Aspect-Based Sentiment Analysis Using Aspect Map

Yunseok Noh ¹, Seyoung Park ¹ and Seong-Bae Park ^{2,*}¹ School of Computer Science and Engineering, Kyungpook National University, Daegu 702-701, Korea² Department of Computer Science and Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, Korea

* Correspondence: sbpark71@khu.ac.kr

Received: 24 June 2019; Accepted: 5 August 2019; Published: 8 August 2019

Featured Application: Aspect-based sentiment analysis; Opinion mining.

Abstract: Aspect-based sentiment analysis (ABSA) is the task of classifying the sentiment of a specific aspect in a text. Because a single text usually has multiple aspects which are expressed independently, ABSA is a crucial task for in-depth opinion mining. A key point of solving ABSA is to align sentiment expressions with their proper target aspect in a text. Thus, many recent neural models have applied attention mechanisms to learning the alignment. However, it is problematic to depend solely on attention mechanisms to achieve this, because most sentiment expressions such as “nice” and “bad” are too general to be aligned with a proper aspect even through an attention mechanism. To solve this problem, this paper proposes a novel convolutional neural network (CNN)-based aspect-level sentiment classification model, which consists of two CNNs. Because sentiment expressions relevant to an aspect usually appear near the aspect expressions of the aspect, the proposed model first finds the aspect expressions for a given aspect and then focuses on the sentiment expressions around the aspect expressions to determine the final sentiment of an aspect. Thus, the first CNN extracts the positional information of aspect expressions for a target aspect and expresses the information as an aspect map. Even if there exist no data with annotations on direct relation between aspects and their expressions, the aspect map can be obtained effectively by learning it in a weakly supervised manner. Then, the second CNN classifies the sentiment of the target aspect in a text using the aspect map. The proposed model is evaluated on SemEval 2016 Task 5 dataset and is compared with several baseline models. According to the experimental results, the proposed model does not only outperform the baseline models but also shows state-of-the-art performance for the dataset.

Keywords: aspect-based sentiment analysis; class activation mapping; convolutional neural networks; opinion mining; weakly supervised learning

1. Introduction

Personal opinions are prevalent these days as considerable reviews and comments are available on various websites such as [IMDb.com](https://www.imdb.com), [Amazon.com](https://www.amazon.com) and [Yelp.com](https://www.yelp.com). The opinions in these texts are used in analyzing the reputation of movies, improving products, and providing a recommendation on personal items such as books or restaurants. Sentiment analysis is a well-known opinion-mining task, which determines whether a text contains a positive opinion or a negative one. Since the texts created by users are often noisy as they include a number of grammatical errors and emoticons, it is necessary to apply some proper text preprocessing to the texts before analyzing their sentiments [1]. Various sentiment analysis approaches have been developed upon the preprocessed texts such as lexicon-based methods [2–5], statistical machine learning methods [6–9], and recent deep learning methods [10–14].

In general, a single text contains multiple aspects and the aspects are expressed independently. Thus, for in-depth sentiment analysis, the sentiments of a text should be examined at aspect level,

not at text level. Several methods have been proposed for aspect-based sentiment analysis (ABSA) [15]. Because a target aspect is fixed in ABSA, it is important to properly exploit the information for the aspect within a text, as well as to understand the text. For understanding a text, deep learning has been proven to be a dominant method [10,16,17]. As a result, convolutional neural networks (CNNs) and bi-directional recurrent neural networks (bi-RNNs) are widely used as a text encoder in ABSA [18–21].

For exploiting the information of a target aspect in ABSA, Guan et al. employed the aspect vector of a target aspect [18], where the vector is generated by embedding aspect categories onto a continuous vector space. Then, they determined the aspect-level sentiment of a text with a concatenated vector of the aspect vector and a text vector. However, this simple application of the target aspect information has a weakness of dealing with all words in a text with the same importance. Many RNN-based ABSA models have solved this problem by adopting attention mechanisms [21–23]. Because attention mechanisms have an ability to focus on salient words for the task of interest, they have been used to generate a context vector that selectively embodies informative expressions for aspect-level sentiment classification.

