Learning Multiple Layers of Knowledge Representation for Aspect Based Sentiment Analysis

Duc-Hong Pham[a,c], Anh-Cuong Le[b,*]

[a] *Faculty of Information Technology, Electric Power University, Hanoi city, Vietnam*
hongpd@epu.edu.vn
[b] *NLP-KD Lab, Faculty of Information Technology, Ton Duc Thang University, HoChiMinh city, Vietnam*
leanhcuong@tdt.edu.vn
[c] *Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam*

Abstract

Sentiment Analysis is the task of automatically discovering the exact sentimental ideas about a product (or service, social event, etc.) from customer textual comments (i.e. reviews) crawled from various social media resources. Recently, we can see the rising demand of aspect-based sentiment analysis, in which we need to determine sentiment ratings and importance degrees of product aspects. In this paper we propose a novel multi-layer architecture for representing customer reviews. We observe that the overall sentiment for a product is composed from sentiments of its aspects, and in turn each aspect has its sentiments expressed in related sentences which are also the compositions from their words. This observation motivates us to design a multiple layer architecture of knowledge representation for representing the different sentiment levels for an input text. This representation is then integrated into a neural network to form a model for prediction of product overall ratings. We will use the representation learning techniques including word embeddings and compositional vector models, and apply a back-propagation algorithm based on gradient descent to learn the model. This model consequently generates the aspect ratings as well as aspect weights (i.e. aspect importance degrees). Our experiment is conducted on a

*Corresponding author at: Ton Duc Thang University, Ho Chi Minh City, Vietnam

data set of reviews from hotel domain, and the obtained results show that our model outperforms the well-known methods in previous studies.
Keywords: Sentiment analysis; aspect based sentiment analysis; representation learning; multiple layer representation; compositional vector models; word embeddings.

---

1. Introduction

The explosive growth of social media on the Internet has helped people not only receive information on the networks but also generate information to others. Online interaction is becoming more real. People can discuss and give information about anything on social networks, Twitter, forums, blogs, etc. There is a special kind of information that is about opinions, evaluations, feelings, and attitudes. This information comes from the customers when they talk about the services or products they have used, or about the social events in their lives. Online interaction also changes the traditional purchasing behaviors, as well as social studies. Customers often look for online reviews about a product or service that they intend to use. Authorities also go online to find information about people's comments about social events. With this trend, there are more studies on automatic analysis and synthesis of information from customer reviews collected from social media. Thanks to the useful information provided by these studies, manufacturers can improve their products, the authorities can adjust policies accordingly, as well as customers can choose the product best suited to their conditions.

The development of technology along with the demand of analyzing opinionated information has led to a new research topic in natural language processing and data mining named "opinion mining and sentiment analysis". Studies on this problem started from the 2000s, addressing some problems including polarity classification [1, 2], subjectivity classification [3, 4, 5, 6], and opinion spam detection [7, 8, 9]. Early studies focused on the simple inputs which usually contain the opinion on one subject and the task is how to classify this opinion

into the classes negative, neutral or positive [10, 11, 12]. Recent problems with more complicated inputs have attracted many researchers. A review often contains evaluations on different product aspects, or contain comparable opinions. Some problems of concern include detecting comparable sentences [13, 14], determining aspects [15, 16, 17], rating aspects [18, 19, 20] or determining aspect weights [21, 22, 23]. Aspect-based sentiment analysis recently becomes an important problem, in which we need to give the synthesized sentiment on every product aspects. That is because in some cases, customers want to know not only the evaluation on the product but also on each of its aspects. Manufacturers may also want to know about the evaluation on each product aspect as well as its importance degree to customers. Some studies addressed this problem with the two assumptions of the input: the first one assumes each review is assigned wit aspects' ratings; the second one assumes each review is assigned with the overall rating for the product but is not assigned with aspects' ratings. This paper follows the second one and the problem here is how to derive the aspect ratings as well as aspect weights from a set of reviews given overall ratings.

Some previous studies such as [24, 25] have proposed a model called the Latent Rating Regression (LRR) which is a kind of Latent Dirichlet Allocation to analyze both aspect ratings and aspect weights, or [26] used the Maximum A Posterior (MAP) technique to tackle the aspect sparsity problem. However, these previous studies have had limitations with their methods. They developed classification methods in which they used a bag of words from the input text as the feature set. These models used directly the words as features and derived results with the independent hypothesis of those features. Recently models for generating word embeddings [27, 28, 29], in which the real-valued vectors represented for input words have been wildly used in various models such as in [30, 31]. Some deep learning techniques have been applied for aspect based sentiment analysis such as convolutional neural network [32], deep memory neural network [33], long short term memory [34]. In general, most of these studies focus on the tasks of aspect term extraction, aspect category detection, aspect level sentiment classification and they have not yet done for the task of

analyzing aspect ratings as well as aspect weights.

Different from previous studies, in this paper we will discover a new approach of representation learning for the task of detecting aspect ratings and aspect weights. Our view focuses on how to utilize the representation learning methods and the deep learning mechanism to form the nature of sentiment representation from textual opinions. This is based on an observation that the overall ratings of a review is a composition of its aspect ratings, and in turn each aspect rating is generated from different textual pieces of the review through words to sentences. Therefore, we will develop a model that draws a multi-layer representation with objective to form a compositional sentiment (i.e. the overall rating). By this model we have formulated the problem as natural as it should be. We have used a general framework of feed forward neural network integrated with the representation learning techniques of word embeddings [28] and compositional vector models [35] for capturing semantic information as well as get richer knowledge in higher representations. The parameters of this model will be learned with the objective of reaching the target overall ratings given by the training data set. This result consequently returns the aspect ratings and aspect weights as the problem objectives.

Especially in the architecture of the proposed model we have designed a layer called "higher aspect representation" aiming for sharing information between aspects that leads to enrich knowledge for the model. It can also solve the problem of long range dependencies between each aspect and its sentiments. It is worth to emphasize that our model has a different architecture to all previous studies and that is based on the multiple layer representation for knowledge combination from the textual opinions to overall sentiment ratings.

We evaluate our proposed model on the data collected from Tripadvisor[1] and use the five aspects including *Value*, *Room*, *Location*, *Cleanliness*, and *Service*. This data set is also used by the related previous research [24, 25]. The experimental result has shown the effectiveness of our model for multi-

[1] www.tripadvisor.com

layer representation in comparison with other models of feature representation such as bag of word, word vector averaging, or paragraph vector.

The rest of this paper is organized as follows: Section 2 presents related works; Section 3 introduces basic models for representation learning which will be used for our model; Section 4 includes the definition and notations of the problem. Section 5 presents our proposed model with multi-layer representation. Section 6 presents our algorithm for learning the proposed model; Section 7 describes our experiments and results. Some conclusions are presented in the last section.

2. Related Work

The task of aspect-based sentiment analysis can be divided into the sub-tasks as aspect term extraction, aspect category detection, aspect sentiment classification, aspect rating, and aspect weight determination. In general, a lot of studies have been solved these sub-tasks since the pioneering work of Hu et al. [15]. In this paper we just consider the works which is closed to our work that are aspect rating and aspect weight detection.

Aspect rating aims to assign a numeric rating (i.e $1 \sim 5$ stars) to each aspect in which a higher aspect rating means a more positive sentiment. Several studies also combine the tasks of determining overall ratings and aspect ratings into an account and learn the unified model. Snyder et al. [18] proposed the Good Grief Algorithm based on PRanking training algorithm for ranking aspects (i.e. rating aspects) using the dependencies among aspects. Titov et al. [19] used a topic based model and a regression model for extracting aspect terms as well as detecting aspect ratings.

Aspect weight detection is known as the problem of determining the importance degrees of aspects. Several studies have been addressed this problem such as Zha et al. [22] which has developed a probabilistic aspect ranking algorithm to determine the importance of aspects by using aspect term frequency and the influence of consumer opinions given to each aspect over their overall opinions

(i.e. aspect ratings). Pham et al. [20] proposed a least square based model to identify the important degree of aspects from reviews. However in this study the authors used the assumption that the overall ratings and the aspect ratings are explicitly provided in the training data.

