Understanding Citizen Issues through Reviews: A Step towards Data Informed Planning in Smart Cities

Noman Dilawar, Hammad Majeed, Mirza Omer Beg, Naveed Ejaz, Khan Muhammad, Irfan Mehmood, and Yunyoung Nam

1 Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad 46000, Pakistan; noumandilawar@gmail.com (N.D.); hammad.majeed@nu.edu.pk (H.M.); omer.beg@nu.edu.pk (M.O.B.)
2 Department of Computing and Technology, Iqra University, Islamabad 46000, Pakistan; naveed.ejaz@iqraisb.edu.pk
3 Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University, Seoul 143747, Korea; khan.muhammad@ieee.org
4 Department of Software, Sejong University, Seoul 143747, Korea
5 Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, Korea

*Correspondence: irfanmehmood@ieee.org (I.M.); ynam@sch.ac.kr (Y.N.);
Tel.: +82-010-9222-8794 (I.M.); +82-41-530-1282 (Y.N.)

Received: 27 July 2018; Accepted: 28 August 2018; Published: 7 September 2018

Featured Application: This work can be converted into an application that takes user reviews as input and classifies them into Aspect Categories at the output. These classified categories can facilitate decision makers planners to identify the most discussed categories. This information can be utilized to make informed decisions during the planning process..

Abstract: Governments these days are demanding better Smart City technologies in order to connect with citizens and understand their demands. For such governments, much needed information exists on social media where members belonging to diverse groups share different interests, post statuses, review and comment on various topics. Aspect extraction from this data can provide a thorough understanding of citizens’ behaviors and choices. Also, categorization of these aspects can better summarize societal concerns regarding political, economic, religious and social issues. Aspect category detection (ACD) from people reviews is one of the major tasks of aspect-based sentiment analysis (ABSA). The success of ABSA is mainly defined by the inexpensive and accurate machine-processable representation of the raw input sentences. Previous approaches rely on cumbersome feature extraction procedures from sentences, which adds its own complexity and inaccuracy in performing ACD tasks. In this paper, we propose an inexpensive and simple method to obtain the most suitable representation of a sentence-vector through different algebraic combinations of a sentence’s word vectors, which will act as an input to any machine learning classifier. We have tested our technique on the restaurant review data provided in SemEval-2015 and SemEval-2016. SemEval is a series of global challenges to evaluate the effectiveness of disambiguation of word sense. Our results showed the highest F1-scores of 76.40% in SemEval-2016 Task 5, and 94.99% in SemEval-2015 Task 12.

Keywords: smart cities; supervised learning; aspect category detection; aspect-based sentiment analysis
1. Introduction

The world population is growing at an extraordinary rate and cities are becoming large and dense [1]. Governments are therefore facing challenges in managing such cities and alleviating citizens’ problems in a timely and efficient way. Information and communication technologies (ICT) can be used to overcome these challenges by making cities smart. The main objective of smart cities is to make them more attractive, conducive and connected, and also to enhance the living standards of their residents. Research in this area has gained a lot of attention in recent years [2,3], and as a result of this, citizens feel more connected with the rest of the world.

An unprecedented amount of textual data is being produced on social media; this is also known as “Big Data”. People compose statuses and messages on micro-blogging (e.g., Twitter) and social networking sites to express their opinions on numerous topics. Users of micro-blogging sites can be anyone, including politicians, celebrities, businessmen and students; thus, it is possible to collect information from different communal and age groups in a society. User generated textual data contains information that can improve the decision-making process of governments to improve the lifestyles of their citizens [4,5]. Sentiment analysis plays a vital role in understanding public demands and uses the information provided in textual data. Recent times have proven that governments may be critically affected if they ignore citizen’s demands (or sentiments) or fail to respond appropriately to them [6].

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 [7]. The simplest form of sentiment analysis classifies reviews, paragraphs or any text as positive or negative [8,9]. This type of analysis is incapable of handling conflicting sentiments within a text. For example, a simple sentiment analysis approach for the sentence “Despite having experienced leadership, Government is unable to resolve water crises.” would annotate it with the label, “conflicting”. This is due to the concurrent negative and positive sentiments about the government although careful examination would reveal that this sentence is expressing a positive sentiment for the government from the perspective of its leadership and a negative sentiment about the issue of the water crises. Such examples prove that simple sentiment analysis does not provide in-depth information about sentiments and a detailed sentiment analysis is required to capture the multiple dimensions of the opinionated text content [10].

In 2004, a framework was proposed to extract feature-based (or aspect-based) summaries from customer reviews [11]. It works by decomposing conventional sentiment analysis into three subtasks: (1) product features extraction (i.e. identification of the features about which customers have expressed their opinions); (2) assigning sentiments to product features; and (3) generating summaries on the basis of extracted information. This type of feature-based method for sentiment analysis is known as ABSA (aspect-based sentiment analysis).

Aspect category detection (ACD) is one of the important tasks 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-2015 [12] and SemEval-2016 [13], categories are assigned from a predefined set of Entity (e.g., restaurant, food, laptop) and Attribute (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. An opinion without knowing its target is of limited use [14]. Obtained aspect categories are associated with their sentiment polarities to generate opinionated aspect-based summaries as shown in Table 1.

