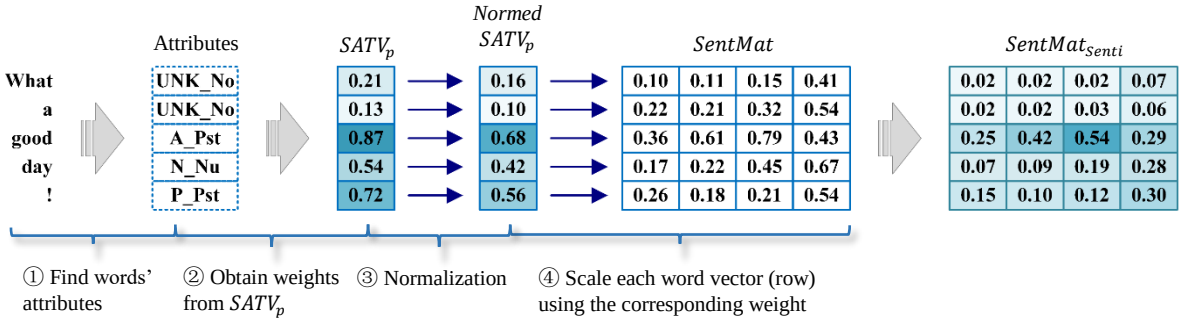
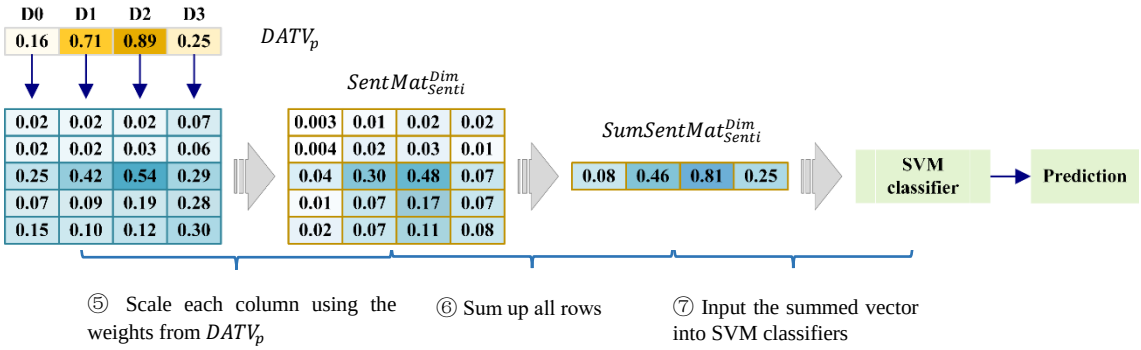


Fig. 7. A depiction of the sentiment attention sub-vector

3) Sentence Representation Using the Attention Vector



(a) The scaling process of the sentence matrix with the $SATV_p$



(b) The scaling process of the sentence matrix with the $DATV_p$

Fig. 8. The vectorization process of a sentence using the $SATV_p$ and $DATV_p$

As mentioned above, the ATV_p is the concatenation of the $SATV_p$ and $DATV_p$, with the total length being $nd = ns + d$, i.e. $ATV_p \in R^{nd}$. ATV_p can modify the sentence matrix $SentMat$, which can be further used as the inputs of the SVM classifier. Assuming that a sentence is "What a good day!" (five tokens contained), the scaling process of $SentMat$ using ATV_p is shown in Fig. 8. The steps involved are summarized as follows:

a) Obtaining the instance matrix SentMat $\in R^{5\times d}$ through concatenating the word vectors along rows;

b) Scaling each row of the $SentMat$ matrix with the normalized weights obtained from $SATV_p$ and thus obtaining the $SentMax_{Senti} \in R^{5\times d}$ matrix;

c) Scaling each column of the $SentMax_{Senti}$ matrix with the weights obtained from the $DATV_p$ and thus obtaining the $SentMat_{Senti}^{Dim} \in R^{5\times d}$ matrix;

d) Summing up all rows in the $SentMat_{Senti}^{Dim}$ matrix and then obtaining the $SumSentMat_{Senti}^{Dim} \in R^{d}$ vector;

e) Inputting the $SumSentMat_{Senti}^{Dim}$ vector into the SVM classifier for training.

4.2.2 Optimization Approach for the Attention Vector

Intending to achieve the fine-turned weights for the sentiment attributes and space dimensions, the attention vector should be fully optimized to select the most relevant sub-space for an aspect, with details described as below.

1) Optimization Objective

For an aspect-level sentiment classification task, the goal is to find the most suitable attention vector that can maximize the classification accuracy regarding to a specific aspect in a given sub-dataset. Therefore, the optimization objective is formulated as in Eq. (6), where ATV_p^{best} represents the best attention vector optimized for asp_p, where $ATV_p^{best} = DATV_p^{best} \oplus SATV_p^{best}$.

$$ATV_p^{best} = arg \max_{ATV_p} Accuracy(D_p, ATV_p) \tag{6}$$

2) Classification Target

The traditional SVM with a linear kernel is employed for sentiment classification. For the aspect asp_p, the corresponding sub-dataset D_P should be separated into a training set and a testing set, and each sentence contained should be vectorized in accordance with the steps shown in Fig. 8, using ATV_p. After that, the SVM classifier will be trained on the training set with 5-fold cross validation and the average accuracy of all folds is regarded as the training performance of the current solution for the ATV_p, as shown in Fig. 9. Besides, the performance of the trained SVM classifier can be further verified on the testing set.

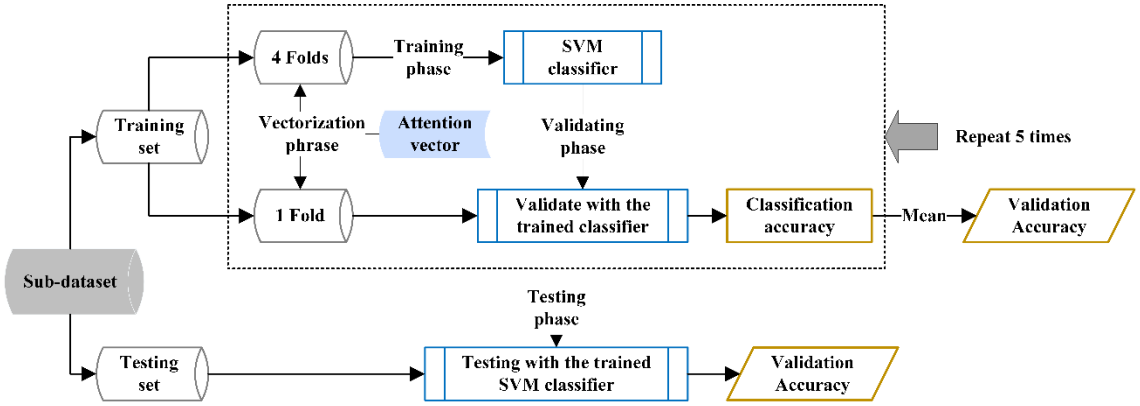


Fig. 9. The training process with the SVM for a specific aspect

3) Evolutionary Optimization

ABC variants are employed to provide approximate solutions for ATV_p, with the SVM classifier accuracy used as the fitness function. Since it aims to search an attention vector with maximum accuracy for a given aspect, each individual in the population is represented by a candidate solution for the ATV_p and the classification accuracy generated by the SVM classifier as shown in Fig. 9 is regarded as the corresponding fitness value. For clarity, the evolutionary framework of the ABC algorithm is presented in Fig. 10. Note that the ABC algorithm should be repeatedly performed for multiple times, thus the average vector of all global best solutions obtained from each run will be used as the final best solution for the ATV_p^{best}.