Some other studies considered both aspect ratings and aspect weights as latent factors in an unified model and developed models for that problem. Wang et al. [24] proposed a probabilistic rating regression model to infer aspect ratings and aspect weights for each review. An extension of this model was provided by Wang et al. [25] which is an unified generative model called Latent Aspect Rating Analysis. Xu et al. [26] proposed the Sparse Aspect Coding Model (SACM) and used textual review contents associated with item intrinsic information. They employed $l1$-regularizer into the model to control the sparsity on the aspect proportions and used technical Maximum A Posterior (MAP) to estimate the rating on each aspect for each review. Note that the obtained results of determining aspect ratings and aspect weights in these works are strongly dependent on feature selection.

In recent year, representation learning and deep learning models have been efficiently applied for semantic representation learning of different levels of textual inputs such as words, phrases, sentences and documents. They have achieved remarkable results for the task of sentiment analysis. For example, Mikolov et al. [28] used a neural language model to learn word representations; Pennington et al. [29] proposed a weighted least squares model using global word-word co-occurrence counts and thus makes an efficient use of statistics, then produces word representations. The studies [36, 37, 38, 39] used word representations and applied a convolutional neural network model to extract higher level features of sentences/documents. Glorot et al. [40] used a stacked denoising autoencoder to extract a semantic representation for each review. Socher et al. [41] proposed a family of recursive deep neural networks (RNN) to compute compositional vector representations for phrases. Le et al. [42] considered the paragraph vector to be shared across all contexts, which is generated from the same paragraphs. It then proposed the paragraph vector

model to learn representations of sentences/paragraphs or documents. Pham et al. [43] used word representations and applied effective compositional vector models for the problem of rating Vietnamese comments. Some other studies [32, 33, 34] utilized deep learning techniques such as convolutional neural network, deep memory neural networ, long short term memory.

In this paper, different from all previous related studies we will propose an novel architecture for representing multiple layers of knowledge representation for textual opinions. Based on this representation we will design an effective model for predicting aspect ratings as well as aspect importance degrees.

3. Basic Models for Representation Learning

This section introduces two basic models of representation for being used in our proposed model. The first model is for generating word embeddings (it is also called word2vec) which presents each word as a vector in a semantic space, and this vector will be used as the input in our proposed model. The second model is used for generating higher layers of representation, such as sentences or documents.

3.1. Word Embedding and the CBOW model

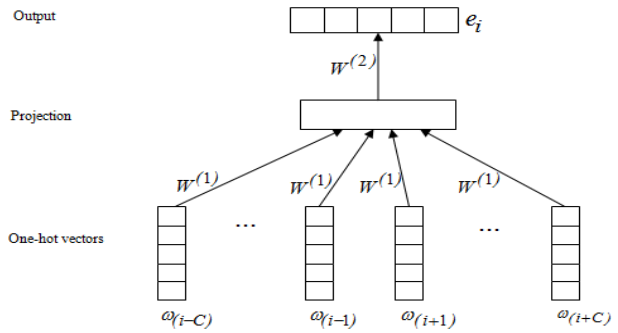

Figure 1: The continuous bag-of-words model

Word embedding models represent words with real-valued vectors whose relative similarities correlate with semantic similarity. Such vectors are used

both as an end in itself for computing similarities between terms, and as a representational basis (i.e. features) for NLP tasks like text classification, document classification, information retrieval, question answering, name entity recognition, sentiment analysis, and so on.

Word embedding models base on statistics of word occurrences in a corpus to encode semantic information which expresses how meaning is generated from these statistics, and how the resulting word vectors might represent that meaning. Bengio et al. (2003) introduced a model that learns word vector representations as part of a simple neural network architecture for language modeling. The skip-gram and continuous bag-of-words (CBOW) models proposed by Mikolov et al. in [28] have been widely used in many NLP tasks. The skip-gram model uses the current word to generate the context words meanwhile in the CBOW model the current word is generated from its context words. Recently [29] proposed GloVe which is a new global log-bilinear regression model for the unsupervised learning of word representations that outperforms the original models of Skip-gram and CBOW on word analogy, word similarity.

In the following, we briefly describe the CBOW model in [28] which is used in our proposed model.

Let $V = \{\omega_1, \omega_2, ..., \omega_{|V|}\}$ be a dictionary, each word $\omega \in V$ is represented by a one-hot encoded vector $u(\omega)$ of $|V|$ dimensions, which means to be 1 at the indexing position of ω, and all other $|V| - 1$ indexing positions are 0. Given a sentence S including m words, $S = (\omega_1, \omega_2, ..., \omega_m)$. Denoted $\omega_i \in S$ is the target word, which is predicted based on the context $h_i = (\omega_{(i-C)}, ..., \omega_{(i-1)}, \omega_{(i+1)}, ..., \omega_{(i+C)})$, where C is a size of context. The CBOW model takes the one-hot vectors of context words in h_i as input and the one-hot vector of the word ω_i as output, the architecture of the CBOW model is presented in Figure 1.

There are two parameter matrices to be learnt in the CBOW model: $W^{(1)} \in R^{|V| \times n}$ and $W^{(2)} \in R^{n \times |V|}$, in which $W^{(1)}$ is the weight matrix between the input layer and the projection layer and $W^{(2)}$ is the weight matrix between the projection layer and the output layer, where n is an arbitrary size which defines

the size of our embedding space. The i-th row of $W^{(1)}$ is the n-dimensional embedded vector for word w_i, the j-th column of $W^{(2)}$ is an n-dimensional embedded vector for word w_j. At the projection layer, the weights are shared for all words by the average of the vectors of the input context words. The objective function of CBOW is computed as follows:

$$E_\theta = \frac{1}{|V|} \sum_{i=1}^{|V|} \log p(\omega_i | \omega_{i-C}, ..., \omega_{i-1}, \omega_{i+1}, ..., \omega_{i+C})$$

Where $\theta = \{\mathrm{W}^{(1)}, \mathrm{W}^{(2)}\}$ is a parameter set of the model, and the probability $p(\omega_i | \omega_{i-C}, ..., \omega_{i-1}, \omega_{i+1}, ..., \omega_{i+C})$ is computed by using the softmax function. In order to compute these parameters, we can apply the back-propagation algorithm with stochastic gradient descent to minimize the function $-E_\theta$.

3.2. Compositional Vector Models

Vector representation for words has proved very efficient for many NLP tasks when comparing with using spelling word forms. Despite their widespread use, it is typically directed at representing words in isolation that doesn't cover semantic representation of larger structures like phrases, sentences, or even documents. In fact, one common method for generating a vector of a sentence (a text in general) is to average all the vectors of words in this sentence. Suppose that we are working to learn the representation of a sentence. Figure 2 is an illustration of using the compositional vector model for sentence representation, in which each word in the input sentence is represented by a vector (usually done by a word2vec model), and all word vectors of the sentence will be combined through a called composition function to generate the target vector (i.e. the new representation of the input sentence). Mitchell et al. [44] used the combination rules with addition and multiplication operators to generate a higher-level representation for a sentence/document. Some more complex composition functions using parsed tree, matrix-vector composition, convolutional neural networks or tensor composition have been proved useful [45, 46, 47, 48]. Some of these works employed deep linguistic structures as parsed trees to design their composition

functions, and others used other semantic signals such as sentiment or topic labels for designing the objective functions.

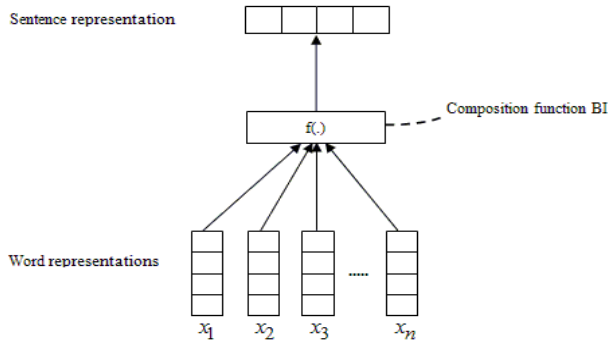

Figure 2: An illustration of using the compositional vector model for sentence representation