The main problem of the context vector is that it loses positional information of salient words, even if the information is important in ABSA to know where the aspect expressions of a target aspect appear. For instance, Figure 1 shows an example review on a restaurant with two different target aspects and their corresponding sentiment expressions. In this figure, one target aspect is *LOCATION* represented with an aspect expression “view of river”, and the other is *FOOD* expressed as “sushi rolls”. Note that the positive opinion about *LOCATION* can be found by looking at the leftmost word “nice”. Similarly, the negative word “bad” for the aspect *FOOD* is found near the aspect expression “sushi rolls”. That is, the sentiment expressions appear near their aspect expressions. If a corpus in which aspect expressions and their aspect labels are annotated at word level is available, the positional information of the aspect expressions can be easily incorporated into the existing ABSA methods. However, such a corpus is not available in general, but only the corpora in which the sentiments are annotated at text level are ample. In addition, it is extremely tedious and labor-intensive to construct a corpus with word-level annotations.

Figure 1. An example of aspect-based sentiment analysis (ABSA). Sentiment expressions such as “nice” or “bad” appear near their related aspect expressions “view of river” and “sushi rolls,” respectively.

This paper proposes a novel deep neural network that consists of two CNNs, to exploit the proximity property between an aspect expression and its sentiment mention. The first CNN of the proposed model extracts an aspect map for a target aspect where the aspect map indicates significant words in a text with respect to the target aspect, and the other CNN determines the sentiment polarity of the target aspect by referring to the aspect map. Note that a corpus where the aspects are annotated at word level is needed to learn aspect map extraction. However, such a corpus is not available as stated above. Therefore, the aspect map should be extracted in a weakly supervised manner. Recently, the class activation map (CAM) has been proposed for weakly supervised object localization in the community of computer vision [24]. It localizes the objects of an image by linking the feature maps of CNN to relevant classes where the linking is implemented by global average pooling. Thus, the network supported by CAM can learn the localization of objects from the weakly labeled images, and is similar to the aspect map extraction from weakly labeled texts. The CNNs for text analysis usually have a single convolutional layer with various kernel sizes unlike those for image analysis [10].

Thus, they can capture n -gram word patterns for various ns . As a result, multiple CAMs are generated according to n and the number of different kernels. The aspect map for a target aspect in ABSA is a summation of all these CAMs.

After the aspect map is extracted, the second CNN classifies the sentiment of a target aspect. The convolutional layer of this CNN produces feature maps that activate the sentiment expressions such as “nice” and “bad” in Figure 1. Generally, in ABSA, only the sentiment expressions near the aspect expressions of a target aspect are under interest. Thus, the aspect map is applied to the feature maps of sentiment expressions to leave the target aspect-specific sentiment feature maps. Then, the second CNN determines the sentiment polarity of a text for the target aspect using the aspect-specific sentiment feature maps.

In the recent work of Xue and Li [20], a gated convolutional neural network is introduced for ABSA. Similar to our work, this model generates both sentiment and aspect features by using tanh gates and relu gates with aspect embeddings respectively. At every position, aspect features from relu gates are used to amplify sentiment features from tanh gates, thus the gated CNN tries to solve the problem of the context vector of attention mechanisms. Unlike this work which just utilizes aspect embeddings to generate aspect features and learns all features under the sentiment classification objective, the proposed model learns the relationship between words and aspects separately from the sentiment classification. Consequently, the proposed model can obtain a certain aspect map and can effectively exploit the aspect map for ABSA.

The rest of this paper is organized as follows. In Section 2, we briefly introduce previous work on recent neural models for aspect-based sentiment analysis. In Section 3, we describe the overall process of the proposed model, and then, in Section 4, we explain the detailed architectures and mechanisms of two CNNs used in the proposed model. The experimental results are given in Section 5. Finally, we conclude the study in Section 6.

2. Related Work

Aspect-based sentiment analysis has been regarded as an important task in opinion mining [15]. Because CNNs and RNNs have proven their superiority for text understanding without labor-intensive feature engineering, most recent studies lean toward deep learning. Early deep learning methods generally used a target aspect as a kind of context information for aspect-aware sentiment classification [18,25]. For the classification, the previous studies embedded a target aspect onto a context vector, and fed the vector to the classification layer together with a text vector encoded by a CNN or an RNN. However, because a text is encoded independently from the target aspect, these methods fail in detecting specific expressions for the target aspect.