4.3 The Optimized Attention Vector Applied on the CNN model

It can be seen that the approach proposed in this paper pays attention to solving the aspect-level sentiment classification problem from the view of discovering the salient dimensions for a specific aspect in the semantic space, which greatly differs from the conventional models that aim to design complicated deep neural-networks.

The CNN model with a "non-static" channel is employed here. After obtaining the best attention vectors for all aspects, each sentence in the whole dataset D could be transformed into a

scaled sentence matrix using the corresponding best $DATV$ regarding to a concerned aspect. It is worth noting that only $DATV$ is used to scale the sentence matrix, since this model can simultaneously adjust the word embeddings during the training process. The scaling process is depicted in Fig. 11, where $SentMat_{Dim}$ represents the scaled sentence matrix by the $DATV_p^{best}$. Due to the fact that each sentence may have several aspects, the best $DATV$ used to modify the sentence matrix depends on the target aspect. For example, in sentence "Not only was the food outstanding, but the little 'perks' were great.", two aspects are involved, i.e. {"food": positive; "service": positive}. When the target aspect is "*food*", then food-related $DATV$ will be used to scale the sentence, or the service-related $DATV$ will be applied if the aspect concerned is "service".

After completing the vectorizing and scaling process of the dataset, the CNN model will be trained on the training set and its general performance will be validated on the testing set, as shown in Fig. 12.

Literally, the optimized $DATV$ can be used to improve the performance of various state-of-the-art models, such as RNN variants and other CNN variants by simply scaling the word embeddings, which is simple and applicable. For simplicity, this paper only conducts experiments on the one-layer CNN model.

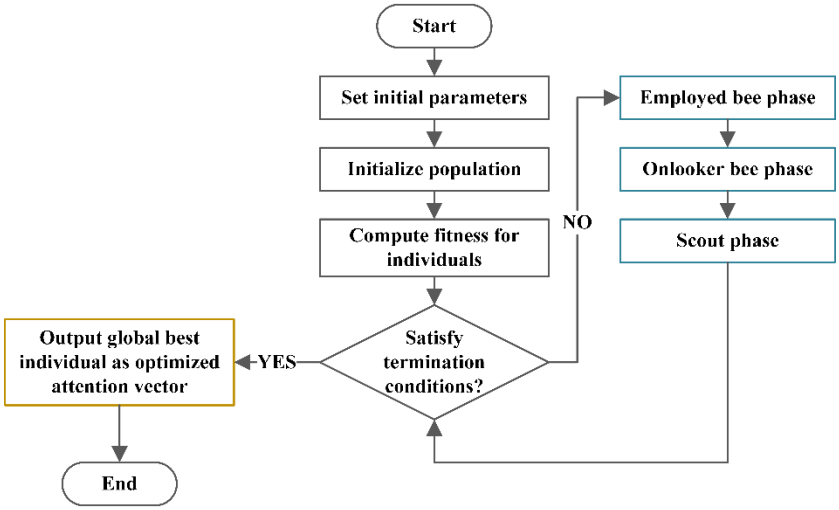


Fig. 10. Evolutionary framework of the ABC algorithm for optimizing the attention vector

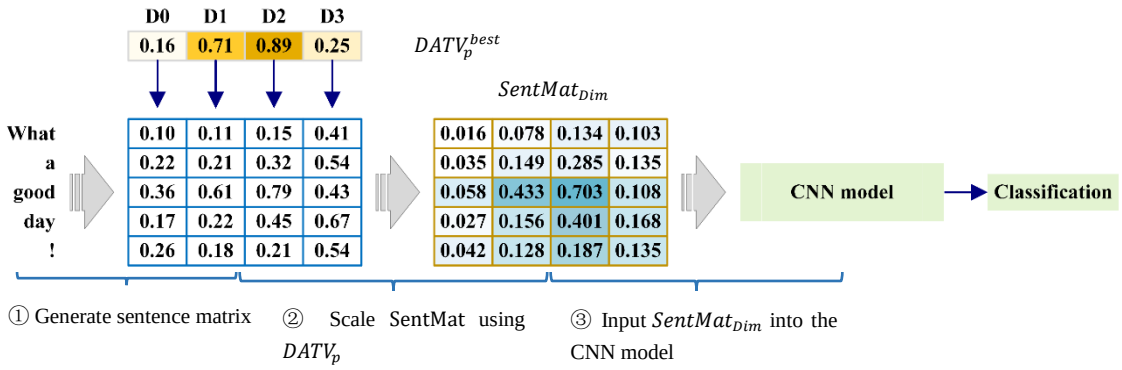


Fig. 11. The optimized dimension attention $DATV_p^{best}$ applied on the CNN model

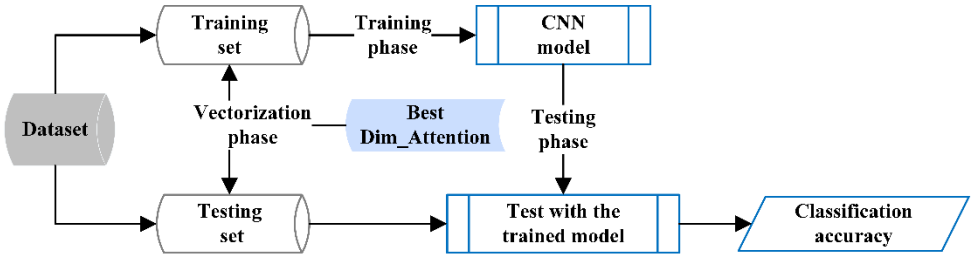


Fig. 12. The training and testing phases with the best dimension attention sub-vector on the CNN model

5 Experiments and Analysis

5.1 Test Suites

5.1.1 Evolutionary Algorithms

In order to thoroughly validate the efficiency of the proposed approach, five ABC variants are employed for experiments as presented in Table 3, including Gbest-guided ABC (GABC), Global Best ABC (ABCbest1), Enhancing ABC (EABC) and Cellular ABC with Gaussian distribution (CGABC). For fairness, the parameter settings and tuning methods of these algorithms comply with the originally relative papers as presented in Table 3. The maximum number of iterations for all algorithms is set as 1200 and the population size is 50. Each algorithm runs 30 times.

Algorithm	Abbreviation	Typical hyper-parameters setting	Reference
ABC	ABC	Limit = 200	(Karaboga & Basturk, 2007)
Gbest-guided ABC	GABC	c = 1.5, Limit = 1.0 ∗ SN ∗ D	(Zhu & Kwong, 2010)
Global Best ABC	ABCbest1	Limit = 0.6 ∗ SN ∗ D	(Gao et al., 2012)
Enhancing ABC	EABC	A = 1; μ = σ = 0.3; Limit = 200	(Gao et al., 2014)
Cellular ABC with Gaussian distribution	CGABC	Cellular topology: C25; 2D grid shape: 5 × 10; Limit = 200	(Zhang et al., 2018)

Table 3. Parameter settings for all evolutionary algorithms used in the comparisons.