Hermann et al. [35] introduced two composition functions named ADD and BI. The ADD function computes a sentence representation by summing all word vectors of the sentence, which is a distributed bag-of-word that doesn't take into account the word order. The BI function is designed to capture bi-gram information, using non-linearity over bi-gram pairs. In particular, let x denote a sentence including n word vectors $x_1, x_2, ..., x_n$, the dimension size of word vector is m, then the composition function is defined by:

$$v(x) = \sum_{i=1}^{n} f([x_{i-1} + x_i]) \tag{1}$$

where $v(x)$ is the representation vector of x, $f([a+b])$ is element-wise weighted addition of two vectors a and b. The function $f(.)$ is defined using the hyperbolic tangent function as follows:

$$f(y) = \tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}} \tag{2}$$

According to [35] using the nonlinear function *tanh* enables the model to learn interesting interactions between words in a sentence/document, which the bag-of-words approach of ADD is not capable of learning. The BI composition

function is also used effectively in some other studies such as [43, 49]. In addition, following this approach can avoid complicated requirements of the language resources (such as parsed trees or topic labels) but still capture useful combined information. Especially it can be easy to apply in low resource languages. Because of all its advantages, the BI functions is chosen in our model as the activation function for learning representations of sentences and aspects. Particularly, the representation vector $v(x)$ for sentence x is computed according to the activation function as follows:

$$v(x) = \sum_{i=1}^{n} f(\mathrm{M} \odot [x_{i-1} + x_i] + [\mathrm{b}]) \tag{3}$$

where $\mathrm{M} \in \mathrm{R}^{\mathrm{m}\times\mathrm{m}}$ is the weight parameter matrix at sentence level, $\mathrm{b} \in \mathrm{R}^{\mathrm{m}}$ is a bias vector, which are learned during training and $\odot$ denotes element-wise multiplication operator.

Note that in our model, each aspect denotes its sentiment from the set of sentences related to this aspect. Therefore, we first use the compositional vector model to compute the representation for each sentence. After that, at the aspect level, we use the sentence representations as input to compute the representation of the aspect.

4. Problem Definition

We follow the description in [24] to define the problem of aspect based sentiment analysis. By that, we are given a set of textual reviews $D = \{\mathrm{d}_1,\mathrm{d}_2,...,\mathrm{d}_{|D|}\}$ of a specific product (e.g. a hotel) containing sentiments about this product and its aspects. Each document $d \in D$ is assigned with an overall rating O_d which determines the whole sentiments of the product mentioned in d. Suppose that the overall rating O_d is composed from individual ratings of the product's aspects. Moreover, as a common observation, these aspects have different influences to the overall rating (here we denote by aspect weights). The problem here is how to determine aspect ratings as well as aspect weights .

In the following, we define necessary notations being used in our proposed model.

Overall Rating An overall rating of document d is denoted by O_d which is a value ranging from 1 star to 5 stars, as shown in many commerce websites. Actually we set O_d as a real value in $[1, 5]$ for easy computation.

Aspect Let $\{A_i\}$ where $i = 1, \ldots, k$ denote k aspects of the product, for example when describing a hotel we can talk about its aspects like *price*, *location*, *staff*, and so on. Note that in [24] each aspect is identified by a set of predefined words, but in our work we will use all related sentences for determining the corresponding aspect.

Aspect Ratings for each document $d \in D$ we denote aspect ratings for d is a k-dimensional vector $r_d = (r_{d1}, r_{d2}, \ldots, r_{dk})$, where the i-th dimension is a numerical measure, indicating the degree of sentiment in the review d corresponding to aspect A_i.

Aspect Weights for each document $d \in D$ we denote the aspect weights for d is a $k-$dimensional vector $\alpha_d = (\alpha_{d1}, \alpha_{d1}, \ldots, \alpha_{dk},)$, where the i-th dimension is a numerical measure, indicating the importance degree of aspect A_i on d and we require $0 \leq \alpha_{di} \leq 1$ and $\sum_{i=1}^{k} \alpha_{di} = 1$.

5. The Proposed Model with Multi-Layer Representation

We here propose a multiple layer representation for generating the overall sentiment from a given review. The objective of our model is to build an architecture that can model the process of prediction as the representation learning from the input as the lowest representation (i.e. sequence of words) to the highest representation (i.e. the overall sentiment).

The figure 3 shows our model's architecture. In this model, each word from the input text (i.e. review) is transformed into the corresponding semantic vector using the word embedding technique [28]. And then we combine all the words in a sentence to generate the sentence representation by using a composition model. Note that each review naturally might contain opinions

about different aspects, so it needs a pre-processing task of determining input segments corresponding to each aspect. This task is known as aspect segmentation in which we will group and assign related sentences to corresponding aspects. To this end, the aspect segmentation algorithm in [24] will be used in our approach. And then the corresponding sentence representations of an aspect will be composed to generate its representation. One more layer of aspect representation is added to enrich the representation before making aspect ratings. Finally the weighted combination of aspect ratings will generate the overall rating of the whole review.

In summary, our the model is a type of neural networks which contains the six layers: (1) word representation; (2) sentence representation; (3) aspect representation; (4) higher aspect representation; (5) aspect rating; (6) overall rating. In the following, we will present each layer with necessary formulations and notations. This model is named LRNN-ASR, where LRNN stands for the "Latent Rating Neural Network" and ASR stands for "Aspect Semantic Representation".

Word Representation Layer: at this layer we will use the CBOW - a word embedding model introduced by Mikolov et al. [28] - to obtain word representations. It is worth to recall that word embedding model takes words from a vocabulary as input and embeds them as vectors into a lower dimensional space which we can consider as a semantic space. The model for putting a word into a semantic space is learnt from a large data set of unlabeled sentences (it is independent and different from the data set of reviews). Thanks to being embedded in a common semantic space, a word has its own knowledge and makes relationship with others. Therefore, this is the first level of knowledge representation in our model.

Let us to define notations for later formulas: for each review $d \in D$ and aspect i-th, assume that its corresponding paragraph contains p sentences then we denote these sentences by $\{s_{di1}, s_{di2}, ..., s_{dip}\}$. Suppose each sentence s_{dij} contains q words we denote these words by $\{w_{dij1}, w_{dij2}, ..., w_{dijq}\}$. For each word w_{dijl}, using the CBOW method we will obtain its vector denoted by e_{dijl}.

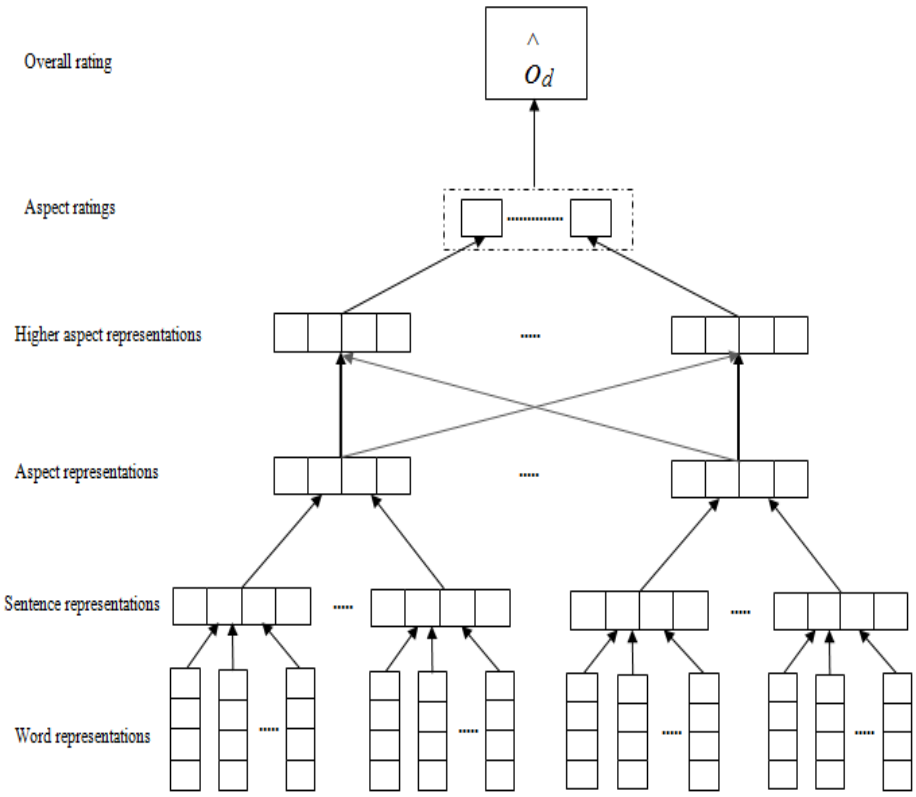

Figure 3: The proposed model with multi-layer representation

Sentence Representation Layer: after the first knowledge representation layer, separate word representations of a sentence are then synthesized to form an unified knowledge of the sentence. To do this task, we will use the compositional vector model presented in section 3.2, concretely using the equation 3. For each sentence s_{dij} with length q, the first layer gives its word vectors, that are $e_{dij1}, \ldots, e_{dijq}$. Denote the expected sentence vector by $v(s_{dij})$ then it is estimated by using the equation 3, as follows:

$$v(s_{dij}) = \sum_{l=1}^{q} f(\mathrm{U_i} \odot [e_{dij(l-1)} + e_{dijl}] + [\mathrm{u_{i0}}]) \tag{4}$$

where $\mathrm{U_i} \in \mathrm{R^{m \times m}}$ is the weight parameter matrix at sentence level of aspect A_i, $\mathrm{u_{i0}} \in \mathrm{R^m}$ is a bias vector, which are learned during training and $\odot$ denotes element-wise multiplication operator.