Table 1. ABSA (aspect-based sentiment analysis) identifies 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, great 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 the past, several approaches have been proposed to address this task. The most common among these is support vector machine (SVM) classification [15,16]. These approaches rely on syntactic features of sentences and ignore the semantic relationships among the words. Modern methods use word embedding [17] to represent the words as vector features. Sentence vectors can be obtained by coalescing word vectors. In our research, we transform a sentence into a vector by combining its word vectors. The acquired sentence vector is then used as an input to a machine learning classifier. Various sentence vector representations are proposed and evaluated on benchmark datasets.

This work is focused on detecting aspect categories from English restaurant reviews, which is presented as a case study to demonstrate the usefulness of resolving citizen issues in smart cities through textual data readily available on social networking sites. It does so by refining the sentence vector representation process on top of the word2vec [18] model’s word embedding. The proposed approach is simple and does not require high-end computation. The novel element of this research is finding the most appropriate sentence vector representation from the combination of different algebraic operations (e.g., sum, multiplication, division) on sentence vectors. Each algebraic operation on a combination of sentences’ word vectors results in an independent vector, called a sentence vector. The goal of this study is to obtain a sentence vector that can act as an input for any machine learning algorithm. By using the proposed approach, we were able to improve upon the best results reported for challenging real-world problems. The rest of this article is organized as follows: a review of related work can be found in Section 2; our proposed approach is discussed in Section 3; the tasks and datasets used in our experiments are found in Section 4; the experimental setup is provided in Section 5; results and discussions are reported in Section 6; Section 7 discusses the application of the proposed technique for improved planning; the limitations of our work is presented in Section 8; and finally, Section 9 concludes the paper and outlines future research directions.

2. Related Work

2.1. Aspect Category Detection in Sentiment Analysis

Individual opinions are directly associated with the aspects (e.g., food, taste) incorporated by an entity (e.g., a restaurant). Aspect-based sentiment analysis is a powerful opinion mining technique to understand the sentiments associated with the entity’s aspects, in which the goal is to find aspects and their associated sentiments from a given review [19].

The identification process of aspect phrases/words from on-line user reviews has been well studied since 2014. In recent years, topic modeling approaches have been used extensively for this task. Such methods detect ratable aspects from on-line user reviews and cluster them into their corresponding categories. In [20], multi-grained topic models were presented by extending the work of Hofmann [21] and Brody [22], which extracted aspect categories with high accuracy. Unlike previous topic modeling approaches, the methods proposed in [23] were focused on simultaneously extracting aspect terms and their associated opinions.

ACD is an important part 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 [24] were the most popular tools for doing this task.

2.2. Word Embedding for Sentences and Their Importance in Aspect Category Detection

Word embedding techniques presented in Section 1 for single-word vector space models capture a great sense of the language features, such as plurality, grammatical structures and even central concepts like “capital city of” [25]. Understanding the semantics of longer phrases is still a challenging problem. Recent attempts have applied different machine learning models [26,27] to capture the meaning of single-word vector compositions for deeper language understanding.
In [28], a semi-supervised method for word embedding is proposed and vectors for longer phrases are obtained by a word averaging method. Eventually, a logistic regression classifier is trained on sentence vector representations for predicting aspect categories with the highest accuracy in SemEval-2014 Task 4 (http://alt.qcri.org/semeval2014/task4/), slot 3. A sentence-matrix is defined in [29,30] for representing sentences in vectors, where rows are represented by words of a sentence. A pre-trained word2vec model is used for obtaining word vectors to represent each row of a matrix. Evaluation of this sentence representation method was performed by training a convolution neural network (CNN) for different tasks and achieved state-of-the-art results on 4 out of 7 tasks. A similar sentence matrix was used to extract more enhanced features by using deep CNN [31]. On top of these features, the one-vs.-all strategy was used to train single layer feed-forward binary classifiers against each given aspect category. A normalized average vector (NAV) method was proposed in [32] to train SVM for aspect category detection. As the name suggests, this method adds normalized word vectors.

3. Proposed Methodology

The proposed methodology starts with the removal of stop words and noise symbols. The remaining words are transformed into vector representations using a word2vec model. Word vectors are combined to form a fixed length vector to represent a given sentence. The sentence vector is passed as a feature vector to a neural network model. The graphical representation of the system architecture is shown in Figure 1.

3.1. 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 normalized and un-normalized forms of the resultant vector are used for further processing. These operations are discussed in detail in the following sections.

3.1.1. Normalized Representation of Sentence Vector

In this category, normalization techniques for representing a sentence as a feature vector are discussed. In the following expressions, \( w_i \) represents the word vector of \( i \)th word and \( n \) is the count of words in the given sentence \( S \), where \( L1 \) and \( L2 \) are the mathematical norm of vectors.