5.1.2 Evaluation Datasets

The experiments are conducted on restaurant reviews in the SemEval 2014 Task 4[5] (Pontiki et al., 2014). The reviews have been labeled with four sentiment polarities, i.e. positive, neutral, negative and conflict, while only positive and negative polarities are concerned in this paper. This dataset provides the sentiment polarity and location for each aspect term occurring in a sentence. For example, in the sentence "The staff was so horrible to us", aspect term "staff" is highlighted as negative with the position being 2. The aspect terms are generalized into five aspects, including "service", "food", "price", "anecdotes/miscellaneous" (termed as anecdotes) and "ambience". As in the sentence, the aspect related to "staff" is "service" with the polarity marked as negative. Statistics is given in Table 4, where No. Positive and No. Negative denote the corresponding number of positive and negative reviews included.

[5] http://alt.qcri.org/semeval2014/

5.1.3 Sentiment Lexicons

Nine polarized lexicon datasets associated with both word emotion and sentiment are employed for the experiments, which include AFINN-165 Lexicon (Nielsen, 2011), Bing Liu's opinion lexicon (Bing-Liu) (Hu & Liu, 2004), sentiment words from www.enchantedlearning.com6 (Enchanted-Learning), MPAA (Wilson et al., 2005), Macquarie semantic orientation lexicon (MSOL) (Mohammad et al., 2009), NRC emotion (NRC-Emotion) lexicon (Mohammad et al., 2013; Kiritchenko et al., 2014), past and future plus (PF+) lexicon (Augustyniak et al., 2014), sentiment composition lexicon for negators, modals, and degree adverbs (SCL-NMA) (Kiritchenko & Mohammad, 2016), SentiWordNet 3.0 lexicon (Baccianella et al., 2010). Then, the words occurred in all lexicons should be merged into one composite lexicon, termed as Cmpst-Lex. Due to the fact that a word appearing in multiple lexicons may have contradictory polarities, a strict selection process is adopted here, where only those words that have constant polarities in all related lexicons could be retained. Table 5. lists the number of positive, negative and neutral words contained in each lexicon, represented as No. Positive, No. Negative and No. Neutral, respectively.

Besides, the exemplary content words of the three auxiliary sets, namely the negation set, the intensifier set and the transition set, are presented in Table 6. Note that two punctuations "?" and "!" are included in the negator set and the intensifier set, respectively, since they can indirectly reflect the attitudes of the users. The words given in the negation set are adjustable according to the tokenization method used.

Datasets	Aspects	Training set		Testing set	
		No. Positive	No. Negative	No. Positive	No. Negative
Restaurant	**Service**	324	218	6	2
	Food	867	209	31	9
	Price	179	115	9	2
	Anecdotes	546	199	32	6
	Ambience	263	98	5	2

Table 4. The number of reviews for different aspects in two datasets

Auxiliary set name	No. Positive	No. Negative	No. Neutral
PF+	12	13	0
AFINN-165	1176	2204	2
Bing-Liu	2003	4782	0
Enchanted-Learning	266	224	0
MPAA	2304	4148	0
MSOL	30413	45930	0
NRC-Emotion	2312	3243	0
SCL-NMA	1607	1575	25
SentiWordNet 3.0	17332	19844	110130
Cmpst-Lex	29623	42762	88811

Table 5. The number of positive, negative and neutral words contained in each lexicon

5.1.4 Training Settings

All experiments regarding to the SVM classification tasks are conducted on the support vector classifier (SVC) class embedded in an SVM library provided by a public toolkit called Scikit-Learn7. Besides, a linear kernel is employed and the default value for the penalty parameter C of

[6] https://www.enchantedlearning.com/wordlist/

[7] https://scikit-learn.org/stable/

the error term is set as 0.1. For the parameter setting of the CNN model, the number of epochs is 20; filter sizes involved are 3, 4 and 5, with 100 filters for each size; L2 norm constraint is fixed to 3.0; drop rate and batch size are 0.5 and 50, respectively; the number pf epochs is set to be 10; and AdaGrad (Duchi et al., 2011) is used as optimization method, with learning rate exponentially decreasing from 0.01 to 0.001 and decay coefficient set as 2.5.

Lexicon name	Exemplary content words
Intensifier set	certainly, especially, extremely, fairly, highly, increasingly, less, more, most, much, particularly, pretty, probably, quite, rather, really, relatively, so, too, very, !
Negation set	ain't, aint, can't, n't, cant, couldn't, couldn't, didn't, doesn't, don't, don't, hasn't, haven't, hasn't, haven't, never, no, not, nothing, won't, wont, wouldn't, wouldn't, cannot, ?
Transition set	but, contrast, while, however, yet, whereas, though, conversely, still, nevertheless, admittedly, nonetheless, despite, notwithstanding, albeit, although, spite, regardless

Table 6. Lists of exemplary content words in the intensifier, negation and transition sets

5.2 Feasibility Analysis of the Attention-based Word Embeddings

This group of experiments intends to investigate the feasibility of ATV, that is to say that whether it is capable of tuning the word embeddings into a desirable state for a specific aspect or not. Herein, the ABC algorithm combined with the GloVe-50 embeddings is employed, and statistical results on the five aspects are listed in Table 7, where "Best", "Mean", "Min" and "Std" denote the maximum, average, minimum and standard variance of the classification accuracies obtained from 30 runs. For clear comparisons, the results of traditional methods using SVM are also presented, where "Mean_Vector" represents that the input of SVM is the average vector of all word vectors contained in the sentence matrix SentMat, while "Tf-idf_Vector" means that the corresponding word vectors are first scaled by the corresponding Tf-idf values and then summed up before used as the input. Despite using the word embeddings, the SVM classifier is also trained with the BoW models, where the sentences are tokenized as unigrams and then converted into count-based ("Count_Unigram") or tf-idf-based ("Tf-idf_Unigram") sparse vectors.

Method	Statistic	Service	Food	Price	Anecdotes	Ambience
Traditional	**Mean_Vector**	0.766	0.806	0.622	0.732	0.729
	Tf-idf_Vector	0.769	0.829	0.714	0.771	0.792
	Count_Unigram	0.731	0.838	0.724	0.762	0.767
	Tf-idf_Unigram	0.598	0.806	0.609	0.732	0.729
ATV	**Best**	**0.927**	**0.888**	**0.864**	**0.818**	**0.932**
	Mean	0.888	0.876	0.823	0.791	0.905
	Min	0.872	0.865	0.797	0.770	0.890
	Std	0.010	0.007	0.017	0.012	0.009

Table 7. Experimental results on the training datasets using various models

From Table 7, it can be seen that compared with the traditional methods, the classification performance can be highly improved through tuning the word embeddings using the ATV, which validates the above hypothesis that the senses of words have been embedded into certain dimensions in the semantic space. Besides, the convergence curves of the ABC algorithm on the five training datasets are plotted in Fig. 13, including the mean curves of both the global best accuracies and the average accuracies of the whole population obtained from 30 runs. Obviously, the global best accuracy can be gradually improved along with the iterations, indicating that the ABC algorithm has the ability of training the ATV. Surprisingly, the average population performances on three sub-datasets, i.e. the "service", "price" and "ambience" aspects, are slightly

decreased after certain iterations (i.e. 200), while the performances on the "food" and "anecdotes" aspects are nearly stable throughout the evolutionary process. This can be attributed to two reasons: 1) Since the classification problem is quite difficult, the *Limit* parameter of the ABC algorithm is set to be a small value (i.e. 200), in order to quickly replace the scout bees stuck in the local optima with randomly generated individuals. This means that the population may be refreshed frequently, which may not enable a stable performance of the whole population; 2) The sub-datasets are relatively small, especially for the "service", "price" and "ambience" aspects, which may result in a lack of information for the classification.