Note that by using this method we can capture information of all bi-grams in the sentence. The parameter matrix $\mathrm{U_i} \in \mathrm{R^{m \times m}}$ will be learnt by using a back-propagation algorithm which is based on the ground-truth overall ratings.

Aspect representation layer: aspect representations are generated by composing sentence representations using a compositional vector model. This layer receives sentence representation vectors corresponding to each aspect as the input and computes the aspect representation vector. We compute representation vector of aspect A_i for document d as the following equation:

$$x_{di} = \sum_{j=1}^{p} f(\mathrm{V_i} \odot [v(s_{di(j-1)}) + v(s_{dij})] + [\mathrm{v_{i0}}]) \tag{5}$$

where $\mathrm{V_i} \in \mathrm{R^{m \times m}}$ is the weight parameter matrix at aspect level of aspect A_i, $\mathrm{v_{i0}} \in \mathrm{R^m}$ is a bias vector. These vectors will be determined at the training phase.

Higher aspect representation layer: many studies have shown that using a multiple layers neural network will help to enrich knowledge of the representations and consequently improve prediction tasks. In this task, we are processing multiple aspects and actually each of them may influence the others. Therefore by building one more layer for aspect representation (which we call

"higher aspect representation layer") we aim to obtain and utilize the shared information between aspects. In the figure 3, look at the connection between the aspect representation layer and the higher aspect representation layer, you can see the bold lines and the thin lines. Each aspect representation has only one its own higher representation and they are linked by a bold line. The other thin lines which connect to a higher aspect representation from its neighbour aspect representations reflect their influences. Note that we can ensure the major information of a higher aspect representation comes from its corresponding aspect representation by initializing the appropriate weights for these connection lines. The actual weights are then determined from the training phase.

It is also interesting that this architecture can solve the problem in the case when an aspect having its sentiments in long distances. That is when a sentiment belongs to an aspect in different aspect segments, then the transformation from aspect to higher aspect representations will be the very good opportunity for this sentiment to be unified with its aspect.

By this construction we can ensure that each higher aspect representation is the representative for the corresponding aspect as well as help enriching the knowledge of the model. Therefore from the higher aspect representation can effectively generate the corresponding aspect ratings as shown in the the figure 3.

Denote $x^*_{d1}, ..., x^*_{dk}$ are the higher-level representation vectors of the k aspects in review d respectively. We propose an equation to compute the representation x^*_{di} of aspect A_i as follows:

$$x^*_{di} = [\sum_{j=1}^{k} (\delta(\mathrm{i} = \mathrm{j}).\beta.x_{di} + \delta(\mathrm{i} \neq j).\frac{\gamma}{k-1}x_{dj})] \tag{6}$$

where $0 < \beta \leq 1$ is the representation weight of the representation x_{di} in its new representation x^*_{di}, $0 \leq \gamma < 1$ is the shared feature weight between aspect A_i and $k-1$ remaining aspects, $\beta + \gamma = 1$,

$$\delta(y) = \begin{cases} 1; \text{ if } y = true \\ 0; \text{ if } y = false \end{cases}$$

This representation can be considered as the shared features between aspects, which helps the LRNN-ASR model to capture the relationship between aspects

for each specific aspect representation.

Aspect rating layer: note that our purpose is to derive aspect ratings and aspect weights. We design the aspect ratings layer which is generated from the higher aspect representation layer. As discussed above each higher aspect representation is derived from the corresponding aspect and furthermore enhanced with shared information from its neighbour aspects. These representations then generate elements of aspect ratings vector, one for one. By this way each of vector elements is also the representative of the corresponding aspect. Therefore the process of fitting the weighted sum over these vector elements to the overall ratings means the process of determining the aspect ratings and aspect weights. Concretely, from the higher aspect representation vectors $x^*_{d1}, ..., x^*_{dk}$ considered as the input, we will use the sigmoid function of a linear combination to compute the aspect rating r_{di} for aspect A_i as follows:

$$r_{di} = \text{sigm}(\sum_{l=1}^{m} \text{x}^*_{\text{dil}} \text{w}_{il} + \text{w}_{i0}) \tag{7}$$

where the parameter w_{il} and the bias w_{i0} will be determined from the learning phase.

Overall rating layer: the overall rating is the highest level of abstraction of the model and also be seen as the result of the model. From the aspect ratings layer, we use a weighted sum function to generate the overall rating. As a result, by learning the model we also determine the importance degrees of aspects, which is the second our objective. Suppose that we are using the weighted sum of weights α_d over the aspect ratings r_d to generate the overall $\hat{O_d}$ for the document d as the following formula:

$$\hat{O_d} = \sum_{i=1}^{k} r_{di} \alpha_{di} \tag{8}$$

where $0 \leq \alpha_{di} \leq 1$ for *i = 1, 2, . . . , k* with the condition $\sum_{i=1}^{k} \alpha_{di} = 1$
To avoid the computational complexity of optimization problem when estimating

the parameter$\{\alpha_{d1}, \ldots, \alpha_{dk}\}$ we use a set of auxiliary variables $\{\hat{\alpha}_{d1}, \ldots, \hat{\alpha}_{dk}\}$ and set the value α_{di} by:

$$\alpha_{di} = \frac{\exp(\hat{\alpha}_{di})}{\sum_{l=1}^{k} \exp(\hat{\alpha}_{dl})} \tag{9}$$

The equation 8 then becomes:

$$\hat{O}_d = \sum_{i=1}^{k} r_{di} \frac{\exp(\hat{\alpha}_{di})}{\sum_{l=1}^{k} \exp(\hat{\alpha}_{dl})} \tag{10}$$

The parameter set $\{\hat{\alpha}_{d1}, \ldots, \hat{\alpha}_{dk}\}$ will be learned at the learning phase which is presented in the next section.

6. Model Learning

Learning a model is the process of determining values for the model parameters that best fit the training data set, or in other view this process aims to minimize the error function over the training data set by adjusting the model's parameters. We'll use the back-propagation algorithm based on gradient descent for this task. Firstly we need to present in a more detail way the model's parameters, as follows.

- Let $\mathrm{U} = [\mathrm{U}_1^*, \mathrm{U}_2^*, ..., \mathrm{U}_\mathrm{k}^*]$ denote the set of parameters for learning sentence vectors at the sentence representation layer, corresponding to k aspects. Here, $\mathrm{U}_\mathrm{i}^* = \{\mathrm{U}_\mathrm{i}, \mathrm{u}_{\mathrm{i}0}\}$ contains the weight parameter matrix and the bias vector corresponding to aspect A_i as presented in equation 4.

- Let $\mathrm{V} = [\mathrm{V}_1^*, \mathrm{V}_2^*, ..., \mathrm{V}_\mathrm{k}^*]$ denote the set of parameters for learning aspect vectors at the aspect representation layer, corresponding to k aspects. Here, $\mathrm{V}_\mathrm{i}^* = \{\mathrm{V}_\mathrm{i}, \mathrm{v}_{\mathrm{i}0}\}$ contains the weight parameter matrix and the bias vector corresponding to aspect A_i as presented in equation 5.

- Let $\mathrm{W} = [\mathrm{w}_1^*, \mathrm{w}_2^*, ..., \mathrm{w}_\mathrm{k}^*]$ denote the set of parameters for learning aspect ratings, where $\mathrm{w}_\mathrm{i}^* = \{\mathrm{w}_\mathrm{i}, \mathrm{w}_{\mathrm{i}0}\}$ contains the weight vector w_i and the

bias w_{i0} corresponding with aspect A_i for $i = 1, \ldots, k$, as presented in equation 7.