1. **Averaging Difference of Word Vectors.**

   \[
   \text{Avg}^{\text{Sub}}_{\text{Sub}} = \frac{\overrightarrow{w_1} - \sum_{i=2}^{n} \overrightarrow{w_i}}{n}
   \]  

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

Figure 1. System architecture of aspect category detection (ACD).
The $L_1$-norm of average word vector is used to represent a sentence as a feature vector as per Equation (2):

$$L_1 - \text{AvgSOW} = \frac{1}{n} \frac{\sum_{i=1}^{n} \vec{w}_i}{\sum_{i=1}^{n} |\vec{w}_i|}$$  \hspace{1cm} (2)

(3) **$L_1$-Normalized Difference of Average Word Vectors.**

This method is the opposite of the method mentioned as (2). The sum operator is replaced with a difference operator:

$$L_1 - \text{AvgDOW} = \frac{1}{n} \left( \vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i \right) \frac{1}{\sum_{i=1}^{n} |\vec{w}_i|}$$  \hspace{1cm} (3)

(4) **$L_2$-Normalized Average Difference of Word Vectors.**

This method is similar to the $L_2$-NAV (sum) proposed in [30]; here, we have used subtraction of the word vectors instead of addition in a sentence to represent a feature vector of a sentence:

$$L_2 - \text{AvgDOW} = \frac{1}{n} \left( \vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i \right) \frac{1}{\sum_{i=1}^{n} |\vec{w}_i|}$$  \hspace{1cm} (4)

(5) **$L_1$-Normalized Sum of Word Vectors.**

In this method all the word vectors in a sentence are summed and normalized by $L_1$-norm. $L_1$-norm is used to scale vector features in such a way that sum must be equal to 1:

$$L_1 - \text{SOW} = \frac{\sum_{i=1}^{n} \vec{w}_i}{||\sum_{i=1}^{n} \vec{w}_i||}$$  \hspace{1cm} (5)

(6) **$L_2$-Normalized Sum of Word Vectors.**

In this method all the word vectors in a sentence are summed and normalized by the $L_2$-norm:

$$L_2 - \text{SOW} = \frac{\sum_{i=1}^{n} \vec{w}_i}{||\sum_{i=1}^{n} \vec{w}_i||}$$  \hspace{1cm} (6)

(7) **$L_1$-Normalized Difference of Word Vectors.**

This method is the opposite to (5). Subtraction of word vectors is used instead of addition:

$$L_1 - \text{DOW} = \frac{\vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i}{||\vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i||}$$  \hspace{1cm} (7)

(8) **$L_2$-Normalized Difference of Word Vectors.**

This is the opposite method of (6). Here we use subtraction of word vectors instead of addition:

$$L_2 - \text{DOW} = \frac{\vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i}{||\vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i||}$$  \hspace{1cm} (8)
3.1.2. Un-Normalized Representation of Sentence Vectors

In this method, vector representations of sentences are obtained by omitting the normalization step, and this is known as un-normalized representation of a sentence vector. It is a much simpler way of representing sentence features using word vectors.

(1) Difference of Word Vectors.

In this method, sentence vectors are obtained by subtracting all the word vectors of a sentence. Then the resultant vector is used as a feature vector to illustrate the semantics of a sentence:

\[ DOW = \vec{w}_1 - \sum_{i=2}^{n} \vec{w}_i \]  

(9)

(2) Concatenation of Sum and Difference of Word Vectors.

In this method both feature vectors \( SOW \) (sum of word vectors in a sentence) and \( DOW \) are concatenated. The joined form is used as a feature vector for representing any given sentence:

\[ SOW \oplus DOW \]  

(10)

The motivation behind this research was to find a natural and fast method to represent a sentence vector by combining word vectors. The performance of the proposed sentence representations were tested on real-world problems.

3.2. Details of the Neural Network

In this study, the ACD problem is transformed into a multi-label multi-class sentence classification problem to test the performance of the representations proposed in Section 3.1. The goal is to assign single or multiple categories to the given input sentence vector \( \vec{x} \). The output of the classification algorithm is a vector \( \vec{y} \) with size equal to the number of the predefined categories. \( \vec{y} \) as one or more classes enabled for the given sentence. A rectified linear unit (ReLU) feed forward multi-layer neural network \([33,34]\) is used as a classifier.

The input layer of the neural network takes a sentence feature vector as an input. The number of neurons that represent this layer must be equal to the length of the 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. The hidden layer passes input features through non-linearities to perform linear transformations on the input feature vectors. ReLU is used as an activation function. ReLU is preferred because (1) it has low computational cost as compared to sigmoid or tanh functions as it does not require expensive operations like calculating exponential; and (2) ReLU has a fast convergence rate on stochastic gradient descent as compared to sigmoid and tanh functions.

The Softmax regression is used as a cost function and is shown in Equation (11).

\[ L_i = -\log \left( \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \right) \]  

(11)

After simplifying the above equation, we get:

\[ L_i = -f_{y_i} + \log \left( \sum_j e^{f_j} \right) \]  

(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 is used. L2-regularization is one way to control over-fitting in neural networks. It penalizes the squared magnitude of all the neural network parameters except bias inputs, and then adds it in to the objective function, as shown in Equation (13):

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

$\lambda$ is a regularization controlling factor to penalize weight. The final objective function became $L_4 + R(W)$ by using regularization.

The 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 [35] is used to optimize the network weights. Mathematically, the optimizer can be written as:

$$m_i = \beta_1(m_{i-1}) + (1 - \beta_1)(\Delta L(W))_i$$  \hspace{1cm} (14)

$$v_i = \beta_2(v_{i-1}) + (1 - \beta_2)(\Delta L(W))^2_i$$  \hspace{1cm} (15)

$$W_{i+1} = W_i - \alpha \frac{\sqrt{v_i} - \beta_2 v_i}{1 - \beta_1^t} (m_i)$$  \hspace{1cm} (16)

A hyper parameter called the learning rate ($\alpha$) is used to control the step size during each update of weights. An exponential decay method is 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 the result vector $x_i \in R^D$, weight matrix $W \in [K \times D]$ and bias $b \in [1 \times K]$ as an input and return scores for all classes $y_i \in [1 \times K]$ as shown in Equations (17) and (18).