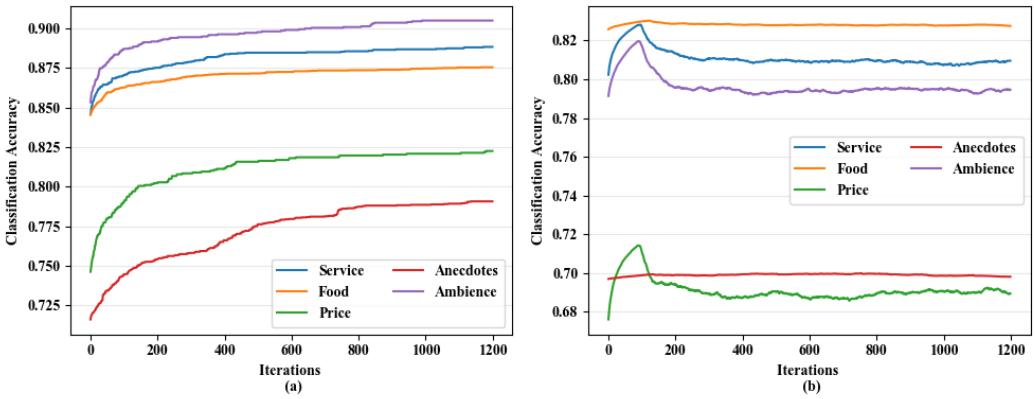


Fig. 13. Convergence curves of the ABC algorithm on the training datasets using the proposed *ATV* based model: (a) Convergence curves of the global best solutions; (b) Convergence curves of the population.

5.3 Optimization Analysis of the *ATV* Vector

1) Optimization Analysis: Different ABC Variants

The performance comparisons among the ABC, ABCbest1, GABC, EABC and CGABC algorithms are given in Table 8 with the GloVe-50 embeddings used. It is noticeable that using different ABC variants can have a huge impact on the optimization of *ATV*. The standard ABC algorithm performs comparatively poor when compared with other algorithms, while the CGABC algorithm has the best performance almost in all cases, except the "Std" value on the "ambience" aspect, where the EABC algorithm has the lowest value. The convergence curves in Fig. 14 shows that the CGABC algorithm converges fastest, exhibiting overwhelming superiorities over the other algorithms, while the convergence speed of the ABC algorithm is quite slow. It can be concluded that it is essential to design an effective ABC variant in order to train suitable attention vectors for the aspect-level sentiment classification tasks.

2) Optimization Analysis: Different Word Embedding Models

The experimental results of the SMV classifier on the training sub-datasets with various dimensionalities are given in Table 9. In respect of the "Best" and "Min" results, the proposed method performs best when the GloVe-200 word embeddings are used, which shows comparable performance with the GloVe-100 with regard to the "Min" results. However, although the classification abilities of the GloVe-50 and SGNS-300 have less advantages, they exhibit strongest stabilities, i.e. having minimum "Std" results. Surprisingly, word embeddings with large dimensionalities, such as the GloVe-300 and SGNS-300, do not show superiorities over those with smaller ones during the optimization process of the *ATV* vector.

The convergence curves are presented in Fig. 15. The GloVe-100 has the best convergence rate on the "service" aspect and shows competitive performances with the GloVe-200 on the other aspects. The SGNS-300 shows the worst convergence speed in most cases, especially on the "food", "anecdotes" and "ambience" aspects, where the sub-datasets are relatively larger than others.

Statistic	Algorithm	Service	Food	Price	Anecdotes	Ambience
Best	**ABC**	0.927	0.888	0.864	0.818	0.932
	ABCbest1	0.917	0.893	0.881	0.838	0.945
	GABC	0.917	0.893	0.881	0.831	0.945
	EABC	0.927	0.902	0.898	0.831	0.932
	CGABC	**0.936**	**0.912**	**0.915**	**0.851**	**0.959**
Mean	**ABC**	0.888	0.876	0.823	0.791	0.905
	ABCbest1	0.899	0.888	0.848	0.809	0.918
	GABC	0.899	0.885	0.845	0.807	0.916
	EABC	0.903	0.888	0.846	0.808	0.919
	CGABC	**0.920**	**0.903**	**0.893**	**0.835**	**0.944**
Min	**ABC**	0.872	0.865	0.797	0.770	0.890
	ABCbest1	0.890	0.874	0.814	0.797	0.904
	GABC	0.881	0.874	0.814	0.791	0.904
	EABC	0.890	0.874	0.814	0.791	0.904
	CGABC	**0.908**	**0.893**	**0.864**	**0.818**	**0.918**
Std	**ABC**	0.010	0.007	0.017	0.012	0.009
	ABCbest1	0.008	0.005	0.016	0.009	0.010
	GABC	0.009	***0.004***	0.016	0.008	0.011
	EABC	0.010	0.006	0.018	0.010	**0.007**
	CGABC	**0.007**	**0.004**	**0.013**	**0.006**	0.011

Table 8. Performance comparisons among the ABC, ABCbest1, GABC, EABC and CGABC

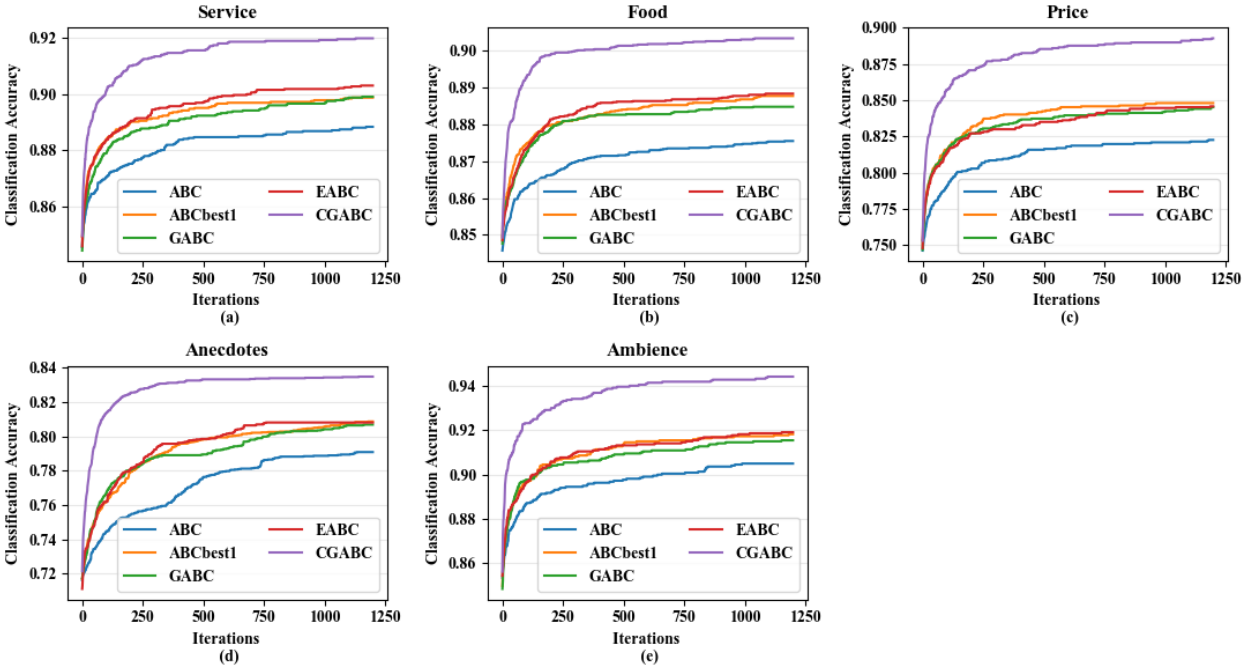


Fig. 14. Convergence curves of various ABC algorithms on the training datasets.