Recently, attention mechanisms have been employed to solve this problem. In ATAE-LSTM [22], the aspect vector is concatenated to every word representation, and then an attention mechanism finds proper sentiment expressions of the aspect vector. The attention mechanism in ATAE-LSTM can capture the relation between aspects and aspect-specific sentiment words like *price/cheap*, but fails in classifying the text in Figure 1 due to general sentiment words such as “nice” and “bad”. Because general sentiment words are used with too many aspects, the attention mechanism cannot detect their correct aspect.

Gated convolutional network with aspect embedding (GCAE) [20] is a CNN-based model that includes two different gate units on top of two convolutional layers. It first generates two different feature maps from two convolutional layers, and then each feature map is delivered to the tanh gate and the ReLU gate, respectively. The ReLU gate receives additional information to control the propagation of sentiment features. After the outputs of the two gates are combined by element-wise multiplication, the combined features are delivered to the classification layer. The gating mechanism of the ReLU gate is applied only to n -gram local features. Thus, it is possible to detect aspect-specific sentiment features within the distance of kernel size from aspect expressions. However, the convolutional layer

with the ReLU gate in GCAE must match both aspect and sentiment expressions, which leads to performance deterioration.

Tang et al. demonstrated the usefulness of the positional information of aspect expressions in ABSA [19]. They introduced the location attention for ABSA in their study. To utilize the proximity between a sentiment expression and an aspect expression, the location attention is calculated by measuring the absolute distance of each word to the aspect expression. With this location attention, the words that appear near the aspect expression are more focused for sentiment classification. They were successful in showing the effectiveness of the positional information of aspect expressions. However, because their method requires aspect expressions for the target aspect as its input, it does not work when only text-level annotations are available.

Lee et al. applied class activation mapping to the task of sentiment classification [26]. They trained multi-channel CNNs on movie reviews, and then demonstrated sentiment-specific expressions mined by CAMs. Their method for obtaining CAMs is partially similar to the proposed model, but different in that the original CAM [24] was used as it is in their work. Thus, what they have shown is just the word localization ability of CAM. On the other hand, the proposed model adopts CAM to solve ABSA.

3. Sentiment Analysis Using Aspect Map

Aspect-based sentiment analysis (ABSA) is a task of making a sentiment decision s for a given text $\mathbf{t}$ and a target aspect c . For instance, if the very short review text in Figure 1 is considered as $\mathbf{t}$, it contains two aspects of *LOCATION* and *FOOD*. Thus, two input tuples should be considered for ABSA as in Figure 2. If the tuple $(\mathbf{t}, \text{LOCATION})$ is given to an ABSA system, then the system should output **positive** as a sentiment label, while **negative** is the right decision when the tuple is $(\mathbf{t}, \text{FOOD})$. In this paper, we divide the ABSA task into two pipelined tasks. That is, after an aspect map for an input tuple is extracted, sentiment classification is performed for the input tuple with the extracted aspect map.

Figure 2. The description of the ABSA task with two different tuples from a single review text.

Figure 3 describes the overall architecture of the proposed model, which consists of two CNNs. The first CNN is aspect-map extraction network (AMEN). The goal of AMEN is to identify which parts of a text $\mathbf{t}$ are actually related with a target aspect c . Thus, the result of AMEN is an aspect map $\mathcal{A}^c$ for the target aspect c that is extracted from $\mathbf{t}$. That is, AMEN takes both $\mathbf{t}$ and c as its input, and returns $\mathcal{A}^c$ as its output. The aspect map $\mathcal{A}^c \in \mathbb{R}^N$ indicates the relevances of all words in $\mathbf{t}$ to the aspect c , where N is the length of $\mathbf{t}$. Then, the words with a high relevance become an aspect expression for c . For instance, the words “sushi” and “rolls” are extracted as an aspect expression for *FOOD* in Figure 3 by applying the aspect map $\mathcal{A}^{\text{FOOD}}$ to $\mathbf{t}$.

