Sentence Vector Representation Methods for Aspect Category Detection

Noman Dilawar, Hammad Majeed
Department of Computer Science
National University of Computer and Emerging Sciences
Email: {noumandilawar@gmail.com, hammad.majeed@nu.edu.pk}

Abstract—**Online user reviews plays a vital role in making important business decisions. Aspect category detection from a review sentence *(e.g. "food#quality" and "food#price" in restaurant reviews)* is one of the major tasks in opinion mining. Given a predefined set of aspect categories, previous approaches have used hand crafted features, word embedding based sentence vector features and a classification algorithm. These approaches are focused on feature extraction which adds a complexity in performing this task. In this paper, we propose multiple methods to represent a sentence vector using algebraic combination of its word vectors. Feature vector is then directly used to train a multi-layer neural network for multi-label sentence classification algorithm. The proposed model is tested on a real world online challenge (SE-ABSA 2016). The experimental results of our approach has shown the highest F1 scores and outperforms the existing approaches.**

I. Introduction

Sentiment analysis or opinion mining is a computational method for automatically detecting the attitude, emotion and sentiment of a speaker in a given piece of text [1]. The simplest form of the sentiment analysis classifies the given review, paragraph or a sentence as *positive* or *negative*[2][3]. This type of analysis is incapable of handling conflicting sentiments within a text. For example, a simple sentiment analysis approach for the sentence "The biryani is delicious but expensive" would annotate it with the label "conflicting". This is due to the negative and positive sentiments about the food *(biryani)* at the same time. But careful examination would reveal that this sentence is expressing a positive sentiment for the food *(biryani)* in the perspective of its taste *(delicious)* and a negative sentiment about its price *(expensive)*. Such examples prove that the simple sentiment analysis does not provide an in-depth information about the sentiments and a detailed sentiment analysis is required to capture multi-dimensions of the opinionated text content [4].

In 2004, a framework was proposed to extract feature-based (or aspect-based) summaries from customer reviews [5]. It works by decomposing conventional sentiment analysis into three subtasks: (1) Product features extraction (i.e. identification of the features towards, the customers have expressed their opinions) (2) Assigning sentiments to product features (3) Generating summaries on the basis of extracted information. This type of feature based method for Sentiment Analysis is known as **A**spect-**B**ased **S**entiment **A**nalysis (ABSA).

Aspect **C**ategory **D**etection (ACD) is one of the important task in ABSA, which identifies the aspect categories from customer reviews. These categories are often predefined, which makes it a multi-label classification task. For example in SemEval-2016, category/ies is assigned from a set of predefined set of **E**ntity (e.g. restaurant, food, laptop) and **A**ttribute (e.g. design, quality, price) pairs (**E#A**). In the sentence "The biryani is delicious but expensive", "`food#prices` and `food#quality`" should be detected as the aspect categories. Opinions without knowing its target is of limited use [6]. Obtained aspect categories are associated with their sentiment polarities to generate opinionated aspect based summaries as shown in TABLE I.

TABLE I. ABSA identiies the aspects of the entities and the sentiments expressed for each.

Opinion	Category	Polarity
Easy to use and great for online gaming	ease of use	positive
Liking the graphics, quality, speed	picture/video	positive
All in all, a greate value!!!	value	positive
Great performance with nice design and fun	design/style	positive
... that stupid light drains the battery so fast	battery	negative
The controls are great	controls	positive

In past several approaches have been proposed to address this task and the most common one is SVM classification [7][8]. These methods rely on syntactic features of sentences and less focused towards semantic relationships among words. Modern methods for doing this task use word embedding [18][19] to represent words as vector features. These word vectors are usually combined to form a sentence vector. In [9] a representation learning approach is proposed to capture semantic relationships of words using semi-supervised word embedding algorithm. Obtained word vectors of a sentence are averaged to represent its continous feature vector. The sentence vector is then used to train logistic regression model for aspect category detection. Another

approach [10] represent sentence as a matrix of its word vectors. Sentence matrix is then used to extract more enhanced features by using deep convolutional neural networks. These extracted features are then used to train binary classifiers for each aspect category using one-vs-all strategy. These methods require feature extraction process to train learning models for aspect category detection. Moreover, sentence representation methods as a combination of its word vectors is not well studied. To overcome these limitations, we proposed a study on representing sentences as a combination of its word vectors. Effectiveness of our sentence vector representations are evaluated on a benchmark dataset.

This paper is focused on detecting aspect categories from English restaurant reviews. It does that by improving sentence vector representation on top of *word2vec* model's word vector features. Proposed approach is very simple and computationally inexpensive. A sentence vector is formed by combining its word vectors using different algebraic operations (e.g sum, multiplication, division). Each algebra operation on a sentence word vectors provides a distinct feature vector representations. Therefore, multiple sentence feature vectors can be obtained for a single sentence. The goal behind this activity is to find a best feature vector representation that can act as an input fecture to a classifier. By using this approach, we are able to improve upon the best results reported for the real world challenging problems. The rest of this article is organized as follows: We describe aspect-based sentiment analysis in Section II; Review of related work is in Section III; our proposed approach is discussed in Section IV; task and datasets used in our experiments are in Section V; experimental setup is provided in Section VI; results and discussions are reported in Section VII; Limitations of our work is presented in Section VIII. finally, Section IX concludes the paper and outlines future research directions.