Statistics	Dimensions	Service	Food	Price	Anecdotes	Ambience
Best	**GloVe-50**	0.936	0.912	0.915	0.851	**0.959**
	GloVe-100	**0.954**	0.930	**0.949**	0.858	**0.959**
	GloVe-200	**0.954**	**0.935**	**0.949**	**0.865**	**0.959**
	GloVe-300	**0.954**	0.921	0.915	**0.865**	0.945
	SGNS-300	0.936	0.902	0.915	0.770	0.932
Mean	**GloVe-50**	0.920	0.903	0.893	0.835	0.944
	GloVe-100	**0.934**	**0.920**	0.908	0.847	**0.946**
	GloVe-200	0.926	**0.920**	**0.909**	**0.855**	0.942
	GloVe-300	0.929	0.913	0.897	0.848	0.937
	SGNS-300	0.919	0.893	0.886	0.755	0.911
Min	**GloVe-50**	0.908	0.893	0.864	0.818	0.918
	GloVe-100	**0.917**	0.907	**0.881**	0.831	**0.932**
	GloVe-200	0.908	**0.912**	**0.881**	**0.838**	**0.932**
	GloVe-300	**0.917**	0.902	0.847	**0.838**	0.918
	SGNS-300	0.908	0.879	0.864	0.743	0.890
Std	**GloVe-50**	**0.007**	**0.004**	**0.013**	**0.006**	**0.011**
	GloVe-100	0.009	0.005	0.018	**0.006**	0.008
	GloVe-200	0.010	0.006	0.014	0.007	0.009
	GloVe-300	0.009	0.005	0.016	0.007	0.007
	SGNS-300	**0.007**	**0.004**	**0.013**	**0.006**	**0.011**

Table 9. The statistical results of the *ATV* based method with various word embeddings

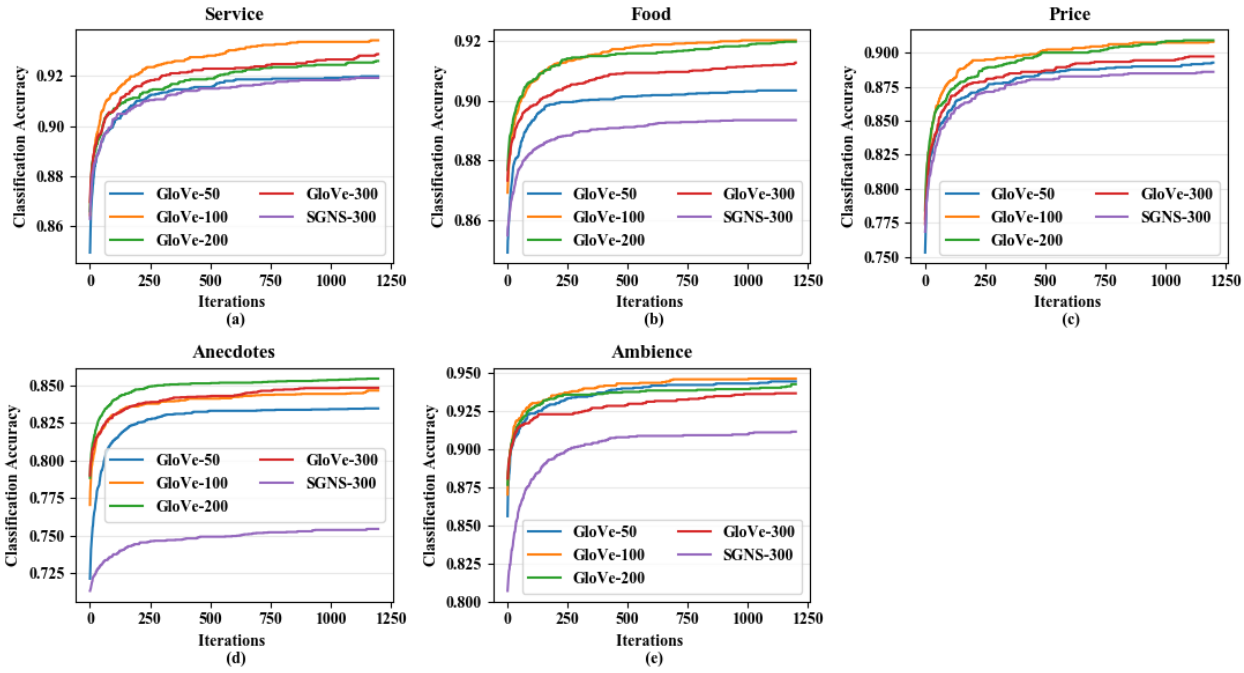


Fig. 15. Convergence curves on the training sub-datasets with various word representation models

5.4 Characteristic Analysis of the Attention-based Word Embeddings

There exists a need to analyze the characteristics of the optimized *ATV*, i.e. finely discussing the elements contained in the *DATV* and *SATV* sub-vectors and trying to extract certain regularities across different aspects.

1) Characteristics of the Sentiment Attention Sub-Vector

The $SATV$ sub-vector optimized by the CGABC algorithm for each aspect is presented in Table 10, where "-" means that the attribute is not included. Interestingly, the weights for the different types of intensifiers and transitions, i.e. the {A_INT, C_INT, N_INT, R_INT, V_INT} and the {C_TRS, N_TRS, R_TRS}, are all tuned to 0 on all aspects, which may be due to the fact that the aim of the ATV vector is to search for those dimensions that can directly indicate the aspect and its relative sentiment, thus the intensifiers and transitions, which have indirect relations with the topic, will achieve less "attentions". For the attributes without PoS tags, i.e. {"A", "C", "N", "R", "V"}, the values on all aspects are larger than zeros, almost around 0.5, expect that of the "V" attribute (i.e. Verb), the values of which are relatively low, especially on the "anecdotes" and the "ambience" aspects, being 0.380 and 0.375, respectively. The reason could be that consumers are prone to comment on restaurants using descriptive words, such as adjectives and adverbs, thus verbs have less opportunities to dominate or reveal the sentiments of sentences. In most cases, the values of positive attributes are slightly higher than that of neutral attributes, such as "A_PST", "N_PST", "R_PST" and "V_PST", while the "C_PST" has larger values than the "C_NU" on all aspects, since the conjunctions like "yet", "but" and "or", have limit relationships with the sentiments, apart from implying the emotional transition. The negative attributes, i.e. {"A_NG", "C_NG", "N_NG", "R_NG", "V_NG"}, have extremely low values, being negative in some situations, demonstrating the effectiveness of the $SATV$ that can adjust the weights of the dimensions through degrading the emphasis on certain negative words.