$$\sigma(x) = 1/(1 + e^{(-x)})$$ \hspace{1cm} (17)

$$f(x, W, b) = \sigma(Wx + b)$$ \hspace{1cm} (18)

where example $x_i$ varies, $i = \{1, \ldots, n\}$, $y_i \in \{1, \ldots, K\}$, $D$ is the dimensions of the 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]. An example of a score function is shown in Figure 2. $X_i$ is the sentence representation and is [56, 23, 1, 24, 2], $W$ is a $3 \times 2$ weight matrix. $b$ represents the bias vector. By performing the matrix multiplication and addition steps on the left-hand side, we get the scores for all the $K$ classes. For this example, $K = 3$. The input sentence belongs to the class with the highest score.

![Figure 2. Example of score function.](image-url)

4. Tasks and Datasets

We used English restaurant review datasets (Available Online: 1. http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools; 2. http://alt.qcri.org/semeval2015/task12/index.php?id=data-
and-tools) provided in the SemEval-2016 and SemEval-2015. Each sentence in the restaurant reviews dataset was annotated with aspect terms (e.g., “pizza”, “fish”, “food”, “restaurant”) and those aspect terms were assigned/labeled to their aspect categories (e.g., “FOOD”, “QUALITY”) with polarities (e.g., “Negative”, “Positive”). SemEval-2016 English restaurant reviews dataset contains 2000 training and 676 test sentences. In these sentences there were 1708 training and 587 test sentences labeled with aspect categories and remaining sentences were labeled with ‘outOfScope’ or ‘None’ categories. Sentences with ‘outOfScope’ or ‘None’ categories were not used in the final evaluation of the ACD results in SemEval-2016 Task 5. Similarly, SemEval-2015 English restaurant reviews contain 1120 training and 582 test sentences with categories. Each restaurant review consists of multiple sentences annotated with their respective aspect categories. Aspect categories are the combination of Attribute and Entity (E#A) pairs as discussed before. A sample of a single customer review from the restaurant dataset is shown in Figure 3.

Aspect categories are unevenly distributed across the training and test review sentences; therefore, the dataset is highly unbalanced. This can affect the training and prediction accuracies. The distribution of aspect categories in SemEval-2016 Task 5 is shown in Table 2. We have disregarded the repetition for the same categories of a single sentence and consider their counts as one as discussed in SemEval-2015 and SemEval-2016.

![Figure 3](image-url)  
**Figure 3.** Annotation of sentences in a single training review instance in English restaurant reviews dataset.

**Table 2.** Aspect categories distribution across training and test sentences.

| 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    |
| DRINK#QUALITY               | 46       | 21    |
| DRINKS#STYLE_OPTIONS        | 30       | 12    |
| LOCATION#GENERAL            | 28       | 13    |
| DRINKS#PRICES               | 20       | 3     |
| TOTAL: 12                   | TOTAL: 2258 | TOTAL: 743 |
In SemEval-2015 and SemEval-2016, ACD systems are divided into two categories. One is constrained (C) systems and the second is unconstrained (U) systems. In constrained systems, no external training dataset is allowed during training while in unconstrained systems, it is allowed to use datasets from outside sources (e.g., Yelp, Amazon). We were working in the category of unconstrained systems.

We used distributed representations of words to build dense sentence vectors. For this, we used a skip-gram approach (as discussed in Section 2) to train the word2vec model. English restaurant review datasets contain only 3315 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 the word2vec model. Consequently, we ended up using Yelp restaurant reviews dataset (This dataset can be found at: http://www.yelp.com/dataset_challenge) to train our model. The effect of domain specific word vectors is explained in [30]. Yelp restaurant reviews contained 131,778 unique words and about 200 million tokens with 2225,213 sentences. We used 2000 sentences from the SemEval-2016 training set as well as the first 500,000 sentences from Yelp restaurant reviews to train the word2vec model. It is important to note here that the challenge allowed the participants to use datasets other than the provided training dataset.

5. Experimental Setup

First, the restaurant review sentences were passed through a pre-processing stage. At this stage, a stream of tokens was generated from sentences and stop-words were removed. English restaurant review sentences from Yelp and SemEval-2016 Task 5 were combined to train the word2vec model. The word2vec model was trained using Gensim [36]. Each sentence vector \( \vec{x} \) in the English restaurant review dataset belongs to multiple categories, which can be interpreted as an output vector \( \vec{y} \). Furthermore, a multiple one hot encoded scheme was used to represent multiple categories or classes in \( \vec{y} \). The dimensions of vector \( \vec{y} \) were fixed and equal to the predefined set of twelve classes (12 × 1).

The problem with ACD 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 was implemented using Tensorflow [37]. Softmax regression as a cost function was used to train the neural network model, along with the sigmoid function on the output layer, to return the output scores. We tuned a single threshold (\( \tau = 0.785 \)) for conducting all of our experiments. We always considered a predicted class, if its score becomes greater than the decided threshold and note that in this case, we also have multiple categories. Tuning of training hyper-parameters of word2vec and neural network models are discussed later.