The other CNN is aspect map-based sentiment analysis network (AMSAN), which classifies the sentiment polarity of the text $\mathbf{t}$ for the target aspect c . Because AMSAN has to know what is the aspect expression for c in $\mathbf{t}$ for correct classification, it takes, as its input, the aspect map $\mathcal{A}^c$ as well as $\mathbf{t}$ and c . The aspect map $\mathcal{A}^c$ is a type of positional filter that leaves a sentiment expression appearing near the aspect expression for c . For example, in Figure 3, the word “bad” survives and affects the process of sentiment classification, while the word “nice” is excluded by $\mathcal{A}^c$ because it is far from the aspect expression “sushi rolls.”

Figure 3. Overall process to determine the aspect-level sentiment polarity by the proposed model. Background colors depict an aspect map that indicates the relevance of each word to the aspect *FOOD*. Blue color means that a word is almost irrelevant to *c*. Red colors mean that a word is relevant to *c* as strong as the color strength.

4. Learning AMEN and AMSAN

4.1. Aspect-Map Extraction Network (AMEN)

The aspect-map extraction network (AMEN) aims to extract an aspect map that indicates which parts of a text are relevant to a given aspect. However, it cannot be trained directly to return the aspect map because the training set for ABSA has only the information about the aspects *cs* and their corresponding sentiment polarities *ss* of a text *t*, but not any information about the relevancy of words in *t* to *cs*. Therefore, AMEN should learn the relevant words in a weakly supervised manner, while its actual function is to map *t* to *cs*.

Figure 4 depicts the architecture of AMEN. AMEN has four layers of a word embedding layer, a convolutional layer, a global average pooling layer and a classification layer. The word embedding layer maps every word $w \in \mathbf{t}$ to its corresponding word vector $\mathbf{v} \in \mathbb{R}^D$, where D is the dimension of word embedding vectors. Because $\mathbf{t}$ is assumed to have N words, the word embedding layer returns a matrix $\mathbf{X} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_N^T]$ as its output. After $\mathbf{X}$ is obtained, the convolutional layer represents it as a matrix of feature values using kernels with various sizes. There are usually K kernels for every size $h \in H$ in the convolutional layer, and each kernel $\mathbf{W}_{h_k} \in \mathbb{R}^{D \times h}$ ($1 \leq k \leq K$) is applied to a window of h words. Then, the feature value $z_i^{h_k}$ for the i -th word window $\mathbf{X}_{i:i+h-1}$ by a kernel $\mathbf{W}_{h_k}$ is computed as

$$z_i^{h_k} = f(\mathbf{W}_{h_k} * \mathbf{X}_{i:i+h-1} + b_{h_k}), \quad (1)$$

where $f(\cdot)$ is a non-linear function such as tanh or ReLU, $b_{h_k} \in \mathbb{R}$ is a bias and $*$ denotes the convolution operation. Thus, the feature map obtained by a kernel $\mathbf{W}_{h_k}$ is represented as

$$\mathbf{z}^{h_k} = \langle z_1^{h_k}, z_2^{h_k}, \dots, z_{N-h+1}^{h_k} \rangle. \quad (2)$$

Because there are K kernels for every size h , $K \cdot |H|$ feature maps are produced from the convolutional layer. To make the length of all feature maps equal to that of the input text *t*, the *half padding* [27] is applied for all convolutional operations. As a result, the final output of the convolutional layer is a matrix $\mathbf{Z} \in \mathbb{R}^{(K \cdot |H|) \times N}$ whose row vector is $\mathbf{z}^{h_k}$.

The global average pooling (GAP) is then applied to each row vector $\mathbf{z}^{h_k}$ of $\mathbf{Z}$, and a feature value $\hat{z}^{h_k}$ is produced for every $\mathbf{z}^{h_k}$ by summing all $z_i^{h_k}$ s. That is,

$$\hat{z}^{h_k} = \sum_{i=1}^N z_i^{h_k}. \quad (3)$$

Because $K \cdot |H|$ feature values are generated by GAP, the final feature vector $\mathbf{g}$ has $K \cdot |H|$ dimensions. Then, the classification layer finally predicts the vector of aspect categories by

$$\mathbf{y} = f(\mathbf{W}_y \cdot \mathbf{g} + \mathbf{b}_y), \quad (4)$$

where $\mathbf{W}_y \in \mathbb{R}^{C \times (K \cdot |H|)}$ is a weight matrix and $\mathbf{b}_y \in \mathbb{R}^C$ is a bias vector.