PoS Tag	Attribute	Service	Food	Price	Anecdotes	Ambience
Adjectives	**A**	0.423	0.493	0.606	0.527	0.527
	A_INT	0.000	0.000	0.000	0.000	0.000
	A_NG	-0.048	0.012	-0.036	0.090	-0.124
	A_NU	0.402	0.467	0.457	0.539	0.529
	A_PST	0.555	0.692	0.612	0.658	0.746
Conjunctions	**C**	0.563	0.571	0.440	0.508	0.489
	C_INT	0.000	0.000	0.000	0.000	0.000
	C_NG	0.151	0.141	0.021	0.149	0.089
	C_NU	0.634	0.483	0.480	0.481	0.464
	C_PST	0.482	0.402	0.416	0.462	0.426
	C_TRS	0.000	0.000	0.000	0.000	0.000
Nouns	**N**	0.504	0.411	0.407	0.547	0.585
	N_INT	0.000	0.000	0.000	-	0.000
	N_NG	-0.257	0.046	-0.078	-0.003	0.019
	N_NU	0.478	0.395	0.546	0.562	0.415
	N_PST	0.530	0.491	0.339	0.500	0.568
	N_TRS	0.000	0.000	-	-	-
Adverbs	**R**	0.563	0.545	0.510	0.508	0.536
	R_INT	0.000	0.000	0.000	0.000	0.000
	R_NG	0.141	0.152	-0.105	0.033	0.050
	R_NU	0.610	0.527	0.542	0.549	0.486
	R_PST	0.663	0.658	0.432	0.576	0.513
	R_TRS	0.000	0.000	0.000	0.000	0.000
Verbs	**V**	0.453	0.435	0.519	0.380	0.375
	V_INT	-	0.000	0.000	0.000	0.000
	V_NG	0.024	0.232	0.055	-0.004	0.082
	V_NU	0.619	0.523	0.473	0.450	0.425
	V_PST	0.508	0.552	0.528	0.522	0.427
Others	**UNIIMPT**	0.530	0.409	0.507	0.458	0.389

Table 10. Average *SATV* sub-vector optimized for the five aspects

2) Characteristics of the Dimension Attention Sub-vector

This set of experiments is conducted by computing the cosine similarities between the words contained in each sub-dataset and the corresponding *DATV* sub-vector optimized by the CGABC algorithm with the GloVe-50 embeddings. Then, for the aspect asp_p, the top 20 words that are most similar to the $DATV_P^{best}$ are listed in Table 11. Despite the same words with respect to the common topic for the five aspects, such as "eats", "dish", "cooked" and "meal", there also exists some specific words that are highly related to each aspect (as bolded and underlined). For example, words like "treats", "convenient", "helpful" and "patient" are close to the "service" aspect, and the "ambience" aspect is connected with certain emotional words, such as "comforting", "unbearable" and "enjoyable". It indicates that the $DATV_P^{best}$ can build up a strong relationship with words related to a specific aspect, in other words, $DATV_P^{best}$ has the ability of distinguishing the differences among various dimensions and selecting the most appropriate ones for each aspect.

Service		**Food**		**Price**		**Anecdotes**		**Ambience**	
Word	Similarity	Word	Similarity	Word	Similarity	Word	Similarity	Word	Similarity
tasty	0.464	**broth**	0.516	eats	0.465	eats	0.501	eats	0.503
fridge	0.462	**flavorless**	0.479	tasty	0.462	cooking	0.453	**'pub'**	0.489
treats	0.451	eats	0.475	noodles	0.455	dish	0.431	tasty	0.445
noodles	0.435	**tasty**	0.461	dish	0.425	ingredients	0.426	cooked	0.417
cooked	0.427	fridge	0.452	dishes	0.425	**wasting**	0.423	convenient	0.410
meals	0.420	**fatty**	0.452	ingredients	0.424	calories	0.422	meals	0.406
calories	0.417	**cooking**	0.445	**economical**	0.416	**drawback**	0.410	dishes	0.397
convenient	0.415	**lasagna**	0.432	**cheapest**	0.387	sushi	0.386	**comforting**	0.393
dish	0.414	**calories**	0.423	entrees	0.384	dumplings	0.381	meal	0.392
ingredients	0.414	**noodles**	0.420	seafood	0.383	meal	0.378	sushi	0.385
dishes	0.409	cooked	0.419	meal	0.383	seafood	0.372	desserts	0.374
platter	0.404	**soups**	0.418	congee	0.381	entrees	0.368	**unbearable**	0.371
pasta	0.397	**dish**	0.416	toppings	0.378	entree	0.359	entrees	0.370
meal	0.397	**ingredients**	0.413	vegetables	0.373	eating	0.356	**enjoyable**	0.364
helpful	0.395	cheapest	0.411	sushi	0.365	**worthwhile**	0.355	eating	0.361
craving	0.394	**meals**	0.409	pints	0.360	appetizer	0.351	entree	0.357
irritating	0.391	**paneer**	0.409	soup	0.357	**weeknight**	0.350	terminal	0.350
patient	0.390	platter	0.406	dumplings	0.356	**cravings**	0.348	dumplings	0.344
deliveries	0.386	**dishes**	0.405	deliveries	0.353	**vegetarian**	0.348	food	0.343
freshest	0.384	**overcooked**	0.404	appetizers	0.351	delicious	0.347	eat	0.340
entrees	0.384	**sardines**	0.404	**inexpensive**	0.348	**hassle**	0.342	drinking	0.336

Table 11. Similar words to the average *DATV* sub-vector optimized for the five aspects

5.5 Performance Analysis of the ATV-CNN Model

1) Performance analysis of the ATV-CNN model with various word embeddings across the five aspects

The accuracies and F1-scores of the ATV-CNN and CNN model with various dimensionalities have been listed in Table 12. It is obvious that the ATV-CNN model can easily outperform the CNN model. Since the testing sets of the sub-datasets only contain a few sentences as shown in Table 4, the ATV-CNN model can achieve good accuracy, especially on the GloVe-300 and the SGNS-300 embeddings, where the percent of correct predictions is up to 100% on all aspects. Different from the training process with the SVM, where the GloVe-200 performs best, the performance of the ATV-CNN model (i.e. the accuracy and F1-score) shows an upward tendency with an increasing dimensionality.

2) Performance Analysis of the ATV-CNN Model with Various Word Embeddings

The performance of the ATV-CNN model will be further validated on the whole dataset D, i.e. simultaneously training on the sub-datasets of different aspects. For comprehensive comparisons, both binary prediction (i.e. {"positive", "negative"}) and 3-class prediction ({"positive", "negative", "neutral"}) are contained in this group of experiments, and the corresponding accuracies and F1-scores are given in Table 13.

Surprisingly, the ATV-CNN model can achieve satisfactory performances with various word embeddings, where both the accuracies and F1-scores are up to 100% regarding to the binary prediction. For the three-class prediction, the accuracies and F1-scores are at least 95.6 % and 91.4 %, respectively, and the GloVe-300 word embeddings obtain the best performance, with an weak advantage (nearly 1%) over the SGNS-300. It indicates that different from the training process, where GloVe-100 and GloVe-200 show superiorities, large dimensionalities are preferred by the ATV-CNN model. This may be explained by the fact that the ATV-CNN model with "non-static" channel can refine the word embeddings during the training process.