II. Related Work

A. Aspect Based Sentiment Analysis

Aspect-based sentiment analysis is a powerful opinon mining technique. Opinions are actually expressed towards entities (e.g. Restaurant) and their aspects (e.g. food). The goal of Aspect-Based sentiment analysis is to find aspects and their associated sentiments [6][11].

Aspect detection task from online user reviews is well examined since the initial work of [5]. In recent years, topic modeling approaches have been used extensively for this task. Such methods detect ratable aspects from online user reviews and cluster them into their corresponding categories. In [12], multi-grained topic models are presented by extending [13][14]. This model is able to extract aspects and their categories with high accuracy. Unlike previous topic modeling approaches, [15][16] focused on simultaneously extracting both aspects and their associated opinions.

Aspect category detection (ACD) is a subtask of aspect-based sentiment analysis. Aspect categories are coarser than aspects. Given a set of predefined aspect categories the goal is to assign one or more aspect categories to a review sentence. In previous years, support vector machines [17] were the most popular ones for doing this task. In [8] one-vs-all support vector machine (SVM) classifiers are trained on manually annotated restuarant reviews dataset. In their work, they directly used stem words as training features. Same classfier is trained by extracting multiple syntactic features from sentences [7]. Furthermore, a word list is constructed to improve their predictions.

B. Continuous sentence vectors for aspect category and sentence classification

Word embedding algorithms [18][19] for continuous vector representation of words, captures a great sense of semantic and syntactic information. These word vectors are then used to obtain the continuous representation of sentence vectors. Sentence vectors act as an input feature for a classifier. In [9], a semi-supervised word embedding algorithm is used to obtain word vectors. Sentence vectors are obtained by averaging the obtained word vector in a sentence [20]. Acquired vectors are then used to extract more deeper and hybrid features to train logistic regression classifier for predicting aspect categories. Their approach achieves the best scores in SemEval-2014 aspect category detection task. In [23], *sentence-matrix* is defined to represent sentences. Each word in the sentence is represented by a fixed length vector obtained from a pre-trained word2vec model [18]. The columns of the sentence-matrix are of fixed length and represents the vector representation. The rows of the matrix represents the words in the sentences. Zeros are padded in the rows in case of shorter sentences. This is done to make the matrices compatible for algebraic operations. A convolution neural network (CNN) model is trained on this matrix for sentence classification. Same sentence matrix is used to extract more enhanced features by using deep convolutional neural networks [10]. On top of these features, one-vs-all strategy is used to train single layer feed-forward binary classifiers for each training aspect category. Their system achieves the best scores in aspect category detection task in SemEval-2016 [21] on English restaurant reviews dataset. In [18], the sum of the word vectors is used to represent short sentences or phrases. In [25], the word vectors of all the words in the sentence are obtained and then averaged to represent the sentence. As a minor modification to this representation, a *Normalized Average Vector (NAV)* method was proposed in [26] to train SVM for aspect category detection. As the name suggests this method adds normalized word vectors.

III. Proposed Methodology

The proposed methodology starts with removing the stop words and symbols. Then the remaining words are

transformed into vector representation using word2vec. The obtained word vectors are combined into a fixed size vector that represents the given sentence. The sentence vector is passed to a Multi-layer Neural Network to train the model. The graphical representation of the system architecture is shown in Fig. 1.

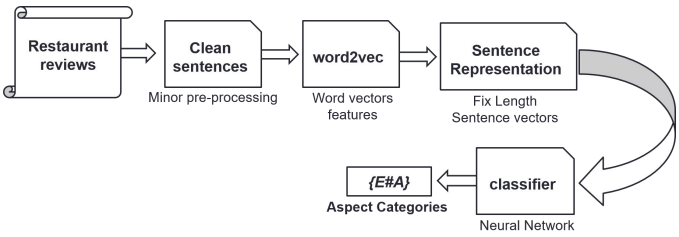

Fig. 1. System Architecture of Aspect Category Detection

A. Sentence Representation

The word vectors of the given sentence are combined to form its vector representation by using simple arithmetic operations (subtract, average, sum). The combining operation is performed on the normalized and unnormalized word vector representations. These operations are discussed in the detail in the following sections.

1) Normalized Representation of Sentence Vector: In this category, normalization techniques for representing a sentence as feature vector are discussed. Three different approaches for defining a sentence vector are proposed. In which w_i is a word vector of i^{th} word and n is the the count of words in the given sentence.

- **Averaging Difference of Word Vectors**

$$Avg\_Sub = \frac{\vec{w_1} - \sum_{i=2}^{n} \vec{w_i}}{n} \tag{1}$$

- **L1-Normalized Sum of Average Word Vectors**

$$L1 - AvgSOW = \frac{1/n \sum_{i=1}^{n} \vec{w_i}}{||1/n \sum_{i=1}^{n} \vec{w_i}||} \tag{2}$$

- **L1-Normalized Difference of Average Word Vectors**

$$L1 - AvgDOW = \frac{1/n(w_1 - \sum_{i=2}^{n} \vec{w_i})}{||1/n(w_1 - \sum_{i=2}^{n} \vec{w_i})||} \tag{3}$$

- **L2-Normalized Average Difference of Word Vectors**

$$L2 - AvgDOW = \frac{1/n(w_1 - \sum_{i=2}^{n} \vec{w_i})}{||1/n(w_1 - \sum_{i=2}^{n} \vec{w_i})||^2} \tag{4}$$