Figure 4. The architecture of aspect-map extraction network (AMEN). The convolutional layer in this figure has three kernels of size $h = 1, 2$ and 3 .

AMEN is trained with a dataset $\mathcal{D}_a = \{(\mathbf{t}, \mathbf{c}^t)\}$ extracted from the ABSA dataset $\mathcal{D}_s$. Figure 5 shows how to prepare $\mathcal{D}_a$ from $\mathcal{D}_s$. Note that a text $\mathbf{t}$ can appear several times in $\mathcal{D}_s$ with different aspects. On the other hand, $\mathcal{D}_a$ represents $\mathbf{t}$ only once. Thus, $\mathcal{D}_a$ is a compact representation of $\mathcal{D}_s$ between a text $\mathbf{t}$ and its aspects $\mathbf{c}$ s, where $\mathbf{c}^t$ is a binary vector that indicates all $\mathbf{c}$ s for $\mathbf{t}$ as 1s but otherwise 0s. Then, θ_{AMEN} , a set of all parameters of AMEN, is trained to minimize

$$\sum_{(\mathbf{t}, \mathbf{c}^t) \in \mathcal{D}_a} \mathcal{L}(\mathbf{c}^t, \text{AMEN}(\mathbf{t}; \theta_{\text{AMEN}})), \quad (5)$$

where $\mathcal{L}$ is the cross-entropy loss.

Note that $\hat{z}^{h_k}$ in Equation (3) is the representative of a feature map $\mathbf{z}^{h_k}$ and is directly connected to all aspects. Thus, the aspect map $\mathcal{A}^c$ of a target aspect c becomes

$$\mathcal{A}^c = \sigma(\mathbf{w}_y^c \cdot \mathbf{Z}), \quad (6)$$

where $\mathbf{w}_y^c$ is a row vector of $\mathbf{W}_y$ for an aspect c and $\sigma(\cdot)$ is the sigmoid function. For example, let us consider Figure 6, where the target aspect is *LOCATION*. AMEN of this figure has $H = \{1, 2, 3\}$ and $K = 1$. Then, $\mathcal{A}^{\text{LOCATION}}$ is

$$\mathcal{A}^{\text{LOCATION}} = \sigma(w^{11} \cdot \mathbf{z}^{11} + w^{21} \cdot \mathbf{z}^{21} + w^{31} \cdot \mathbf{z}^{31}).$$

Assume that $\mathbf{z}^{1_1}$ is the feature map that is most relevant to *LOCATION* and $\mathbf{z}^{2_1}$ is the most irrelevant feature map, as shown by the width of weights w^{1_1} , w^{2_1} and w^{3_1} . $\mathcal{A}^{LOCATION}$ indicates “view of river” as an aspect expression for *LOCATION* because “view” and “river” are activated by the strongly relevant feature map $\mathbf{z}^{1_1}$ and “of” is activated by the weakly relevant feature map $\mathbf{z}^{3_1}$. Although “sushi” and “rolls” are activated strongly by $\mathbf{z}^{2_1}$, they are ignored due to the small value of w^{2_1} .

One problem of using the aspect map in aspect-level sentiment analysis is that the aspect map could be noisy. In practice, the noises in the aspect map are represented as strong activation of the words irrelevant to a target aspect. As a result, too many words are indicated as an aspect expression. This phenomenon is observed in all applications of CAM, since CAM is obtained by a weakly-supervised manner. The noise in the aspect map could be severer, since AMEN is trained with multi-labeled examples $(\mathbf{t}, \mathbf{c}^t)$ s. To deal with this problem, the loss function for AMEN in Equation (5) is modified as

$$\sum_{(\mathbf{t}, \mathbf{c}^t) \in \mathcal{D}_a} \mathcal{L}(\mathbf{c}^t, \text{AMEN}(\mathbf{t}; \theta_{\text{AMEN}})) + \lambda \|\mathcal{A}^t\|_1. \quad (7)$$

Here, $\mathcal{A}^t \in \mathbb{R}^{C \times N}$ is the matrix $\mathbf{W}_y \cdot \mathbf{Z}^t$ where $\mathbf{Z}^t$ is the feature matrix $\mathbf{Z}$ for an input text $\mathbf{t}$.