Metric	Word embeddings	Models	Service	Food	Price	Anecdotes	Ambience
Accuracy	GloVe-50	CNN	0.875	0.950	0.909	0.842	0.857
		ATV-CNN	0.875	0.975	0.909	0.921	**1.000**
	GloVe-100	CNN	0.875	0.975	0.909	0.947	**1.000**
		ATV-CNN	0.875	0.975	**1.000**	0.947	**1.000**
	GloVe-200	CNN	0.875	0.975	**1.000**	**1.000**	**1.000**
		ATV-CNN	0.875	**1.000**	**1.000**	**1.000**	**1.000**
	GloVe-300	CNN	0.875	0.975	**1.000**	0.974	**1.000**
		ATV-CNN	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	SGNS-300	CNN	**1.000**	0.975	**1.000**	**1.000**	0.857
		ATV-CNN	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
F1-score	GloVe-50	CNN	0.795	0.922	0.807	0.457	0.788
		ATV-CNN	0.795	0.963	0.807	0.811	**1.000**
	GloVe-100	CNN	0.795	0.963	0.807	0.885	**1.000**
		ATV-CNN	0.795	0.963	**1.000**	0.885	**1.000**
	GloVe-200	CNN	0.795	0.963	**1.000**	**1.000**	**1.000**
		ATV-CNN	0.795	**1.000**	**1.000**	**1.000**	**1.000**
	GloVe-300	CNN	0.795	0.963	**1.000**	0.947	**1.000**
		ATV-CNN	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	SGNS-300	CNN	**1.000**	0.963	**1.000**	**1.000**	0.788
		ATV-CNN	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table 12. Experimental results of the ATV-CNN model with various dimensionalities

Word embeddings	Binary Prediction		Three-class Prediction	
	Accuracy	F1-score	Accuracy	F1-score
GloVe-50	**1.000**	**1.000**	0.956	0.914
GloVe-100	**1.000**	**1.000**	0.965	0.926
GloVe-200	**1.000**	**1.000**	0.973	0.938
GloVe-300	**1.000**	**1.000**	**0.991**	**0.973**
SGNS-300	**1.000**	**1.000**	0.982	0.963

Table 13. The performance of the ATV-CNN model with various word embeddings

5.6 Comparisons between the ATV-CNN Model and the State-of-the-art Models

The ATV-CNN model is further compared with the state-of-the-art models, including TD-LSTM / TC-LSTM models proposed in (Tang et al., 2016), LSTM/AE-LSTM/AT-LSTM/ATAE-LSTM

models proposed in (Wang et al., 2016) and the IAN model proposed in (Ma et al., 2017). Note that the results of these models are all taken directly from the original paper, as presented in Table 14[8].

Apparently, the accuracies of the compared models are all below 90% for both the binary and three-class prediction, which is substantially inferior to that of the ATV-CNN model. More specifically, the unsatisfactory performance of the LSTM model may be attributed to the lack of aspect information, which may result in the same predictions when given different aspects. The TD-LSTM and TC-LSTM models are slightly better than the LSTM model, benefiting from processing the left and right contexts with targets. Nevertheless, these two models only take account of the target rather than the aspect information, which does not exactly fit into the aspect-level classification problem. The performances of the AT-LSTM and ATAE-LSTM can stably surpass that of the other LSTM-based models, due to the usage of the attention mechanism. Similarly, the IAN model designed two connected attention networks to learn the representations of target and context, whereas it performs worst among all models. On a whole, although certain models can achieve performance improvements by introducing the attention mechanism, the scope of them are confined to discovering the relationships between the individual contexts and the aspects, neglecting the characteristics of the word embeddings. Considering the shortcomings hidden in the word embeddings, not surprisingly, these models cannot outperform the ATV-CNN model, which solves the problem from the view of emphasizing salient dimensions in the semantic space.

Model	Binary Prediction	Three-class Prediction
LSTM	0.883	0.820
TD-LSTM	0.891	0.826
TC-LSTM	0.892	0.819
AE-LSTM	0.889	0.825
AT-LSTM	0.896	0.831
ATAE-LSTM	0.899	0.840
IAN	-	0.786
ATV_CNN	**1.000**	**0.991**

Table 14. Comparisons between the ATT-CNN model and other state-of-the-art models in terms of accuracy

6 Conclusions

In order to solve the aspect-level sentiment classification problem, this paper employs an attention mechanism to refine the word embeddings, which can obtain sophisticated word vectors for each aspect in terms of a specific task. To the best of our knowledge, it is the first time that the attention mechanism is introduced into the word representation mechanisms. Essentially, an attention vector is proposed, involving two sub-vectors, called the Dimension Attention ($DATV$) and the Sentiment Attention ($SATV$). The $DATV$ sub-vector is used to determine the significance of each dimension in the semantic space, according to their relevance with an aspect; and the $SATV$ sub-vector can assign scores for words based on their sentiment polarities and PoS tags. As for the disadvantages involved in the pretrained word embeddings, the $DATV$ can target on a particular sense of polysemy, and the $SATV$ can help distinguish the antonyms. After the optimization process performed by several ABC variants, the $DATV$ is used to scale the pretrained word embeddings, which can be further used as the inputs of the CNN model, named as ATV-CNN model. The experiments demonstrate that the ATV-CNN model has significant advantages over the state-of-the-art models.

[8] Since the models for comparisons were trained on the GloVe model, the accuracies of the ATV-CNN model with the GloVe-300 word embeddings are listed in Table 14.

Acknowledgements

This work was supported in part by the National Science Foundation of China (61572238) and the Provincial Outstanding Youth Foundation of Jiangsu Province (BK20160001).

References

[1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1-167, 2012.

[2] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proc. 9th Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain, Jan. 2004, pp. 412- 418.

[3] B. Pang and L. Lillian, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1-135, 2008.

[4] B. Pang, L. Lillian, and V. Shivakumar, "Thumbs up?: sentiment classification using machine learning techniques" in *Proc. Empirical Methods in Natural Language Processing (EMNLP-2002)*, Philadelphia, PA, USA, July. 2002, pp. 79-86.

[5] W. Liu, Z. Wang, X. Liu, et al., "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11-26, April. 2017.

[6] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Empirical Methods in Natural Language Processing (EMNLP-2014)*, Doha, Qatar, Oct. 2014, pp. 1746–1751.

[7] L. Zhang, S. Wang and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, Jan. 2018.

[8] R. Socher, C.C. Lin, C. Manning, et al., "Parsing natural scenes and natural language with recursive neural networks," in *Proc. the 28th international conference on machine learning (ICML-11)*, Bellevue, Washington, USA, June. 2011, pp. 129-136.

[9] D. Tang, B. Qin and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. the 2015 conference on empirical methods in natural language processing*, Lisbon, Portugal, Sep. 2015, pp. 1422-1432.

[10] X. Wang, Y. Liu, C. Sun, et al., "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *Proc. the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, July. 2015, pp. 1343-1353.

[11] Y. Wang, M. Huang and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. the 2016 conference on empirical methods in natural language processing (EMNLP-2016)*, Austin, Texas, Nov. 2016a, pp. 606-615.

[12] D. Ma, S. Li, X. Zhang, et al., "Interactive attention networks for aspect-level sentiment classification," in *Proc. the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 4068-4074.

[13] T. Mikolov, K. Chen, G. S. Corrado, et al., "Efficient estimation of word representations in vector space," in *Proc. the International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, May. 2013a, pp. 1301-3781.

[14] T. Mikolov, W. Yih and G. Zweig, "Linguistic regularities in continuous space word representations," in Proc. the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, June. 2013b, pp. 746-751.

[15] D. Tang, F. Wei, B. Qin, et al., "Sentiment embeddings with applications to sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 2, pp. 496-509, Feb. 2016a.

[16] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, et al., "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214-224, Mar. 2017.

[17] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge & Data Engineering*, vol. 28, no. 3, pp. 813-830, Oct. 2015.

[18] M. Tsytsarau and T. Palpanas, "Survey on Mining Subjective Data on the web," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 478-514, May. 2012.

[19] Z. Hai, K. Chang and J. Kim, "Implicit feature identification via co-occurrence association rule mining," in *Proc. International Conference on Intelligent Text Processing and Computational Linguistics*, Tokyo, Japan, Feb. 2011, pp. 393-404.