- **L1-Normalized Sum of Word Vectors**

$$L1 - SOW = \frac{\sum_{i=1}^{n} \vec{w_i}}{||\sum_{i=1}^{n} \vec{w_i}||} \tag{5}$$

- **L2-Normalized Sum of Word Vectors**

$$L2 - SOW = \frac{\sum_{i=1}^{n} \vec{w_i}}{||\sum_{i=1}^{n} \vec{w_i}||^2} \tag{6}$$

- **L1-Normalized Difference of Word Vectors**

$$L1 - DOW = \frac{\vec{w_1} - \sum_{i=2}^{n} \vec{w_i}}{||\vec{w_1} - \sum_{i=2}^{n} \vec{w_i}||} \tag{7}$$

- **L2-Normalized Difference of Word Vectors**

$$L2 - DOW = \frac{\vec{w_1} - \sum_{i=2}^{n} \vec{w_i}}{||\vec{w_1} - \sum_{i=2}^{n} \vec{w_i}||^2} \tag{8}$$

2) Un-normalized Representation of Sentence Vector: In this method, vector representation of sentences are obtained by omitting the normalization step. It is called unnormalized representation of a sentence vector. It looks like a much simple way of representing sentence features using word vectors.

- **Difference of Word Vectors**

$$DOW = \vec{w_1} - \sum_{i=2}^{n} \vec{w_i} \tag{9}$$

- **Concatenation of Sum and Difference of Word Vectors**

$$SOW = \vec{w_1} + \sum_{i=2}^{n} \vec{w_i} \tag{10}$$
$$SOW \oplus DOW$$

The motivation behind this research is to find a simple and fast method to represent a sentence vector by using word vectors. Performance of the proposed sentence representations are tested on real world challenging problems.

In this study Aspect Category Detection problem was transformed into a multi-label multi-class sentence classification problem to test the performance of the representations proposed in Section III-A. The goal was to assign single/multiple category(ies) to the given input sentence vector $\vec{x}$. The output of the classification algorithm was a vector $\vec{y}$ with size equal to the number of the predefined categories. $\vec{y}$ has one or more classes enabled for the given sentence. Rectified Linear Unit (ReLU) feed forward multi-layer neural network [27][28] was used as a classifier.

Input layer of the neural network takes a sentence feature vector as an input. The number of neurons to represent this layer must be equal to length of input vector. For this study two hidden layers were used. The purpose of using two layers is to extract enough features to learn the weights from training examples. Hidden layer pass input features through non-linearities to perform linear transformations on the input feature vectors. Rectified Linear Unit (ReLU) [28] was used as an activation function. ReLU is preferred because 1) It has low computational cost as compared to `sigmoid/tanh` functions as it doesnt require expensive operations like, calculating exponential. 2) ReLU has fast convergence rate on stochastic gradient decent as compared to `sigmoid` and `tanh` functions.

The softmax regression was used as a cost function also shown in (11).

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{11}$$

Simplifying it we get:

$$L_i = -f_{y_i} + \log\left(\sum_j e^{f_j}\right) \tag{12}$$

Softmax Regression (or multinomial linear regression) is a generalization of logistic regression. In logistic regression, only binary labels: $y(i) \in \{0,1\}$ are allowed. While Softmax Regression allows more than two classes, $y(i) \in \{1,\ldots,k\}$, k is the number of classes and y is the output vector.

To prevent neural network from over-fitting, L2-regularization was used. L2-regularization is the one of the ways to control over-fitting in neural networks. It penalizes the squared magnitude of all the neural network parameters except bias inputs and then add it in the objective function as shown in (13).

$$R(W) = \frac{1}{2}\lambda|w|^2 \tag{13}$$

λ is a regularization controlling factor to penalize the weights. The final objective function became $L_i + R(W)$ by using regularization.

Weights of the network are randomly initialized by Gaussian distribution with standard deviation of $\sqrt{2/n}$, where n is the number of inputs to the neuron layer.

A stochastic method, Adam optimizer [29] was used to optimize the network weights. Mathematically optimizer can be written as:

$$(m_t)_i = \beta_1(m_{t-1}) + (1-\beta_1)(\Delta L(W_t))_i \tag{14}$$

$$(v_t)_i = \beta_2(v_{t-1})_i + (1-\beta_2)(\Delta L(W_t))_i^2 \tag{15}$$

$$(W_{t+1})_i = (W_t)_i - \alpha\frac{\sqrt{1-(\beta_2)_i^t}}{1-(\beta_1)_i^t}\frac{(m_t)_i}{\sqrt{(v_t)_i}+\epsilon} \tag{16}$$

Hyper parameter called learning rate (α) was used to control the step size during each update of weights. An exponential decay method was used to update the net weights. This parameter automatically slows down the learning rate with the increase in the size of the epochs.

A score function is used that takes three parameters that are: result vector $x_i \in R^D$, weight matrix $W \in [K \times D]$ and bias $b[1 \times K]$ as an input and return scores for all classes $y_i \in [1 \times K]$ as shown in (17) and (18).

$$\sigma(x) = 1/\left(1+e^{(-x)}\right) \tag{17}$$

$$f(x_i, W, b) = \sigma(Wx_i + b) \tag{18}$$

Where, examples x_i varies from $i = \{1,\ldots,n\}$, $yi \in \{1,\ldots,K\}$, D is the dimensions of input vector and K is the number of classes. The use of sigmoid function ensures that the class scores are normalized between the range of $[1,0]$. Example of a score function is shown in Fig. 2.