- Let $\hat{\alpha} = \left[\hat{\alpha}\right]_{|D|xk}$ denote the parameter matrix for learning overall ratings. Each row of $\hat{\alpha}$, e.g. for document d, is the vector $\{\hat{\alpha}_{d1}, \ldots, \hat{\alpha}_{dk}\}$ used for computing the overall rating of document d from its k aspect ratings, as presented in equation 10. Note that from the matrix $\hat{\alpha}$ we will estimate the corresponding aspect weight by the equation 9. We denote $\alpha = [\alpha]_{|D|xk}$ is the aspect weight matrix corresponding to $\hat{\alpha}$.

In addition, we also denote by $R = [r]_{|D|xk}$ the aspect rating matrix where each its row is a aspect rating vector of a review. For each document d and aspect A_i then the aspect rating $r_{di} \in R$ is computed by equation 7. The purpose of model learning phase is to estimate the parameter sets $\{U, V, W, \hat{\alpha}\}$, and then derive the aspect rating matrix R and the aspect weight matrix α.
Let O_d denote the desired target value of the overall rating of review d, then the cross entropy cost function over the review d is:

$$C_d = -O_d \log \hat{O}_d - (1 - O_d) \log(1 - \hat{O}_d) \tag{11}$$

We will use the cross entropy error function over all documents of the data set $D = \{d_1, d_2, ..., d_{|D|}\}$ as the objective function for training the model. This objective function is estimated by:

$$E(U, V, W, \hat{\alpha}) = -\sum_{d \in D} (O_d \log \hat{O}_d + (1 - O_d) \log(1 - \hat{O}_d)) \tag{12}$$

In addition, without loss of generality and to avoid over-fitting, we add a regularization term to the loss function $E(\theta)$ as the following:

$$E(\theta) = -\sum_{d \in D} (O_d \log \hat{O}_d + (1 - O_d) \log(1 - \hat{O}_d)) + \frac{1}{2}\lambda \|\theta\|^2 \tag{13}$$

Where $\theta = [U, V, W, \hat{\alpha}]$ is the set of model parameters, λ is the regularization parameter and $\|\theta\|^2 = \sum_i {\theta_i}^2$ is a norm regularization term. In order to compute

the parameters θ, we apply back-propagation algorithm with stochastic gradient descent to minimize this cost function. Each element of the weights in the parameters θ is updated at time $t + 1$ according to the formula:

$$\theta(t+1) = \theta(t) - \eta \frac{\partial E(\theta)}{\partial \theta} \tag{14}$$

where η is the learning rate. Note that the gradient computation is presented in detail at the Appendix.

Algorithm 1 shows the process in steps for learning the proposed model. In which we are given a set of reviews that each review is assigned with an overall rating as as a customer opinion. To do the learning process, we first take a pre-processing task for the input so that the words are represented by vectors and sentences are grouped into subsets corresponding to aspects. After that, steps of the algorithm are performed to execute the back-propagation algorithm based on the model's architecture in Fig. 3.

Note that when implementing this algorithm we follow the mini-batching technique as presented in [50, 51] to solve the problem of large data. In this case when the number of documents in D is big, the data set D is then divided into smaller subsets and we will implement the step 2 of the algorithm 1 for each subset step by step. This solution will make the algorithm run faster.

Algorithm 1 Learning Model for Determining Aspect Ratings and Aspect Weights using Multiple Layers of Knowledge Representation.

Input: A set of textual reviews $D = \{d_1, d_2, ..., d_{|D|}\}$; each review $d \in D$ is assigned with an overall rating O_d

Output: Values for the parameters: $U, V, W, \hat{\alpha}$

Step 0: preprocessing for representing words by vectors; and do aspect segmentation for grouping sentences corresponding to aspects.

Step 1: Initialize values for: the learning rate η, the error threshold ε, the iterative threshold I, the regularization parameter λ, the shared feature weight γ;

initialize the parameters: $U, V, W, \hat{\alpha}$

Step 2: for *t=1* to I do

for each textual review $d \in D$ do

2.1. Compute α_{di} using Eq. 9;;

2.2. Compute sentence representations at time t using Eq. 4;

2.3. Compute aspect representations at time t using Eq. 5;

2.4. Compute higher aspect representations at time t using Eq. 6;

2.5. Compute aspect ratings at time t using Eq. 7;

2.6. Compute overall rating at time t using Eq. 8;

endfor

Update parameters in θ at time *t+1* using Eq. 14;

Compute the objective function by: $\frac{1}{|D|} \sum_{d=1}^{|D|} \left| O_d - \hat{O}_d(t) \right|$

Break if the objective function is less than the error threshold ε;

endfor

After obtaining W, w_0, R and $\hat{\alpha}$ we can easily to compute the aspect ratings R and aspect weights α according to the equations 9 and 7 respectively.

7. Experiment

7.1. Data and Preprocessing

The dataset used in our experiment is provided by authors of the papers [24, 25], which is located at *http://times.cs.uiuc.edu/wang296/Data*. Actually it is a new version of data used in [24, 25], which includes 174,615 reviews of 1,768 hotels crawled from a very famous tourist website *www.tripadvisor.com*. The data contains reviews about hotels and refers to five different aspects of the hotel, including *Value*, *Room*, *Location*, *Cleanliness*, and *Service*. Each review is assigned with an overall rating for the hotel, and each aspect is also assigned with an aspect rating. These ratings range from 1 star to 5 stars. Note that the aspect ratings won't be used in our models, they are just used for evaluation of the model at testing. In this experiment we randomly select a quarter of the whole data for testing and the remainder for training.

We perform some necessary pre-processing tasks on these reviews: 1) removing sentences that are not in English; 2) removing stop words using a standard stop word list as in [24] and removing low frequent words; 3) removing the differences between inflected forms of a word using the Stanford POS Tagger [52]; 4) removing the sentences which have less than three words;

In addition, to fit the prediction function of aspect ratings (i.e. the function as in Eq. (6)) to the training data, we normalize overall ratings and aspect ratings into real numbers in range $[0, 1]$ by taking their values divided by 5. Some statistics of the data are shown in Table 1.

Table 1: Some Statistics on the Dataset

Number of reviews	174,615
Number of hotels	1,768
Number of sentences	2,126,919
Average number of words in a sentence	7.50
Number of aspects	5

7.2. Implementation of the training phase

We perform the following steps:

- represent words by vectors using a word embedding method;
- aspect segmentation: mapping sentences to corresponding aspects;
- implement and execute the Algorithm 1 to learn the model's parameters.

Word representation: as mentioned earlier, word representation (i.e. word embeddings) plays an important role in our LRNN-ASR architecture. This work helps capturing semantic information of words, which affect directly the quality of sentence representation and aspect representation. In this experiment we perform the CBOW model [28] and use the tool Word2Vec[2]. To do this task, we use all the sentences of the given reviews including 2,126,919 sentences as the input and set 200 as the number of word vectors' dimension.

Aspect Segmentation: we apply the aspect segmentation algorithm presented in [24] for this task. As the obtained results, each sentence then is assigned with a corresponding aspect. To avoid data sparse, in the cases when a review doesn't mention all the aspects or sentiments are given separately, like [24] for each hotel we combine all the sentences regarding the same aspect in one paragraph. The set of paragraphs (each paragraph contains sentiments about one aspect) for each review then becomes a new review of this hotel, which is called a *h-review*. As the result, from the $174,615$ original reviews we obtain a corpus with $1,768$ h-reviews corresponding to the $1,768$ hotels.

Implementation of Algorithm 1

We firstly initialize the parameters of the Algorithm 1 as follows:

Based on the observation that important aspects usually receive a large number of opinions [22], we initialize the aspect weight $\hat{\alpha}_{di}$ for document d and aspect A_i by $\hat{\alpha}_{di} = \log(\frac{n_{di}}{\sum_{l=1}^{k} n_{dl}})$, where $n_{di} = \sum_{p=1}^{n} n_{dip}$ is the total counts of words in the segmented text of aspect A_i, and n_{dip} is the frequency of the p-th word corresponding to aspect A_i, $\sum_{l=1}^{k} n_{dl}$ is the total counts of words in review d of

[2]https://github.com/piskvorky/gensim/

all aspects.