[20] J. Zhu, H. Wang, B. K. Tsou, et al., "Multi-aspect opinion polling from textual reviews", in *Proc. the 18th ACM conference on Information and knowledge management (ACM)*, Hong Kong, China, Nov. 2009, pp. 1799-1802.

[21] J. Wang, L.C. Yu, K.R. Lai, et al., "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, Aug. 2016b, pp. 225-230.

[22] D. Tang, B. Qin, X. Feng, et al., "Effective LSTMs for target-dependent sentiment classification," in Proc. the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016), Osaka, Japan, Dec. 2016b, pp. 3298–3307.

[23] S. Ruder, P. Ghaffari and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," in Proc. *the 2016 conference on empirical methods in natural language processing (EMNLP-2016)*, Austin, Texas, Nov. 2016, pp. 999-1005.

[24] J. Liu and Y. Zhang, "Attention modeling for targeted sentiment," in Proc. the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2 (Short Papers) (EACL), Valencia, Spain, April. 2017, pp. 572-577.

[25] P. Chen, Z. Sun, L. Bing, et al., "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP-2017)*, Copenhagen, Denmark, Sep. 2017, pp. 452-461.

[26] S. Sukhbaatar, A. Szlam, J. Weston, et al., "End-to-end Memory Networks," in *Proc. the 28th International Conference on Neural Information Processing Systems (NIPS'15)*, Montreal, Canada, Dec. 2015, pp. 2440–2448,

[27] C. Li, X. Guo and Q. Mei, "Deep memory networks for attitude Identification," in *Proc. the ACM International Conference on Web Search and Data Mining (WSDM 2017)*, Cambridge, United Kingdom, Feb. 2017, pp. 671-680.

[28] N. Majumder, S. Poria, A. Gelbukh, et al., "IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis," in *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct. 2018, pp. 3402-3411.

[29] A. Abbasi, H. Chen and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, June. 2008.

[30] J. Carvalho, A. Prado and A. Plastino, "A statistical and evolutionary approach to sentiment analysis," in *Proc. the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Warsaw, Poland, Aug. 2014, pp. 110-117.

[31] H. Keshavarz and M. S. Abadeh, "ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs," *Knowl.-Based Syst.*, vol. 122, pp. 1-16, Apr. 2017.

[32] A. S. H. Basari, B. Hussin, I. G. P. Ananta, et al., "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Engineering*, vol. 53, pp. 453-462, 2013.

[33] D. K. Gupta, K. S. Reddy and A. Ekbal, "Pso-asent: Feature selection using particle swarm optimization for aspect based sentiment analysis," in *Proc. International conference on applications of natural language to information systems*, Passau, Germany, June. 2015, pp. 220-233.

[34] M. Pontiki, D. Galanis, J. Pavlopoulos, et al., "Semeval-2014 task 4: Aspect based sentiment analysis," in *Proc. the 8th international workshop on semantic evaluation (SemEval 2014)*, Dublin, Ireland, Aug. 2014, pp. 27-35

[35] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459-471, Nov. 2007.

[36] G. E Hinton, N. Srivastava, A. Krizhevsky, et al., "Improving neural networks by preventing co-adaptation of feature detectors,", 2012

[37] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 3, no. 23, pp. 533-536, Oct. 1986

[38] C. D. Manning, "Computational linguistics and deep learning," *Computational Linguistics*, vol. 41 no. 4, pp. 701-707, Dec. 2015.

[39] A. Mandelbaum and A. Shalev, "Word embeddings and their use in sentence classification tasks," arXiv preprint arXiv:1610.08229, Oct. 2016.

[40] R. Schwartz, R. Reichart and A. Rappoport, "Symmetric pattern-based word embeddings for improved word similarity prediction," in *Proc. the Nineteenth Conference on Computational Natural Language Learning*, Beijing, China, July. 2015, pp. 258-267.

[41] J. R. Firth, "A synopsis of linguistic theory," *Oxford University Press*, Oxford, UK, 1957, pp. 1930-1955.

[42] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, Nov. 2008.

[43] M. Pelevina, N. Arefyev, C. Biemann, et al., "Making sense of word embeddings," in Proc. the 1st Workshop on Representation Learning for NLP, Berlin, Germany, Aug. 2016, pp. 174-183.

[44] F. Hill, R. Reichart and A. Korhonen, "Simlex-999: Evaluating semantic models with (genuine) similarity estimation," *Computational Linguistics*, vol. 41, no. 4, Dec. 2015, pp. 665-695.

[45] D. A. Cruse, "Lexical Semantics," *Cambridge University Press*, Cambridge, UK, 1986

[46] M. Z. Asghar, A. Khan, S. Ahmad, et al., "A review of feature extraction in sentiment analysis," *Journal of Basic and Applied Scientific Research*, vol. 4, no. 3, pp. 181-186, Jan. 2014.

[47] Y. Wang, "Advanced Naïve Bayes Algorithm Design with Part-of-Speech Tagger on Sentiment Analysis,' in *Proc. 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, Dalian, China, Dec. 2017, pp. 1382-1385.

[48] S. Kiritchenko and S. M. Mohammad, "The effect of negators, modals, and degree adverbs on sentiment composition," in *Proc. the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, San Diego, California, June. 2016, pp. 43-52.

[49] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166-3173, Dec. 2010

[50] Gao W, Liu S and Huang L, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741-2753, May. 2012.

[51] W. F. Gao, S. Y. Liu and L.L. Huang, "Enhancing artificial bee colony algorithm using more information-based search equations," *Inf. Sci.*, vol. 270, pp. 112–133, June. 2014.

[52] M. Zhang, N. Tian, V. Palade, et al., "Cellular Artificial Bee Colony Algorithm with Gaussian Distribution," *Inf. Sci.*, vol. 462, pp. 374-401, Sep. 2018.

[53] F. Å. Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs," in Proc. ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages. Volume 718 in CEUR Workshop Proceedings, Heraklion, Crete, Mar. 2011, pp. 93-98.

[54] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," in *Proc. the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, Aug. 2004, pp. 168–177.

[55] T. Wilson, J. Wiebe and P. Hoffmann, "Recognizing Contextual Polarity in Phrase-level Sentiment Analysis," in *Proc. the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, BC, Canada, Oct. 2005, pp. 347-354.

[56] S. Mohammad, C. Dunne and B. Dorr, "Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus," in *Proc. the 2009 Conference on EMNLP*, Morristown, NJ, USA. Aug. 2009, pp. 599-608.

[57] S. Mohammad, S. Kiritchenko and X. Zhu, "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets," in *Proc. the Seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, GA, USA, June. 2013, pp. 321-327.

[58] S. Kiritchenko, X. Zhu, C. Cherry, et al., "NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews," in *Proc. the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Dublin, Ireland, Aug. 2014, pp. 437-442.

[59] L. Augustyniak, T. Kajdanowicz, P. Szymański, et al., "Simpler is better?: lexicon-based ensemble sentiment classification beats supervised methods," in *Proc. the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Beijing, China, Aug. 2014, pp. 924-929.

[60] S. Kiritchenko and S. Mohammad, "The effect of negators, modals, and degree adverbs on sentiment composition," in *Proc. the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, San Diego, California, June. 2016, pp. 43-52.

[61] S. Baccianella, E. Andrea and S. Fabrizio, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," *Lrec*., vol. 10, pp. 2200-2204, Jan. 2010.

[62] J. Duchi, E. Hazan and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121-2159, Jul. 2011.