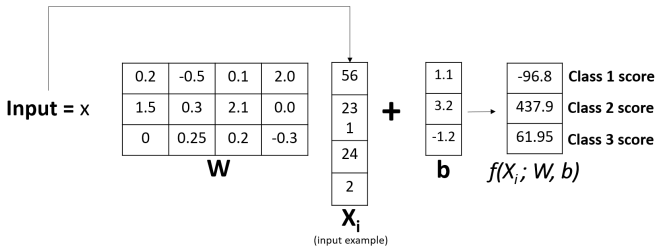

Fig. 2. Example of score function.

IV. Task and Dataset

This study targets the problem of Aspect Category Detection in Sentence-level Aspect-Based Sentiment Analysis. Our goal is to extract all the aspect categories present in the input sentences. We modeled and formulated this problem as a multi-label multi-class sentence classification problem. The task is well explained in the previous sections, therefore we emphasize on the details regrading dataset, experimentation and results in this section.

A. English Restaurant Reviews Dataset for SE-ABSA-2016

We have used English restaurant reviews dataset[1] provided in the SE-ABSA 2016 [21]. Each sentence in Restaurant reviews dataset is annotated with aspect terms (e.g. "pizza", "fish", "food", "restaurant") and those aspect terms are assigned/labeled to their aspect categories (e.g. "FOOD", "QUALITY") with polarities (e.g. "Negative","Positive"). English restaurant reviews dataset contains 2000 training and 676 test sentences. In these sentences there are 1708 training and 587 test sentences labeled with aspect categories and remaining sentences are labeled with *outOfScope* or *None* categories. Sentences with *outOfScope* or *None* categories were not used in the final evaluation of the *Aspect Category Detection* results in SemEval-2016 Task 5. Each restaurant review consists of multiple sentences annotated with their respective aspect categories. Aspect categories is the combination of *Attribute* and *Entity* (E#A) pairs as discussed before. Sample of a single customer review from restaurant dataset is shown below in Fig. 3.

Aspect categories are unevenly distributed across the training and test review sentences therefore, the dataset is highly unbalanced. It can effect the training and prediction accuracies, because high occurrence of a specific class may dominate in the results over less occurring classes. The distribution of aspect categories is shown in Fig. 4. There are total 2258 aspect categories in training

[1] Available at: http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools

```
<Review rid="1727363">
  <sentences>
    <sentence id="1727363:0">
      <text>i actually feel like i should keep it a secret.</text>
    </sentence>
    <sentence id="1727363:1">
      <text>
        This is a wonderful place on all stand points especially value ofr money.
      </text>
      <Opinions>
        <Opinion target="place" category="RESTAURANT#PRICES" polarity="positive"
        from="20" to="25"/>
        <Opinion target="place" category="RESTAURANT#GENERAL" polarity="positive"
        from="20" to="25"/>
      </Opinions>
    </sentence>
    <sentence id="1727363:2">
      <text>An excellent service</text>
      <Opinions>
        <Opinion target="service" category="SERVICE#GENERAL" polarity="positive"
        from="13" to="20"/>
      </Opinions>
    </sentence>
  </sentences>
</Review>
```

Fig. 3. Annotation of sentences in a single training review instance in English Restaurant reviews dataset

and 743 aspect categories in test sets. We have disregarded the repetition for the same categories of a single sentence and consider their counts as one as discussed in SE-ABSA 2015 [30] and SE-ABSA 2016 [21] .

Categories	Training	Test
FOOD#QUALITY	681	226
SERVICE#GENERAL	419	145
RESTAURANT#GENERAL	421	142
AMBIENCE#GENERAL	226	57
FOOD#STYLE_OPTIONS	128	48
RESTAURANT#MISCELLANEOUS	97	33
FOOD#PRICES	82	22
RESTAURANT#PRICES	80	21
DRINKS#QUALITY	46	21
DRINKS#STYLE_OPTIONS	30	12
LOCATION#GENERAL	28	13
DRINKS#PRICES	20	3
Total: 12	Total: 2258	Total: 743

Fig. 4. Table of aspect categories distribution across training and test sentences

B. Parameters and preparation of dataset

In SE-ABSA 2016 task 5, **A**spect **C**ategory **D**etection systems are divided into two categories. One is constrained *(C)* and second is unconstrained *(U)* systems. In constrained systems, no external training dataset is allowed during training where in unconstrained systems, it is allowed to use datasets from outside sources *(e.g. Yelp, Amazon)*. We are working in the category of unconstrained systems.

We have used distributed representations of words to build dense sentence vectors. For this, we have used skip-gram approach *(as discussed in section 2)* to train *word2vec* model [18]. English restaurant reviews dataset for SE-ABSA-2016 task 5 contains only 2000 training sentences. This amount of text is not enough to efficiently train a *word2vec* model. It is important to incorporate only domain specific information during the training of *word2vec* model. Consequently, we ended of using *Yelp* restaurant reviews dataset[2] to train our model. Effect of domain specific word vectors is explained in [26] [23]. *Yelp* restaurant reviews contains 131,778 unique words and about 200 million tokens with 2225213 sentences. We have used 2000 sentences from our training set and additionally first 5,00,000 sentences from Yelp restaurant reviews to train *word2vec* model. It is important to define here that the challenge allowed the participants to use dataset other than the provided training dataset. Many other participants also used this dataset.