The added L_1 regularizer leads the aspect map $\mathcal{A}^t$ to be sparse. Since AMEN is trained to jointly minimize the cross-entropy loss $\mathcal{L}$ and the L_1 regularizer $\|\mathcal{A}^t\|_1$ by Equation (7), its aspect map can indicate aspect expressions more precisely. That is, the words which actually contribute to classify $\mathbf{t}$ into $\mathbf{c}^t$ are activated by the loss, and all other words are forced to be deactivated by the regularizer. Note that excessive regularization on $\mathcal{A}^t$ may kill the actual aspect expressions in the aspect map, which leads to a poor aspect map and the classification performance degradation. Thus, the impact of the regularizer should be controlled by setting λ to an appropriate value.

Figure 5. $\mathcal{D}_a$ expresses $\mathcal{D}_s$ more compactly by incorporating all aspects for the same text $\mathbf{t}$.

Figure 6. Extraction of an aspect map from AMEN. Here, the aspect map for *LOCATION* is extracted.

4.2. Aspect Map-Based Sentiment Analysis Network (AMSAN)

The aspect map-based sentiment analysis network (AMSAN) is trained with $\mathcal{D}_s$, and classifies the sentiment polarity s of a text $\mathbf{t}$ for a target aspect c using the aspect map $\mathcal{A}^c$. The architecture of AMSAN to fulfill this task is shown in the left side of Figure 7. AMSAN consists of a word embedding layer, two convolutional layers, a fully connected layer and a classification layer. Its first two layers (the word embedding layer and the first convolutional layer) are identical to those of AMEN. Thus, the output of these layers is $\mathbf{Z}$ whose elements are $\mathbf{z}^{h_k}$ s in Equation (2).

Figure 7. Architecture of aspect map-based sentiment analysis network (AMSAN). Aspect map for LOCATION is used to identify a neighboring word “nice” as an aspect-specific sentiment expression. For this, local max pooling is adopted for both the aspect map and convolutional feature map of AMSAN.

The aspect map $\mathcal{A}^c$ obtained from AMEN is then applied to $\mathbf{Z}$. This application captures sentiment expressions because AMSAN is trained to classify sentiment polarity. To transform the positional information of the aspect expressions in $\mathcal{A}^c$ to regional information, the local max pooling (LMP) with size p is applied to $\mathcal{A}^c$. By this operation, the aspect region map $\mathcal{R}^c \in \mathbb{R}^{\lceil \frac{N}{p} \rceil}$ is produced. Then, $\mathbf{Z}$ is also reduced to $\mathbf{Z}^p$ by LMP with the identical pooling size p to scale down $\mathbf{Z}$ to the length of $\mathcal{R}^c$. After that, the sentiment expressions of $\mathbf{t}$ for c are obtained by the element-wise multiplication of $\mathcal{R}^c$ and $\mathbf{Z}^p$. That is, $\mathbf{Z}^{\mathcal{R}^c}$, the matrix for sentiment expressions for c , is computed by

$$\mathbf{Z}^{\mathcal{R}^c} = \mathbf{Z}^p \odot \mathcal{R}^c, \quad (8)$$

where $\odot$ is the element-wise multiplication operator. Figure 7 explains these operations with an example text $\mathbf{t}$. In this figure, $\mathbf{Z}$ activates the sentiment expressions “nice” and “bad” in $\mathbf{t}$, and the phrase “view of river” is indicated as an aspect expression for LOCATION by $\mathcal{A}^{\text{LOCATION}}$. However, only the sentiment expression “nice” remains in $\mathbf{Z}^{\mathcal{R}^c}$ because $\mathbf{Z}^p$ is affected locally by $\mathcal{R}^c$.