For other parameters, we initialize them as follows[3]: the learning rate $\eta = 0.015$; the error threshold $\varepsilon = 10^{-4}$; the iterative threshold *I=1000*; the regularization parameter $\lambda = 10^{-5}$; all the elements in U, V, W are randomly initialized in the range of $[-1, 1]$; and the shared feature weight $\gamma = 0.45$.

Aspect weights, which are treated as model's parameters, are generated at the training phase. For example, Table 2 shows the result of aspect weights determined for the five hotels. From this result we can conclude that aspect *Values* is the most important aspect for the hotels "King George" and "Astoria" while it is *Room*, *Location*, *Service* for "Radisson Ambassador Plaza", "Barcelo Punta Cana", "Condado Plaza Hilton" respectively.
Note that there is actually no ground-truth aspect weights, thus we just consider aspect weights as consequence of model when it generates aspect ratings and overall ratings.

Table 2: An example of Aspect weights determined for the five hotels

Hotel Name	Value	Room	Location	Cleanliness	Service
Barcelo Punta Cana	0.160	0.006	**0.716**	0.005	0.014
Condado Plaza Hilton	0.003	0.005	0.369	0.006	**0.617**
King George	**0.877**	0.070	0.026	0.018	0.009
Astoria	**0.680**	0.006	0.053	0.257	0.004
Radisson Ambassador Plaza	0.238	**0.446**	0.300	0.011	0.005

7.3. Evaluation measures

We evaluate the performance of our model by using the three basic evaluation measures as used in [24], including: 1) Root Mean Square Error (RMSQ) on aspect rating prediction (denoted by Δ_{aspect}); 2) aspect correlation inside reviews (denoted by P_{aspect}); 3) aspect correlation across reviews prediction

[3]Some values here are selected by investigating a set of candidate values on a development data set.

(denoted by P_{review}). In the following formulas, we use the notations: r^*_{di} is the ground-truth rating for aspect A_i, and r_{di} is the predicted aspect rating; $P_{r^*_i, r_i}$ is the Pearson correlation between two vectors r^*_i and r_i.
These evaluation measures are then described as follows:

1. Root Mean Square Error on aspect rating prediction is defined as:
   $\Delta_{aspect} = \sqrt{\frac{1}{|D_{test}|} \sum_{d=1}^{|D_{test}|} \sum_{i=1}^{k} (r^*_{di} - r_{di})^2 / k}$

   The lower Δ_{aspect} means the better prediction.

2. Aspect correlation inside reviews is defined as:
   $P_{aspect} = \frac{1}{|D_{test}|} \sum_{d=1}^{|D_{test}|} P_{r^*_d, r_d}$
   P_{aspect} aims to measure how well the predicted aspect ratings can preserve the relative order of aspects within a review given by their ground-truth ratings. Therefore higher P_{aspect} means the better prediction.

3. Aspect correlation across all reviews is defined as:
   $P_{review} = \frac{1}{k} \sum_{i=1}^{k} P_{r^*_i, r_i}$
   P_{review} tells us whether the predicted ratings and the ground-truth ratings for aspect A_i would give a similar ranking of all the reviews in this aspect. Therefore the higher P_{review} means the better prediction.

7.4. Experimental Results and Comparisons

To conduct the experiment we randomly select a quarter of the given dataset for testing and the remainder for training. The training data set is used for learning the model which then runs on the testing data to obtain the aspect ratings. We execute this experiment for five times and take average of the their results as the final one. The obtained results are compared with the labeled data for evaluation.

In order to comparison, we re-implement the method in [24] which is popular and much related to our work in this paper. This is also done for our previous work in [21]. In addition, to show the effectiveness of using higher aspect

representation layer, we also implement an version of our proposed model without using the higher aspect representation layer. Abbreviations for these models express partly their meaning as well as their relations with others: our full model is denoted by "Full-LRNN-ASR" (i.e. Latent Rating Neural Network - Aspect Semantic Representation using the higher aspect representation layer), the name of our model without the higher aspect representation layer is "LRNN-ASR", the model in our previous study [21] has name "LRNN" (i.e. Latent Rating Neural Network), and the model's name in [24] is denoted by "LRR" (i.e. Latent Rating Regression).

Table 3 shows the obtained results of aspect ratings detection through the three evaluation measures. Note that in LRR and LRNN models we have implemented various kinds of features including "bag of word", "word vector averaging", "sentence vector averaging", and "paragraph vector". The detail descriptions for these feature kinds are presented in [21].

Table 3 shows that our models outperform the previous models with all feature kinds for all the evaluation measures Δ_{aspect}, P_{aspect}, and P_{review}. It is interesting to consider the role of the feature selections in the implementation of these models. It is reasonable for understanding that "bag of word", "word vector averaging", "sentence vector averaging" and "paragraph vector" are the feature representations from coarse to fine, i.e. "paragraph vector" is the most informative feature representation in this sequence. Results form LRR and LRNN models show that in most cases the more informative feature representation gives the better performances. This observation is relevant to explain the superior strength of our models which is based on a multi-layer architecture for aspect representation with the very efficient models of representation learning (word embeddings and compositional vector models). Therefore in this comparison, our models provide the most informative feature representation, that leads to better results with much improvements.

Table 3 also shows that the Full-LRNN-ASR model gives a much better result than the LRNN-ASR model. This indicates the importance of this higher aspect representation layer in our proposed architecture. This layer is capable of

capturing the relationships between aspects (i.e. the shared information between aspects) and solving the long distance between a aspect and its sentiments, and therefore enriches information for the model.

Table 3: Experimental results and comparison for detecting aspect ratings

Feature kind	Method	Δ_{aspect}	P_{aspect}	P_{review}
Bag of words	LRR	0.752	0.341	0.621
	LRNN	0.817	0.445	0.587
Word vector averaging	LRR	0.756	0.398	0.644
	LRNN	0.753	0.459	0.641
Sentence vector averaging	LRR	0.781	0.432	0.646
	LRNN	0.770	0.465	0.645
Paragraph vector	LRR	0.747	0.424	0.658
	LRNN	0.742	0.432	0.667
	Our LRNN-ASR	**0.703**	**0.497**	**0.675**
	Our FULL-LRNN-ASR	**0.596**	**0.512**	**0.741**

In an another view, from the obtained model and we follow the model's architecture as presented in figure 3, we can easily compute the aspect ratings for each input h-review. For example, Table 4 shows the result of determining aspect ratings for the five hotels which are randomly selected. In this table the ground-truth aspect ratings are in parenthesis. We can see that the aspect rating predicted from the model are very closed to the ground-truth aspect ratings. This again confirms the effectiveness of our proposed model. It is worth to emphasize that the model which can well generate aspect ratings and overall ratings is also good for determining aspect weights.

Table 4: The aspect ratings determined for the five hotels

Hotel Name	Value	Room	Location	Cleanliness	Service
Barcelo Punta Cana	3.4(3.3)	3.1(3.0)	3.7(4.0)	3.3(3.2)	3.1(3.1)
Condado Plaza Hilton	3.3(3.3)	3.6(3.8)	3.6(4.0)	3.6(3.7)	3.4(3.5)
King George	3.5(3.6)	3.1(3.1)	4.1(4.3)	4.1(3.7)	3.7(3.8)
Astoria	3.4(3.6)	2.7(2.6)	3.9(4.5)	3.5(3.3)	3.2(3.1)
Radisson Ambassador Plaza	3.3(3.2)	3.4(3.6)	3.3(3.6)	3.6(3.7)	3.6(3.5)

8. Conclusion

In this paper, we have proposed a multi-layer architecture for representation of customers' textual opinions with objective to aspect-based sentiment analysis. We used representation learning techniques including word embeddings and compositional vector models to obtain the word, sentence, aspect-based paragraph, and higher aspect representation layers, which helps to enrich knowledge of the input through layers step by step. We then integrated these representations into a neural network and used a back-propagation algorithm based on gradient descent for training a model for aspect ratings prediction as well as generating aspect weights. Experimental results have shown that our proposed model outperforms other popular methods with much improvements. This demonstrates that applying representation learning techniques in a multi-layer architecture for sentiment analysis problem is very effective. It is also confirmed that from a training data with only given overall ratings we can efficiently derive aspect ratings as well as aspect weights.

Acknowledgement

This paper is supported by The Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.22.

References

[1] P.D. Turney, Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, Proceedings of the Association

for Computational Linguistics, 2002, pp. 417-424.

[2] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79-86.