V. Experimental setup

First restaurant review sentences are passed through a pre-processing stage. At this stage stream of tokens are generated from sentences and stopwords are removed. English restaurant review sentences from *Yelp* and *SE-ABSA 2016 Task 5* were combined to train *word2vec* model. *Word2vec* model is trained using *Gensim* [31] library in python. Each sentence vector $\vec{x}$ in English restaurant reviews dataset belongs to the multiple categories that can be interpreted as an output vector $\vec{y}$. Furthermore, multiple one hot encoded scheme was used to represent multiple categories or classes in $\vec{y}$. The dimensions of vector $\vec{y}$ is fixed and equal to the predefined set of twelve classes $[12 \times 1]$.

Aspect **C**ategory **D**etection problem is a classical machine learning problem, where the goal is to predict the output labels $\vec{y}$ for a given input $\vec{x}$. A predictive model based on multi-layer *neural network* is implemented using *Tensorflow* [32]. Softmax regression as a cost function is used to train the *neural network* model along with the sigmoid function on the output layer to return the output scores. We have tuned a single *threshold* ($\tau = 0.785$) for conducting all of our experiments. We always consider a predicted class, if its score becomes greater than the decided threshold and note that in this case we can also have multiple categories as well. Tuning of training hyper-parameters of *word2vec* and *neural network* models are discussed in the later sections.

Word2vec (skip-gram) model was trained on 500,000 restaurant review sentences. These sentences are obtained from *Yelp* and *SE-ABSA 2016 Task 5* datasets. Effective training of our model rely on the tuning of different hyper parameters *(e.g. context window, min word count)*. These parameters can highly influence the quality of word vectors. Five parameters are used to control the training of *word2vec (skip-gram)* model and their best values were chosen by hit and try. Moreover, the summary of the values of parameters is mentioned in TABLE II.

1) Dimension (D): It is used to control the size of word embeddings or vector. The size chosen for

[2]This dataset can be found at: http://www.yelp.com/dataset_challenge

each word vector during training is D=400. So, each $\vec{w}$ is equals to the dimensions of $[1 \times D]$.

2) Minimum word count: In large text corpora there are always some words that are not very frequent and less meaningful. So, this parameter controls the minimum number of word counts that should be allowed for a word to be considered during the training (or learning) process. We are using min word count equal to 1 because we already have small training dataset and we want to have word vectors for the maximum number of words.
3) Context: Context is actually the size of a window around each word. We are using context size of five words. It means for any centered word w_0 in a context size of five, we always have its left w_{-1}, w_{-2} and right w_1,w_2 context. It is one of the most important parameter of *word2vec* model.
4) Down Sampling: This parameter is used to control the most frequent words in the training text corpora. The most common range of down sampling lies between [$1e^{-5}$, $1e^{-3}$].
5) Number of workers: This is the number of threads that we can manage to run on the machine for achieving the desired parallelism during the training process.

TABLE II. WORD2VEC MODEL TUNED PARAMETERS

Parameter	Value
word size	400
Min word count	1
Number of worker	5
Context	5
Down sampling	e^{-3}

We don't have all the word vectors against each word that falls under the domain of restaurant reviews. To fix the missing word vectors, we replaced them with zeroes equal to the dimensions of the existing word vectors $\vec{w}$.

A. Experimental setup

We have used two layer neural network model for **A**spect **C**ategory **D**etection. Vector representation of sentences $\vec{s}$ and output labels $\vec{y}$ were used to train our neural network model. All of the proposed sentence representation techniques were applied incrementally to train multiple models and then each is evaluated on the test dataset. The dataset was divided into training and validation sets with the ratio of 85 and 15. The neural network training parameters are given in table TABLE III.

we have applied adaptive learning rate that gradually slow down the step size to achieve fast and optimal convergence. The adaptive learning rate of the *neural network* was controlled by decay rate, which gradually reduce the learning rate by some factor depending on the number of epochs. L2 regularization was used to control over fitting of the model. Each epoch consists of multiple batches across the whole dataset instances. Batch size is fixed, and contains 80 examples in each set of training examples. Before starting each epoch, training dataset is shuffled. Loss and accuracy is observed after every 5 epochs as shown in Fig. 5.

TABLE III. NEURAL NETWORK MODEL TUNED PARAMETERS

Parameter	Value
Epochs	100
Hidden layer size	$\mathtt{layer\ 1^{st}}: 300, \mathtt{layer\ 2^{nd}}: 250$
Batch size	80
Base learning rate	0.002
Decay rate	0.96
Regularization	0.003

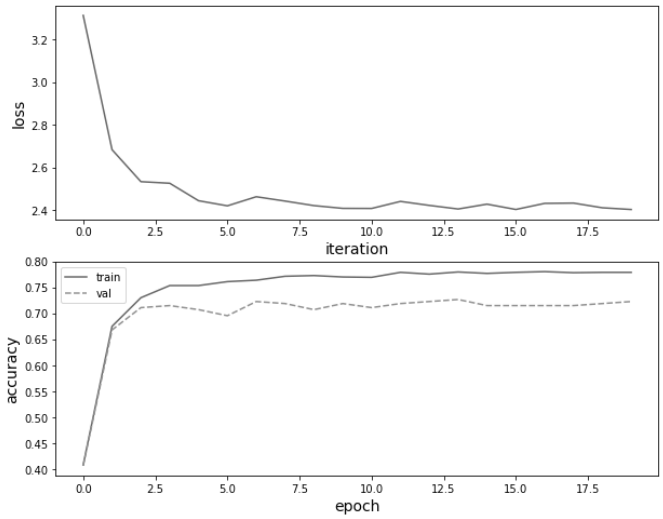

Fig. 5. Loss and Training plots

Aspect **C**ategory **D**etection model is evaluated by computing F1 *(micro averaging)* scores based on the ratio between correctly classified labels and the set of predictions and the gold standard as discussed in SemEval-2016 task 5 [21]. F1 scores are calculated under the known definitions of precision (p), recall (r) which are.