5.1. Word Vectors

A word2vec (skip-gram) model was trained on 502,000 restaurant review sentences. These sentences were obtained from Yelp and SemEval-2016 Task 5 datasets. Effective training of our model relied 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 were used to control the training of the word2vec (skip-gram) model and their best values were chosen by hit and trial. Moreover, a summary of the values of parameters is shown in Table 3.

| Table 3. Word2vec model tuned parameters. |
|------------------------------------------|
| Parameter                  | Value         |
|---------------------------|---------------|
| Word size                 | 400           |
| Min word count            | 1             |
| Number of workers         | 5             |
| Context                   | 5             |
| Down sampling             | \( 1 \times 10^{-3} \) |
Dimension (D): This is used to control the size of the word embedding or vector. The size chosen for each word vector during training was $D = 400$. Thus, each $\vec{w}$ is equal to the dimensions of $[1 \times D]$.

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 used a minimum word count equal to 1 because we already had a small training dataset and wanted to have word vectors for the maximum number of words.

Context: Context is actually the size of a window around each word. We used a context size of five words. It means that for any centered word $w_0$ in a context size of five, we always had its left $w_{-1}$, $w_{-2}$ and right $w_1$, $w_2$ context. This is one of the most important parameters of the word2vec model.

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 $[1 \times 10^{-5}, 1 \times 10^{-3}]$.

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.

We may not have all the word vectors against each word that falls under the domain of restaurant reviews. To address the missing word-vectors problem, we replaced them with zeros equal to the dimensions of the existing word vectors $\vec{w}$.

5.2. Classification Model

We used a two-layer neural network model for aspect category detection. 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 was evaluated on the test dataset. The dataset was divided into training and validation sets with a ratio of 85 and 15. For training our neural network, we used 1708 sentences from SemEval-2016 Task 5. The neural network training parameters are given in Table 4.

| Parameter         | Value                        |
|-------------------|------------------------------|
| Epochs            | 100                          |
| Hidden layer units| layer 1: 300, layer 2: 250   |
| Batch size        | 80                           |
| Base learning rate| 0.002                        |
| Decay rate        | 0.96                         |
| Regularization    | 0.003                        |

We applied adaptive learning rates that gradually slow down the step size to achieve fast and optimal convergence. The adaptive learning rate of the neural network is controlled by the decay rate, which gradually reduces the learning rate by a small 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 was fixed, and contained 80 examples in each set of training examples. Before starting each epoch, the training dataset was shuffled. Loss and accuracy was observed after every 5 epochs, as shown in Figure 4.
6. Results and Discussion

Our ACD model was 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 [34] and SemEval-2015 Task 12 [35]. We compared our experimental results with the current best scores. F1 scores were calculated under the known definitions of precision (\(p\)), and recall (\(r\)), which are:

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

6.1. Normalized Representation of Sentence Vectors

This experiment was performed in three phases. Initially, \(\text{Avg-SOW}\) and \(\text{Avg-DOW}\) methods were used, which are based on word averaging. In the second phase, we used \(L1-\text{AvgSOW}, \ \text{L1-AvgDOW}, \ \text{L2-AvgDOW}\) and \(L2-\text{AvgSOW}\) methods based on normalized average word vectors. In the third phase, normalized sum and difference of word vectors were used to represent sentences that are, \(L1-\text{SOW}, \ \text{L1-DOW}, \ \text{L2-SOW}\) and \(L2-DOW\). Normalized representation of sentence vectors achieved highest F1 scores in the ACD problem as compared to the best systems. Experimentation results are shown in Tables 5 and 6 (proposed methods are in bold letters).
Table 5. Results of normalized representation of sentence vectors on SemEval-2016 dataset.

| 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     |

Table 6. Results of normalized representation of sentence vectors on SemEval-2015 dataset.

| Method       | Precision | Recall | F1 Scores |
|--------------|-----------|--------|-----------|
| Avg-SUM      | 79.16     | 90.19  | 84.31     |
| Avg-DOW      | 74.65     | 89.67  | 81.47     |
| L1-AvgSOW    | -         | -      | -         |
| L2-AvgSOW    | 79.81     | 90.32  | 84.74     |
| L1-AvgDOW    | 93.85     | 94.58  | 94.21     |
| L2-AvgDOW    | 75.55     | 92.12  | 83.02     |
| L1-SOW       | 95.80     | 94.19  | 94.99     |
| L2-SOW       | 80.33     | 93.80  | 86.54     |
| L1-DOW       | 95.25     | 93.29  | 94.26     |
| L2-DOW       | 75.26     | 90.70  | 82.27     |

6.2. Normalized Representation of Sentence Vectors

In un-normalized representation of sentence vectors, we combined all word vectors of a sentence using two arithmetic operators (i.e., addition and subtraction). This is a simpler way of representing a sentence. Two types of analysis were performed under un-normalized methods. In the first experiment, two methods were used, SOW and DOW that are based on the sum and difference of word vectors, whereas in the second experiment, concatenation of SOW⊕DOW was used. All methods in this category outperformed the previous results in the ACD task. Results are shown in Tables 7 and 8 (proposed methods are in bold letters).