[3] R. Mihalcea, C. Banea, J. Wiebe, Learning Multilingual Subjective Language via Cross-Lingual Projections, Proceedings of the Association for Computational Linguistics (ACL), 2007, pp. 976-983.

[4] F. Su, K. Markert, From Words to Senses: a Case Study in Subjectivity Recognition, Proceedings of Coling, Manchester, UK, 2008, pp.825-832.

[5] B. Pang, L. Lee, A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". Proceedings of the Association for Computational Linguistics (ACL), 2004, pp. 271-278.

[6] B. Pang, L. Lee, Subjectivity Detection and Opinion Identification, Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrieval, Vol. 2, No 1-2, 2008, pp. 1-135.

[7] E.P. Lim, V.A. Nguyen, N. Jindal, B. Liu, H. Lauw, Detecting Product Review Spammers using Rating Behaviors, Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 26 - 30.

[8] N. Jindal, B. Liu, Opinion Spam and Analysis, Proceedings of First ACM International Conference on Web Search and Data Mining, Stanford University, Stanford, California, USA, 2008, pp. 11-12.

[9] N. Jindal, B. Liu, Review Spam Detection, Proceedings of WWW, 2007, pp. 8-12.

[10] B. Liu, M. Hu, J. Cheng, Opinion observer: Analyzing and comparing opinions on the web, Proceedings of WWW, 2005, pp. 342-351.

[11] S. Morinaga, K. Yamanishi, K. Tateishi, T. Fukushima, Mining product reputations on the web, Proceedings of KDD, 2002, pp. 341-349.

[12] F. Li, C. Han, M. Huang, X. Zhu, Y.J. Xia, S. Zhang and H. Yu, Structure-Aware Review Mining and Summarization, Proceedings of Coling, 2010, pp. 653-661.

[13] N. Jindal, B. Liu, Identifying comparative sentences in text documents, Proceedings of SIGIR, 2006, pp. 244-251.

[14] H. Kim, C. Zhai, Generating Comparative Summaries of Contradictory Opinions in Text, Proceedings of CIKM09, 2009, pp. 385-394.

[15] M. Hu, B. Liu, Mining and summarizing customer reviews, Proceedings of SIGKDD, 2004, pp. 168-177.

[16] Y. Jo, A.H.Oh, Aspect and sentiment unification model for online review analysis, Proceedings of WSDM, 2011, pp. 815-824.

[17] Y. Wu, Q. Zhang, X. Huang, L. Wu, Phrase dependency parsing for opinion mining, Proceedings of ACL, 2009, pp. 1533-1541.

[18] B. Snyder, R. Barzilay, Multiple aspect ranking using the good grief algorithm, Proceedings of NAACL HLT, 2007, pp. 300-307.

[19] I. Titov, R. McDonald, A joint model of text and aspect ratings for sentiment summarization, Proceedings of ACL, 2008, pp. 308-316.

[20] D.H. Pham, A.C. Le, T.K.C Le, A Least Square based Model for Rating Aspects and Identifying Important Aspects on Review Text Data, Proceedings of NICS, 2015, pp. 16-18.

[21] D.H. Pham, A.C. Le, T.T.T Nguyen, Determing Aspect Ratings and Aspect Weights from Textual Reviews by Using Neural Network with Paragraph Vector Model, Proceedings of CSoNet, 2016, pp. 309-320.

[22] Z. Zha, J. Yu, J. Tang, M. Wang, T. Chua, Product Aspect Ranking and Its Applications, IEEE Transactions on Knowledge and Data Engineering, 2014, Volume 26, No.5, pp. 1211-1224.

[23] D.H. Pham, A.C. Le, A Neural Network based Model for Determining Overall Aspect Weights in Opinion Mining and Sentiment Analysis, Indian Journal of Science and Technology, Volume 9, Issue 18, 2016, pp. 1-6.

[24] H. Wang, Y. Lu, C. Zhai, Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach, Proceedings of SIGKDD, 2010, pp. 168-176.

[25] H. Wang, Y. Lu, C. Zhai, Latent Aspect Rating Analysis without Aspect Keyword Supervision, Proceedings of SIGKDD, 2011, pp. 618-626.

[26] Y. Xu, T. Lin, W. Lam, Latent Aspect Mining via Exploring Sparsity and Intrinsic Information, Proceedings of CIKM, 2014, pp. 879-888.

[27] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, Journal of Machine Learning Research, 2003, pp. 1137-1155.

[28] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, In Proceedings of Workshop at ICLR, 2013.

[29] J. Pennington, R. Socher, and C.D. Manning, Glove: Global vectors for word representation, Proceedings of EMNLP, 2014, pp. 1532-1543.

[30] A. Alghunaim and M. Mohtarami, S. Cyphers, J. Glass, A Vector Space Approach for Aspect Based Sentiment Analysis, In Proceedings of NAACL-HLT, 2015, pp. 116-122.

[31] J. Pavlopoulos and I. Androutsopoulos, Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method, In Proceedings of ACL, 2014, pp. 44-52.

[32] S. Poria , E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowledge-Based Systems 108, 2016, pp. 42-49.

[33] D. Tang, B. Qin, T. Liu, Aspect Level Sentiment Classification with Deep Memory Network. In Proceedings of EMNLP 2016, pp. 214-224.

[34] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for Aspect-level Sentiment Classification, In Proceedings of ACL, 2016, pp. 606-61.

[35] K. Moritz Hermann and P. Blunsom, Multilingual models for compositional distributed semantics, Proceedings of ACL, 2014, pp. 58-68.

[36] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, A Convolutional Neural Network for Modelling Sentences, Proceedings of ACL, 2014, pp.655-665.

[37] Y. Kim, Convolutional neural networks for sentence classification, Proceedings of EMNLP, 2014, pp. 1746-1751.

[38] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, Proceedings of HLT-NAACL, 2015, pp. 103-112.

[39] W. Yin and H. Schutze, Multichannel variable-size convolution for sentence classification. In Proceedings of the Conference on Computational Natural Language Learning, 2015, pp. 204214.

[40] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, Proceedings of ICML, 2011, pp. 513-520.

[41] R. Socher, A. Perelygin, J. Wu, J. Chuang, Ch.D. Manning, A. Ng, and C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of EMNLP, 2013, 1631-1642.

[42] Q. Le, T. Mikolov, Distributed representations of sentences and documents, Proceedings of ICML, 2014, pp. 1188-1196.

[43] D.H. Pham, A.C. Le, T.K.C Le, Learning Semantic Representations for Rating Vietnamese Comments, Proceedings of KSE, 2016, pp. 193-198.

[44] J. Mitchell and M. Lapata, Vector-based models of semantic composition. In In Proceedings of ACL, 2008, pp. 236-244.

[45] R. Socher, J. Pennington, E.H. Huang, A.Y. Ng, C. D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, Proceedings of EMNLP, 2011, pp. 151-161.

[46] R. Socher, B. Huval, C.D. Manning, A.Y. Ng, Semantic compositionality through recursive matrix-vector spaces, Proceedings of EMNLP-CoNLL, 2012, pp. 1201-1211.

[47] K.M. Hermann, P. Blunsom, The Role of Syntax in Vector Space Models of Compositional Semantics, Proceedings of ACL, 2013, pp.894-904.

[48] M. Tsubaki, K. Duh, M. Shimbo, Y. Matsumoto, Modeling and Learning Semantic Co-Compositionality through Prototype Projections and Neural Networks, In the Conference on Empirical Methods in Natural Language Processing, 2013, pp. 130-140.

[49] D. Tang, B. Qin, T. Liu, User Modeling with Neural Network for Review Rating Prediction, In Proceeding of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 1340-1346.

[50] L. Bottou, Stochastic Learning, Advanced Lectures on Machine Learning, 2003, pp. 146-168.

[51] A. Cotter, O. Shamir, N. Srebro, K. Sridharan, Better Mini-Batch Algorithms via Accelerated Gradient Methods, Part of: Advances in Neural Information Processing Systems 24 (NIPS), 2011, pp. 1-9.

[52] K. Toutanova, D. Klein, Ch.D. Manning, and Y. Singer, Feature-rich part-ofspeech tagging with a cyclic dependency network. Proceedings of the Conference of the North American Chapter of the Association for

Computational Linguistics on Human Language Technology, 2003, pp. 173-180.