$$F1 = 2\frac{pr}{p+r} \quad \text{where} \quad p = \frac{tp}{tp+fp}, \quad r = \frac{tp}{tp+fn}$$

The max F1 scores for **A**spect **C**ategory **D**etection task is 73.031% [10] in SemEval-2016 task 5 for restaurant domain. We have compared our experiment results with the current best scores. Previous evaluation scores are available online[3] of SemEval-2016 Task 5.

VI. RESULTS AND DISCUSSIONS

Primary focus of our experiments was to study the effect of Normalized and Un-Normalized sentence vector representations for **A**spect **C**ategory **D**etection task.

[3]Results of SemEval-2016 Task 5 are available here: http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools

1) Normalized representation of sentence vectors: In normalized sentence vector representation methods, we have used different vector normalization techniques for generating sentence vectors on the top of word vectors. We trained *neural network* model on proposed normalized sentence representation methods and evaluated the results on test dataset. In our first experiment, we started with *Avg-SOW* and *Avg-DOW* methods to represent the sentences. We achieved pretty decent scores, but less than the existing best scores. In second experiment, *L1-AvgSOW*, *L1-AvgDOW*, *L2-AvgDOW* and *L2-AvgSOW* sentence representation methods was used. We have outperformed this task and secured highest F1 scores in **A**spect **C**ategory **D**etection as compared to the best scores. In third experiment, *L1-SOW*, *L1-DOW*, *L2-SOW* and *L2-DOW* methods are applied to represent sentences. This time our system outperformed the results reported in second experiment and eventually the best scores in **A**spect **C**ategory **D**etection task. The experimentation results of our proposed normalized methods for generating sentence vectors are mentioned in TABLE IV. *(The bold rows in the following table are the proposed methods)*

TABLE IV. RESULTS OF NORMALIZED REPRESENTATION OF SENTENCE VECTORS

Method	Precision	Recall	F1 Scores
Avg-SUM	66.66	77.79	71.80
Avg-DOW	62.88	79.81	**70.34**
L1-AvgSOW	74.62	73.21	**73.91**
L2-AvgSOW	65.82	77.52	71.19
L1-AvgDOW	75.20	73.88	**74.54**
L2-AvgDOW	65.74	80.08	**72.20**
L1-SOW	77.41	72.40	**74.82**
L2-SOW	69.65	78.46	**73.79**
L1-DOW	77.46	73.08	**75.20**
L2-DOW	67.88	77.65	**72.44**

2) Un-Normalized representation of sentence vectors: In un-normalized representation of sentence vectors, we combine all word vectors of a sentence using any arithmetic operator (e.g. addition, subtraction) without applying any normalization. It is a much simple and faster way of representing a sentence. In first experiment *SOW* and *DOW* sentence representation methods outperformed as compared to the previous discussed approaches. And in second experiment *SOW⊕DOW* method has also outperformed in the **A**spect **C**ategory **D**etection problem slot 1 ABSA task 5 but achieved less scores than *SOW*. Results are shown in table V *(The bold rows in the following table are the proposed methods)*

TABLE V. RESULTS OF UN-NORMALIZED REPRESENTATION OF SENTENCE VECTORS

Method	Precision	Recall	F1 Scores
SOW	**77.04**	**75.90**	**76.40**
DOW	76.26	74.83	75.54
SOW⊕DOW	75.09	76.31	75.70

3) Discussion: Our experimental studies have shown some interesting results and many of our proposed methods have outperformed in the **A**spect **C**ategory **D**etection problem in Sentence-level **A**spect **B**ased **S**entiment **A**nalysis from SemEval-2016 task 5. Our results demonstrate that un-normalized sentence vector representation methods perform better than the normalized methods. It also shows that L1-norm for obtaining a sentence vector is always better than the L2-norm. In our research, we have also used difference of the word vectors in parallel with the sum of the word vector methods. Difference of word vectors have shown some promising results. Consequently, our investigation has shown that proposed vector representation methods for sentence representation is suitable for the aspect category classification task. The complete result summary with the increasing order of F1 scores is shown in Figure VI. The bold methods in the following table are our proposed sentence representation techniques. Results show that, our proposed sentence vector methods can also be applied to solve other type of sentence classification problems.