Table 7. Results of un-normalized representation of sentence vectors on SemEval-2016 dataset.

| 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     |

Table 8. Results of un-normalized representation of sentence vectors on SemEval-2015 dataset.

| Method       | Precision | Recall | F1 Scores |
|--------------|-----------|--------|-----------|
| SOW          | 90.27     | 89.80  | 90.03     |
| DOW          | 93.14     | 92.90  | 93.02     |
| SOW⊕DOW      | 89.28     | 91.35  | 90.30     |

Our experimental studies showed some interesting results and many of our proposed methods outperformed the state-of-the-art approaches in the ACD task. Moreover, our results show that the L1-norm for obtaining a sentence vector performed better than the L2-norm. In our research, we also used the difference of the word vectors in parallel with the sum of the word vector methods.
The difference of word vectors has shown some promising results. Consequently, our investigation shows that the proposed vector representation methods for sentence representation are suitable for the aspect category classification task. Similarly, we can apply this technique to extracting aspect categories from people reviews/comments on social media as it has the same textual data. This could lead governments of smart cities to get a better understanding of citizens’ concerns by mapping aspect categories to existing issues. The complete summary of the result with the increasing order of F1 scores is shown in Tables 9 and 10 (proposed methods are in bold letters).

### Table 9. Results summary of aspect category detection in SemEval-2016 dataset.

| 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-ArgDOW | Normalized  | 75.20 | 73.88 | 74.54 |
| L1-ArgSOW | Normalized  | 74.62 | 73.21 | 73.91 |
| L2-SOW    | Normalized  | 69.65 | 78.46 | 73.79 |
| L2-DOW    | Normalized  | 67.88 | 77.65 | 72.44 |
| L2-ArgDOW | Normalized  | 65.74 | 80.08 | 72.20 |
| Avg-SUM   | Normalized  | 66.66 | 77.79 | 71.80 |
| L2-ArgSOW | Normalized  | 65.82 | 77.52 | 71.19 |
| Avg-DOW   | Normalized  | 62.88 | 79.81 | 70.34 |

### Table 10. Results summary of aspect category detection in SemEval-2015 dataset.

| Method    | Category | Precision | Recall | F1 Scores |
|-----------|----------|-----------|--------|-----------|
| L1-SOW    | Normalized | 95.80  | 94.19  | 94.99  |
| L1-DOW    | Normalized | 95.25  | 93.29  | 94.26  |
| L1-ArgDOW | Normalized | 93.85  | 94.58  | 94.21  |
| DOW       | Un-Normalized | 93.14 | 92.90  | 93.02  |
| SOW⊕DOW   | Un-Normalized | 89.28 | 91.35  | 90.30  |
| SOW       | Un-Normalized | 90.27 | 89.80  | 90.03  |
| L2-SOW    | Normalized  | 80.33  | 93.80  | 86.54  |
| L2-ArgSOW | Normalized  | 79.81  | 90.32  | 84.74  |
| Avg-SUM   | Normalized  | 79.16  | 90.19  | 83.31  |
| L2-ArgDOW | Normalized  | 75.55  | 92.12  | 83.02  |
| L2-DOW    | Normalized  | 75.26  | 90.70  | 82.27  |
| L2-ArgSOW | Normalized  | 74.65  | 89.67  | 81.47  |

7. Smart Planning Using ACD

In this section we discuss the process of augmenting the planning process with the proposed technique in order to make it smarter. Smart planning can be done at individual as well as community level. For example, for the presented case study of restaurant reviews, the user may only consider the reviews classified in the aspect category(ies) and plan on the basis of these reviews. This makes the decision-making process pragmatic and beneficial for the user. Community level planning can be improved by looking at the aspect categories that are marked important by a community. The simplest approach to identify important categories is to look at the most commented aspect categories. Other sophisticated techniques like clustering and customized filtering can also be employed for the identification of important categories. The outcome of this process is fed into the planning process to enable the meaningful participation of the community in the planning process. The work flow diagram of the aforementioned process is shown in Figure 5.
Appl. Sci. 2018, 8, 1589

Interestingly, the community seems to be oblivious to the high food prices if a restaurant is performing well on the aforementioned three categories. This can be beneficial to the planners/restaurant owners to maximize their profits.

For example, for the same case study, smart community level planning might include designing a restaurant that is most beneficial for the general public. For this, a simple count of group-based reviews for each aspect category identifies the most important features expected by the community in a good restaurant. A frequency bar graph for 12 aspect categories for new/unseen reviews is shown in Figure 6.

After examining this graph, it is easy to infer that all good restaurant plans will include special attention to “Food Quality”, “Service” and “Ambience”. Interestingly, the community seems to be oblivious to the high food prices if a restaurant is performing well on the aforementioned three categories. This can be beneficial to the planners/restaurant owners to maximize their profits.

**Figure 5.** A simple work flow to show the use of ACD for making the planning process smart and efficient.

**Figure 6.** Importance of different aspects of the reviews calculated after completion of ACD task. Frequency represents the number of times a reviewer has commented. “Frequency” is the super set of the aspect categories for the restaurant data set. Higher frequencies represent the most commented on or discussed category in the reviews.
8. Limitations of the Proposed Methods

This section explores the limitations of our ACD model. We shall use the results of our best performing model for the SemEval-2016 dataset, which is based on the sum of the word vectors (SOW) to better explain these limitations. Our study shows that the model is uncertain about the aspect categories in certain cases. Sentences from a dataset either contain inadequate information or ambiguous annotations. A learning model trained on such data can introduce ambiguities, and hence effect the performance of the classifier. Such misclassification/s can be avoided by providing enough contextual words to deliver the semantics of a sentence.