Appendix A. Appendix

The gradient of $E(\theta)$ according to $\hat{O}_d$ is computed by:

$$\frac{\partial E(\theta)}{\partial \hat{O}_d} = -(\frac{O_d}{\hat{O}_d} - \frac{1-O_d}{1-\hat{O}_d}) \tag{A.1}$$

The gradient of $E(\theta)$ according to $\hat{\alpha}_{di}$ is, $\frac{\partial E(\theta)}{\partial \hat{\alpha}_{di}} = \frac{\partial E(\theta)}{\partial \hat{O}_d} \cdot \frac{\partial \hat{O}_d}{\partial \hat{\alpha}_{di}}$

$$= \frac{\partial E(\theta)}{\partial \hat{O}_d}(\sum_{l=1}^{k} \delta(\mathrm{i}=\mathrm{l})\alpha_{di}(1-\alpha_{di})r_i - \sum_{l=1}^{k} \delta(\mathrm{i} \neq l)\alpha_{di}\alpha_{dl}r_{dl}) \tag{A.2}$$

where $\delta(y) = \begin{cases} 1; \text{ if } y = true \\ 0; \text{ if } y = false \end{cases}$

The gradient of $E(\theta)$ according to $\mathrm{w_i}$ is, $\frac{\partial E(\theta)}{\partial \mathrm{w}_i} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial \hat{O}_d} \cdot \frac{\partial \hat{O}_d}{\partial r_{di}} \cdot \frac{\partial r_{di}}{\partial \mathrm{w}_i}$

$$= \sum_{d=1}^{|D|} (\frac{O_d}{\hat{O}_d} - \frac{1-O_d}{1-\hat{O}_d}).\alpha_{di}.r_{di}(1-r_{di}).\begin{Bmatrix} x^*_{di}; \\ 1; (i=0) \end{Bmatrix} + \lambda \mathrm{w}_i \tag{A.3}$$

The gradient of $E(\theta)$ according to $\mathrm{V_i}$ is,

$$\frac{\partial E(\theta)}{\partial \mathrm{V_i}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial \hat{O}_d} \cdot \sum_{t=1}^{k} (\delta(\mathrm{i}=\mathrm{t}) \frac{\partial \hat{O}_d}{\partial r_{di}} \frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}} \cdot \frac{\partial \mathrm{x}^*_{di}}{\partial \mathrm{V_i}} + \delta(\mathrm{i} \neq \mathrm{t}) \frac{\partial \hat{O}_d}{\partial r_{dt}} \frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}} \cdot \frac{\partial \mathrm{x}^*_{dt}}{\partial \mathrm{V_i}}) \tag{A.4}$$

in which, $\frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}} \cdot \frac{\partial \mathrm{x}^*_{dil}}{\partial \mathrm{V_i}} = \frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}} . \beta . \frac{\partial \mathrm{x}_{di}}{\partial \mathrm{V_i}}$,
$\frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}} \cdot \frac{\partial \mathrm{x}^*_{dt}}{\partial \mathrm{V_i}} = \frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}} \cdot \frac{\gamma}{k-1} \cdot \frac{\partial \mathrm{x}_{di}}{\partial \mathrm{V_i}}$
$\frac{\partial \mathrm{x}_{di}}{\partial \mathrm{V_i}} = \sum_{j=1}^{p-1} (1 - f(y) * f(y))[v(s_{dij}) + v(s_{di(j+1)})]$
The gradient of $E(\theta)$ according to $\mathrm{v_{i0}}$ is,

$$\frac{\partial E(\theta)}{\partial \mathrm{v_{i0}}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial \hat{O}_d} \cdot \sum_{t=1}^{k} (\delta(\mathrm{i}=\mathrm{t}) \frac{\partial \hat{O}_d}{\partial r_{di}} \frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}} \cdot \frac{\partial \mathrm{x}^*_{di}}{\partial \mathrm{v_{i0}}} + \delta(\mathrm{i} \neq \mathrm{t}) \frac{\partial \hat{O}_d}{\partial r_{dt}} \frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}} \cdot \frac{\partial \mathrm{x}^*_{dt}}{\partial \mathrm{v_{i0}}}) \tag{A.5}$$

in which, $\frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}}.\frac{\partial \mathrm{x}^*_{dil}}{\partial \mathrm{v_{i0}}} = \frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}}.\beta.\frac{\partial \mathrm{x}_{di}}{\partial \mathrm{v_{i0}}}$,

$\frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}}.\frac{\partial \mathrm{x}^*_{dt}}{\partial \mathrm{v_{i0}}} = \frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}}.\frac{\gamma}{k-1}.\frac{\partial \mathrm{x}_{di}}{\partial \mathrm{v_{i0}}}$

$\frac{\partial \mathrm{x}_{di}}{\partial \mathrm{v_{i0}}} = \sum_{j=1}^{p-1}(1 - f(y) * f(y))$

where $f(y) = tanh(y)$, $y = \mathrm{V}_i.[v(s_{dij}) + v(s_{di(j+1)})] + \mathrm{v}_{i0}$

The gradient of $E(\theta)$ according to $\mathrm{U_i}$ is,

$$\frac{\partial E(\theta)}{\partial \mathrm{U_i}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial \hat{O}_d}.\sum_{t=1}^{k}(\delta(\mathrm{i} = \mathrm{t})\frac{\partial \hat{O}_d}{\partial r_{di}}\frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}}.\beta.C + \delta(\mathrm{i} \neq \mathrm{t})\frac{\partial \hat{O}_d}{\partial r_{dt}}\frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}}.\frac{\gamma}{k-1}.C) \tag{A.6}$$

in which $C = \sum_{j=1}^{p-1}(\frac{\partial \mathrm{x}_{di}}{\partial v(s)_{dij}}.\frac{\partial v(s)_{dij}}{\partial \mathrm{U_i}} + \frac{\partial \mathrm{x}_{di}}{\partial v(s)_{di(j+1)}}.\frac{\partial v(s)_{di(j+1)}}{\partial \mathrm{U_i}})$

$$= \sum_{j=1}^{p-1}[\sum_{l=1}^{q-1}(1 - f(z_{dij}) * f(z_{dij}))(e_{dijl} + e_{dij(l+1)})$$

$$+ \sum_{l=1}^{q-1}(1 - f(z_{di(j+1)}) * f(z_{di(j+1)}))(e_{di(j+1)l} + e_{di(j+1)(l+1)})].\mathrm{V_i} \tag{A.7}$$

The gradient of $E(\theta)$ according to $\mathrm{u_{i0}}$ is,

$$\frac{\partial E(\theta)}{\partial \mathrm{u_{i0}}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial \hat{O}_d}.\sum_{t=1}^{k}(\delta(\mathrm{i} = \mathrm{t})\frac{\partial \hat{O}_d}{\partial r_{di}}\frac{\partial r_{di}}{\partial \mathrm{x}^*_{di}}.\beta.D + \delta(\mathrm{i} \neq \mathrm{t})\frac{\partial \hat{O}_d}{\partial r_{dt}}\frac{\partial r_{dt}}{\partial \mathrm{x}^*_{dt}}.\frac{\gamma}{k-1}.D) \tag{A.8}$$

in which $D = \sum_{j=1}^{p-1}(\frac{\partial \mathrm{x}_{di}}{\partial v(s)_{dij}}.\frac{\partial v(s)_{dij}}{\partial \mathrm{u_{i0}}} + \frac{\partial \mathrm{x}_{di}}{\partial v(s)_{di(j+1)}}.\frac{\partial v(s)_{di(j+1)}}{\partial \mathrm{u_{i0}}})$

$$= \sum_{j=1}^{p-1}[\sum_{l=1}^{q-1}(1 - f(z_{dij}) * f(z_{dij}))$$

$$+ \sum_{l=1}^{q-1}(1 - f(z_{di(j+1)}) * f(z_{di(j+1)})).\mathrm{V_i} \tag{A.9}$$

where $f(z_{dij}) = tanh(z_{dij})$, $z_{dij} = \mathrm{U}_i.[(e_{dijl} + e_{dij(l+1)}) + \mathrm{u}_{i0}]$,

$f(z_{di(j+1)}) = tanh(z_{di(j+1)})$, $z_{di(j+1)} = \mathrm{U}_i.[(e_{di(j+1)l} + e_{di(j+1)(l+1)}) + \mathrm{u}_{i0}]$