$\mathbf{Z}^{\mathcal{R}^c}$ is delivered to the second convolutional layer. After the operation of Equation (1) is applied to $\mathbf{Z}^{\mathcal{R}^c}$, the global max pooling is performed to the output of the convolutional layer. The result of this layer is $\mathbf{g}$, which is the globally max-pooled vector. To incorporate the target aspect c into

AMSAN, $\mathbf{g}$ is concatenated with $\mathbf{a}^c$, where $\mathbf{a}^c$ is a D -dimensional word embedding vector of c . Then, the concatenated vector is fed to the fully connected layer to induce a higher-level representation $\mathbf{f} \in \mathbb{R}^F$. That is,

$$\mathbf{f} = f(\mathbf{W}_f \cdot [\mathbf{g}; \mathbf{a}^c] + \mathbf{b}_f), \quad (9)$$

where $\mathbf{W}_f \in \mathbb{R}^{(K \cdot |H| + D) \times F}$ is a weight matrix of the fully connected layer and $\mathbf{b}_f$ is a bias vector of the layer. After all these layers, the classification layer finally determines the sentiment polarity of $\mathbf{t}$ for the target aspect c by

$$\mathbf{y} = f(\mathbf{W}_y \cdot \mathbf{f} + \mathbf{b}_y). \quad (10)$$

Here, $\mathbf{W}_y$ and $\mathbf{b}_y$ indicate a weight matrix and a bias for the classification layer, respectively. In Figure 7, this layer determines the sentiment of $\mathbf{t}$ as *positive* by seeing “nice.” A set of all parameters of AMSAN, θ_{AMSAN} , is trained by minimizing

$$\sum_{(\mathbf{t}, c, s) \in \mathcal{D}_s} \mathcal{L}(s, \text{AMSAN}(\mathbf{t}, c, \mathcal{A}^c; \theta_{\text{AMSAN}})), \quad (11)$$

where $\mathcal{L}$ is the cross-entropy loss.

5. Experiments

5.1. Dataset

The evaluation of the proposed model is conducted on the datasets of SemEval-2014 Task 4 [28] and SemEval-2016 Task 5 [29]. SemEval-2014 Task 4 is a widely used benchmark dataset for ABSA [19,20,22]. However review texts in this dataset are divided into short sentences. As a result, about 90% of instances in this dataset has only 1 aspect. Recently, Xue and Li [20] showed extra results on hard instances that include more than two aspects. We follow them for the purpose of comparison and showing the effectiveness of the proposed model on hard instances. SemEval 2016 Task 5 dataset consists of two subtasks: sentence-level ABSA and text-level ABSA. Both subtasks share two domain datasets, which are a restaurant dataset and a laptop dataset. In the restaurant dataset, the average number of aspects per training instance in the sentence level is 1.26, while that of the text level is 4.28. Similarly, in the laptop dataset, it is 1.16 in the sentence level and 5.27 in the text level. Note that all text-level instances include multiple aspects. Since sentence-level ABSA is performed using SemEval-2014 dataset, for SemEval-2016 dataset we only focus on text-level ABSA which is more challenging and realistic.

Table 1 summarizes the statistics on the datasets used for the evaluation of the proposed model. The restaurant dataset of SemEval-2014 contains 3041 training instances and 800 test instances with 5 kinds of aspects. After representing this dataset as $\mathcal{D}_s = \{(\mathbf{t}, c, s)\}$ as explained in Section 3, $\mathcal{D}_s$ has 3713 training instances and 1025 test instances. The reason why the number of instances increases is that a single text $\mathbf{t}$ can have multiple c s and s s. Note that AMEN is trained with $\mathcal{D}_a$ which is a compact representation of $\mathcal{D}_s$, while AMSAN is trained with $\mathcal{D}_s$. The number of instances in $\mathcal{D}_a$ is equal to that of the dataset. That is, the number of training instances in $\mathcal{D}_a$ is 3041 and that of test instances is 800. The statistics of restaurant and laptop datasets of SemEval-2016 can also be calculated in the same manner and are shown in Table 1. There are 12 and 81 kinds of aspects in the restaurant and laptop dataset respectively. In $\mathcal{D}_s$ of all datasets, there are four sentiment polarities of *positive*, *negative*, *neutral* and *conflict*. Some previous studies on ABSA excluded the *conflict* polarity [21,22,30] because the number of instances for it is very small. However, this exclusion makes it difficult to compare the experimental result of the proposed model with those of the best models in SemEval-2016 Task 5. Thus, we conduct both experiments with and without *conflict*. Following [20], we report ‘with *conflict*’ performances and ‘on *hard* instance’ performances for SemEval-2014 benchmark.