9. Conclusions and Future Work

In this paper, we provide an inexpensive yet accurate representation to identify the aspect categories from people opinions posted on social media. By understanding the hidden aspect categories associated with individual opinions, governments could highlight and deracinate some of the major problems of their citizens. A computationally inexpensive approach was devised for detecting aspect categories from peoples’ opinions. First, we represented a sentence vector by using algebraic combinations of its word vectors and then we used sentence vectors to train a neural network model. Different algebraic combinations of a sentence’s word vectors have been examined and their effectiveness is evaluated on benchmark datasets provided in SemEval-2016 Task 5 and SemEval-2015 Task 12. We compared our experimental results with existing systems and showed that our ACD model outperforms the state-of-the-art in this task, and it achieves the highest F1-scores of 76.40% in SemEval-2016 Task 5 and 94.99% in SemEval-2015 Task 12. The application of the research demonstrated the possible future use of the work in autonomously comprehending online textual data posted by the community. This has the potential to guide planning processes for the benefit of the masses. In the future, we need to address the following challenges to enhance the performance of our model:

**Challenge 1. 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 to something in the context of a paragraph or sentence. Personal pronouns are normally used as a replacement for a noun, people or person. It is essential to have a context to understand sentences that contain personal pronouns. Context helps to understand the complete meaning of a given sentence (or phrase) by considering the referenced noun, people or person in the previous sentences. In Table 11, a few sentences are shown which were not correctly predicted by our system due to the presence of personal pronouns. The first sentence, “Don’t leave the restaurant without it” was assigned to the category of “RESTAURANT#GENERAL” because of the word “restaurant” when the correct category for this sentence is “FOOD#QUALITY”. 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#PRICES”,                                          |
|                                                                        | “FOOD#QUALITY”,                                               | “FOOD#QUALITY”]                                         |
|                                                                        | “RESTAURANT#PRICES”                                           |                                                        |
| AMAZING                                                                  | [“FOOD#QUALITY”,                                               | [“RESTAURANT#GENERAL”]                                  |
|                                                                        | “RESTAURANT#GENERAL”                                          |                                                        |

Another issue exists that occurs due to inadequate information in sentences, that is, where the entire sentence is only composed of one or very few words. Such sentences are unable to provide
complete information about what is being said in the sentence. In order to mitigate such issues, understanding the context is very important. Consider the last example in Table 11, the word “AMAZING” is the input sentence and it could be referring to either the “FOOD#QUALITY” or “RESTAURANT#GENERAL” category. Therefore, the system returned both categories against the input because the sentence can lie in both categories. However, if we provide enough contextual information then it is possible for the system to ignore the category of “FOOD#QUALITY” and select only the “RESTAURANT#GENERAL” category.

**Challenge 2. Ambiguous annotations in provided sentences.**

There are many sentences that 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 Table 12 is annotated with the categories “AMBIENCE#GENERAL”, “FOOD#PRICES”, and “FOOD#QUALITY”, whereas our predicted categories are “AMBIENCE#GENERAL”, and “FOOD#QUALITY”, “RESTAURANT#PRICES”.

| 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”]                                                          |
| It’s worth the wait, especially since they’ll give you a call when the table is ready. | [“SERVICE#GENERAL”]                                                               | [“RESTAURANT#GENERAL”, “SERVICES#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”]                            |

Although, our system successfully predicted two out of the three classes, the ACD model was confused between the “RESTAURANT#PRICES” and “FOOD#PRICES” categories. The next sentence, “It is not worth going at all and spend your money there!!!” is also annotated with the category of “RESTAURANT#GENERAL”, whereas our predicted categories are “RESTAURANT#GENERAL”, and “RESTAURANT#PRICES”. Our system returned categories that are partially correct, because it predicted “RESTAURANT#PRICES” due to the presence of the word “money” in the sentence. Consequently, due to the existence of such ambiguities it is difficult to accurately categorize sentiment. Even humans can sometimes be confused if such ambiguous tagging exists.

Each single review from a restaurant dataset consists of multiple sentences (or a paragraph) where each sentence is labeled with aspect categories. Sometimes understanding an individual review sentence depends on the prior sentences. Incorporating the context in a sentence helps reduce the ambiguities caused by the presence of personal pronouns and inadequate information. We can solve this problem by replacing personal pronouns (e.g., it, they) with suitable reference (or noun) words from the contextual sentence(s). The link between personal pronouns and the context noun words can be mapped by using dependency parsing. Successful substitution of personal pronouns with proper meaningful words in a sentence will reduce the misclassification rate.

For example, Sentence 1, “Don’t leave the restaurant without it” is incorrectly labeled by our system which contains a personal pronoun “it” and Sentence 2, “Green Tea crème brulee is a must!” is the previous sentence of Sentence 1. In Sentence 2 the term, “Green Tea crème brulee” is referring to a personal pronoun in Sentence 1. So, if we substitute the term “it” with, “Green Tea crème brulee”, then Sentence 1 will look like, “Don’t leave the restaurant without Green Tea crème brulee” as shown in Figure 7. This type of pre-processing must be done before presenting a sentence to the system to avoid personal pronoun ambiguities.
Author Contributions: This research work is the collaborative effort of a team. The idea was conceived by N.D., H.M. and N.E. The simulations were carried out by N.D., I.M., K.M. and Y.N. The manuscript was written by N.D., H.M., I.M. and K.M. M.O.B. has discussed the idea in the context of smart cities and significantly improved the document after proof reading.

Funding: This research was funded by the Soonchunhyang University Research Fund and the MSIP (Ministry of Science, ICT and Future Planning), Korea, grant number IITP-2018-2014-1-00720 and The APC was funded by IITP-2018-2014-1-00720.