TABLE VI. SUMMARY OF ALL THE RESULTS IN ASPECT CATEGORY DETECTION PROBLEM

Threshold for conducting all the experiments (τ) = 0.785				
Method	**Category**	**Precision**	**Recall**	**F1 scores**
SOW	Un-Normalized	77.04	75.90	76.40
SOW⊕DOW	**Un-Normalized**	75.09	76.31	**75.70**
DOW	**Un-Normalized**	76.26	74.83	**75.54**
L1-DOW	**Normalized**	77.46	73.08	**75.20**
L1-SOW	**Normalized**	77.41	72.40	**74.82**
L1-AvgDOW	**Normalized**	75.20	73.88	**74.54**
L1-AvgSOW	**Normalized**	74.62	73.21	**73.91**
L2-SOW	**Normalized**	69.65	78.46	**73.79**
NLANGP (current best scores)	—	72.45	73.62	73.03
L2-DOW	**Normalized**	67.88	77.65	**72.44**
L2-AvgDOW	**Normalized**	65.74	80.08	**72.20**
Avg-SUM	Normalized	66.66	77.79	71.80
L2-AvgSOW	Normalized	65.82	77.52	71.19
Avg-DOW	**Normalized**	62.88	79.81	**70.34**

VII. LIMITATIONS OF PROPOSED METHODS

In this section we discuss about the limitations of our system under the best performing methodology based on sum of the word vectors *(SOW)* approach. Our findings show that the Aspect Category Detection (ACD) system is confused about some categories based on the context provided in the training or input sentences. Given sentences contains inadequate information and ambiguous sentence annotations. Training a model on such sentences will lead to the ambiguities in labeling a sentence. Our system results are effected by the following ambiguities:

A. Ambiguity due to the presence of personal pronouns/inadequate information in a sentence

Personal pronouns *(e.g. I, you, he, she, it, we, they, me, him, her, us, and them)* are very often used to refer something from the context by substituting in the place of noun, people or person. It is essential to have a context to understand such type of sentences that contains personal pronouns. Context helps to understand the complete meaning of a given sentence (or phrase) by contemplating the referenced noun, people or person in the previous sentences. In Fig. 6 few sentences are

shown, that are not correctly predicted by our system due to the presence of personal pronouns. The first sentence that is *"Don't leave the restaurant without it"* is assigned to the category of "RESTAURANT#GENERAL" by looking at "restaurant" word but, the correct category for this sentence is "FOOD#QUALITY". This happened because our system totally ignored the personal pronoun "it" during the aspect category classification.

Sentence	Predicted Category	Actual Category
Don't leave the restaurant without it.	['RESTAURANT#GENERAL']	['FOOD#QUALITY']
It was absolutely amazing.	['FOOD#QUALITY']	['RESTAURANT#GENERAL']
It's *very* reasonably priced, esp for the quality of the food.	['FOOD#PRICES' 'FOOD#QUALITY' 'RESTAURANT#PRICES']	['FOOD#PRICES', 'FOOD#QUALITY']
AMAZING.	['FOOD#QUALITY' 'RESTAURANT#GENERAL']	['RESTAURANT#GENERAL']

Fig. 6. Ambiguity due to presence of personal pronouns

There exist another problem that occurs due to the inadequate information in sentences, in which sentences are only composed of one or very few words. Such type of sentences are unable to provide complete information about what is being said in a sentence. In order to mitigate such type of issues again context is very important. As shown in the above figure that the word *"AMAZING"* is an input sentence and it can be referred to any category between "FOOD#QUALITY" or "RESTAURANT#GENERAL". Therefore, system returned both categories for that because it seems like sentence can lie in both categories. But if we provide enough context then it is possible that, system will ignore the category of "FOOD#QUALITY", and prefer "RESTAURANT#GENERAL" only.

B. Ambiguous annotations in provided sentences

There are many sentences which are strictly assigned to specific categories in the provided restaurant reviews dataset (SemEval 2016 ABSA task 5). For example the fourth sentence *"I liked the atmosphere very much but the food was not worth the price."* in Fig. 7 is annotated with the categories ['AMBIENCE#GENERAL', 'FOOD#PRICES', 'FOOD#QUALITY'], where our predicted categories are ['AMBIENCE#GENERAL', 'FOOD#QUALITY', 'RESTAURANT#PRICES'].

Although our system successfully predicted the two out of three categories. Our system is confused in categories of 'RESTAURANT#PRICES' and 'FOOD#PRICES'. Next sentence *"It is not worth going at all and spend your money there!!!"* is also annotated with the category of "'RESTAURANT#GENERAL'", where our predicted categories are ['RESTAURANT#GENERAL', 'RESTAURANT#PRICES']. Again our system returned categories are partially correct, because it has predicted 'RESTAURANT#PRICES' due to the presence of the word "money" in a sentence. Consequently, such type of strict

Sentence	Predicted Category	Actual Category
It is not worth going at all and spend your money there!!!	['RESTAURANT#GENERAL' 'RESTAURANT#PRICES']	['RESTAURANT#GENERAL']
Mama Mia – I live in the neighborhood and feel lucky to live by such a great pizza place.	['AMBIENCE#GENERAL' 'RESTAURANT#GENERAL']	['RESTAURANT#GENERAL']
Its worth the wait, especially since they'll give you a call when the table is ready.	['SERVICE#GENERAL']	['RESTAURANT#GENERAL', 'SERVICE#GENERAL']
I liked the atmosphere very much but the food was not worth the price.	['AMBIENCE#GENERAL' 'FOOD#QUALITY' 'RESTAURANT#PRICES']	['AMBIENCE#GENERAL', 'FOOD#PRICES', 'FOOD#QUALITY']

Fig. 7. Annotation ambiguities

annotations are hard to match even humans are sometimes confused during such type of tagging.

VIII. Conclusion

In this paper, we have proposed a simple and computationally less expensive method to represent a language sentence in vector spaces. Proposed sentence representation methods are divided into two categories, normalized and un-normalized sentence vector representations. The performance of these methods are evaluated by detecting the aspect categories from online restaurant review sentences. **A**spect **C**ategory **D**etection is a part of **A**spect **B**ased **S**entiment **A**nalysis, which is presented as a global challenge in SemEval-2016 task 5. We have compared our experimentation results with the existing systems and shown that our **A**spect **C**ategory **D**etection model has outperformed in this task and achieved the best F1-scores 76.40. Our experimental study also show that, un-normalized sentence vector representation methods always perform better than the normalized sentence vectors. Moreover, the sum of the word vectors method is used and experimented in the past, but we have also used difference of the word vectors method for sentence representation. Our methods can be extended and applied to any sentence classification problem.

A. Future Work

In this section, we discuss some ideas to improve the performance of our **A**spect **C**ategory **D**etection system. We also talk about different directions to extend our work.

1) Improving aspect category detection of sentences by incorporating the context: In the given dataset the restaurant reviews are in the form of multiple sentences (or a paragraph). Where each sentence is labeled with aspect categories. Understanding of an individual sentence depends on the previous sentences in which the given sentence is presented. Therefore, sentence should be labeled by looking at the contextual information. Incorporating the contextual information in a sentence will help to reduce the ambiguities due to the presence of personal pronoun and inadequate information in sentences. We can solve this problem by replacing personal pronouns *(e.g. it, they*

...) with a suitable reference *(or noun)* words by looking into the contextual sentence(s). Reference between personal pronouns and the context words can be mapped by using dependency parser. After successful mapping between personal pronouns and nouns from the context, it is possible to substitute personal pronouns with proper meaningful words.

For example, sentence1: "*Don't leave the restaurant without it*" is incorrectly labeled by our system which contains a personal pronoun "*it*", and sentence2: "*Green Tea creme brulee is a must!*" is the previous sentence of sentence1. In sentence2 the term:"*Green Tea creme brulee*" is referring a personal pronoun in sentence1. So, if we substitute this term:"*Green Tea creme brulee*" with "*it*" then sentence1 will look like, "*Don't leave the restaurant without Green Tea creme brulee*" as shown in Fig. 8. Such type of pre-processing must be done before presenting a sentence to the system to avoid personal pronoun ambiguities. We have experimented with such type of substitutions for personal pronouns and it worked fine. These results will be published in the next paper.

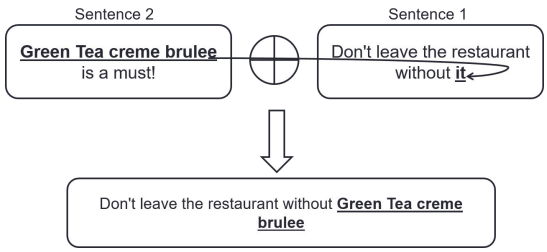

Fig. 8. Personal pronoun substitution from context

2) Integration of word positional importance in sentence vectors: In this paper, we presented different sentence vector representation methods based on word vectors. None of the proposed approaches maintained the order of word positions as they were in the original sentence. Our future goal is to derive a strategy to represent sentence vectors by incorporating the notion of word positions in them.

3) Classification using other learning models: We can also use other neural network architectures for training. We can model our sentence classification problem into two ways that are, 1). Sequence to Sequence modeling and 2). Sequence to label modeling. In order to apply *sequence to sequence models* on our problem, we need to consider given labels of each sentence as words and concatenate them with the sentence words. We can treat a sentence and concatenated label a complete training example. Such type of setting will allow us to apply recursive neural networks (R-NN) and long short Term memory (LSTM) network models for **A**spect **C**ategory **D**etection task. In such type of modeling, the goal is to predict the next word in the sequence on the basis of a given word. One advantage of using such methods is that, we are not required to have fix size sentence vectors or any of kind of sentence vector representations. We are only required to have word vectors for representing each word in the sentence sequence. Where in *sequence to label modeling* we can use convolution neural networks (C-NN), which has shown great success in the field of sentence classification.

REFERENCES

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

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[3] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[4] B. Lu, M. Ott, C. Cardie, and B. K. Tsou, "Multi-aspect sentiment analysis with topic models," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 81–88.

[5] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[6] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Computational linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[7] T. Alvarez-López, E. Costa-Montenegro, and F. J. González-Castano, "Gti at semeval-2016 task 5: Svm and crf for aspect detection and unsupervised aspect-based sentiment analysis." 2016.

[8] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content."

[9] X. Zhou, X. Wan, and J. Xiao, "Representation learning for aspect category detection in online reviews." 2015.

[10] Z. Toh and J. Su, "Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features," *Proceedings of SemEval*, pp. 282–288, 2016.

[11] T. T. Thet, J.-C. Na, and C. S. Khoo, "Aspect-based sentiment analysis of movie reviews on discussion boards," *Journal of information science*, p. 0165551510388123, 2010.

[12] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 111–120.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[14] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[15] S. Brody and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 804–812.

[16] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a maxent-lda hybrid," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 56–65.

[17] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 377–384.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[19] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–43.

[20] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 873–882.

[21] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit, "Semeval-2016 task 5: Aspect based sentiment analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 19–30. [Online]. Available: TOBEFILLED-http://www.aclweb.org/anthology/W/W05/W05-0202

[22] Z. Toh and J. Su, "Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction," 2015.

[23] K. Y, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)pp. 17461751*, 2014.

[24] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *arXiv:1405.4053v2*, 2014.

[25] B. Wang and B. Liu, "Deep learning for aspect-based sentiment analysis," in *Stanford University*, 2015.

[26] A. Alghunaim, "A vector space approach for aspect-based sentiment analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 486–495.

[31] P. Rahim, Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.