Acknowledgments: This work was supported by the Soonchunhyang University Research Fund and also supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2014-1-00720) supervised by the IITP (Institute for information & communications Technology Promotion).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Nam, T.; Pardo, T.A. Conceptualizing smart city with dimensions of technology, people, and institutions. In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, College Park, MD, USA, 12–15 June 2011; pp. 282–291.
2. Puron-Cid, G.; Gil-Garcia, J.; Zhang, J. Smart cities, smart governments and smart citizens: A brief introduction. Int. J. E Plan. Res. 2015, 4, iv–vii.
3. Dameri, R.P.; Camille, R.-S. Smart city and value creation. In Smart City; Dameri, R.P., Camille, R.-S., Eds.; Springer: Cham, Switzerland, 2014; pp. 1–12.
4. Fahim, M.; Baker, T.; Khattak, A.M.; Alfandi, O. Alert me: Enhancing active lifestyle via observing sedentary behavior using mobile sensing systems. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017.
5. Fahim, M.; Baker, T. Knowledge-Based Decision Support Systems for Personalized u-lifecare Big Data Services. In Current Trends on Knowledge-Based Systems; Giner, A.-H., Rafael, V.-G., Eds.; Springer: Cham, Switzerland, 2017; pp. 187–203.
6. Arunachalam, R.; Sarkar, S. The new eye of government: Citizen sentiment analysis in social media. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan, 14 October 2013; pp. 23–28.
7. Liu, B. Sentiment Analysis and Opinion Mining; Morgan & Claypool Publishers: Williston, VT, USA, 2012; Volume 5, pp. 1–167.
8. Pang, B.; Lee, L. Opinion Mining and Sentiment Analysis; Now Publishers Inc.: Hanover, MA, USA, 2008; Volume 2, pp. 1–135.
9. Pang, B.; Lee, L.; Viathyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2002; Volume 10, pp. 79–86.
10. Lu, B.; Ott, M.; Cardie, C.; Tsou, B.K. Multi-aspect sentiment analysis with topic models. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC, Canada, 11–14 December 2011; IEEE: New York, NY, USA; pp. 81–88.
11. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; ACM: New York, NY, USA; pp. 168–177.

12. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 1962; pp. 19–30.

13. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. Semeval-2015 task 12: Aspect based sentiment analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 486–495.

14. Qiu, G.; Liu, B.; Bu, J.; Chen, C. Opinion word expansion and target extraction through double propagation. Comput. Linguist. 2011, 37, 9–27. [CrossRef]

15. Alvarez-Lopez, T.; Costa-Montenegro, E.; González-Castano, F.J. GTI at Semeval-2016 task 5: SVM and CRF for aspect detection and unsupervised aspect-based sentiment analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 306–311.

16. Ganu, G.; Elhadad, N.; Marian, A. Beyond the Stars: Improving Rating Predictions Using Review Text Content. WebDB 2009, 9, 1–6. Available online: http://people.dbmi.columbia.edu/noemie/papers/webdb09.pdf (accessed on 2 January 2016).

17. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1532–1543.

18. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 16; Thrun, S., Saul, L.K., Schölkopf, B., Eds.; MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.

19. Titov, I.; McDonald, R. Modeling online reviews with multi-grain topic models. In Proceedings of the 17th International Conference on World Wide Web, Beijing, China, 21–25 April 2008; ACM: New York, NY, USA, 1947; pp. 111–120.

20. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. J. Mach. Learn. Res. 2003, 3, 993–1022.

21. Hofmann, T. Probabilistic latent semantic indexing. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 15–19 August 1999; ACM: New York, NY, USA, 1947; pp. 50–57.

22. Brody, S.; Elhadad, N. An unsupervised aspect-sentiment model for online reviews. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL), Los Angeles, CA, USA, 2–4 June 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 1962; pp. 804–812.

23. Zhao, W.X.; Jiang, J.; Yan, H.; Li, X. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 1962; pp. 56–65.

24. Joachims, T. A support vector method for multivariate performance measures. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; ACM: New York, NY, USA, 1947; pp. 377–384.

25. Heuer, H. Text Comparison Using Word Vector Representations and Dimensionality Reduction. arXiv, 2016, arXiv:1607.00534.

26. Socher, R.; Huval, B.; Manning, C.D.; Ng, A.Y. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; Association for Computational Linguistics: Stroudsburg, PA, USA, 1962; pp. 1201–1211.

27. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 1188–1196.

28. Zhou, X.; Wan, X.; Xiao, J. Representation learning for aspect category detection in online reviews. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
29. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.

30. Toh, Z.; Su, J. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 282–288.

31. Wang, B.; Liu, M. Deep Learning for Aspect-Based Sentiment Analysis. Available online: https://cs224d.stanford.edu/reports/WangBo.pdf (accessed on 2 April 2016).

32. Alghunaim, A.; Mohtarami, M.; Cyphers, S.; Glass, J. A vector space approach for aspect-based sentiment analysis. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics—Human Language Technologies (NAACL HLT 2015), Denver, CO, USA, 31 May–5 June 2015; pp. 116–122.

33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.

34. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1026–1034.

35. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv 2014, arXiv:1412.6980.

36. Rahim, R.; Sojka, P. Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010.

37. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv, 2016, arXiv:1603.04467.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).