Table 1. Simple statistics on the number of instances in the datasets. Numbers in parentheses are average number of aspects per review.

2014-Restaurant		Train				Test			
AMEN		3041 (1.22)				800 (1.28)			
AMSAN	Positive 2179	Negative 839	Neutral 500	Conflict 195	Positive 657	Negative 222	Neutral 94	Conflict 52	
2016-Restaurant		Train				Test			
AMEN		335 (4.28)				90 (4.49)			
AMSAN	Positive 1012	Negative 327	Neutral 55	Conflict 41	Positive 286	Negative 84	Neutral 23	Conflict 11	
2016-Laptop		Train				Test			
AMEN		395 (5.27)				80 (6.68)			
AMSAN	Positive 1210	Negative 708	Neutral 123	Conflict 41	Positive 330	Negative 161	Neutral 31	Conflict 13	

5.2. Model Training

Because the datasets are relatively small to train deep learning models, two well-known techniques are employed. One is transfer learning. It is known that pre-training a neural network with a large amount of labeled or unlabeled data from similar tasks is helpful in training it with a small dataset [31–33]. Thus, for each dataset, a CNN with the same architecture with AMEN is pre-trained. The 300-dimensional GloVe vectors [34] are used to initialize the word embedding layer of the CNN, and then the CNN is trained with the 4M Yelp restaurant reviews (<https://www.yelp.com/dataset/challenge>) for the restaurant dataset and 3M Amazon Electronics reviews (We used only the reviews of the categories related to *Laptop* and *Computer* in Amazon Electronics.) [35] for the laptop dataset. Because aspect labels are not available in these reviews, only the weights of the word embedding layer of the pre-trained CNN are used to initialize those of AMEN. For AMSAN, the weights of the word embedding layer and the first convolutional layer are initialized with those of the pre-trained CNN. The other technique used is regularization. Several regularization techniques to prevent deep neural networks from overfitting to training data have been proposed [31]. Among them, the dropout [36] is applied to the output of the convolutional layers of both AMEN and AMSAN as well as the fully connected layer of AMSAN. In addition, L_2 regularization is also applied to the output of the classification layers of both AMEN and AMSAN.

The softmax activation and categorical cross-entropy loss are used for both AMEN and AMSAN, because they show empirically better performance than the sigmoid activation and the binary cross-entropy loss in this task. The filter size h is (1, 2, 3) for AMEN and (3, 4, 5) for AMSAN, and the number of kernels K is set to 100 for all h s. The dimension of the fully connected layer of AMSAN is 100. The AdaGrad [37] is used for both AMEN and AMSAN with a batch size of 16 instances and the default learning rate of $1e-2$. The maximum number of epochs is 30 for AMEN and 20 for AMSAN. Because training both networks has randomness caused by random initialization and dropout, the experiments are conducted 30 times and then the mean accuracy with standard deviation is reported. All neural networks in this paper are implemented with Keras [38].

5.3. Experimental Results

The following four models are used as baselines to validate the proposed model.

- **UWB [39]:** This is the best model in SemEval 2016 Task 5 for both text-level restaurant and laptop datasets. It uses a large number of hand-crafted features. After representing all sentences in a dataset with a set of hand-crafted features, it classifies the sentiment of every sentence by the maximum entropy classifier. Then, the sentiment of a text is determined by the majority vote of sentence sentiments. Because the datasets for SemEval 2016 are small, top-ranked models in SemEval 2016 seldom adopt a deep learning method. This baseline is only for SemEval-2016 benchmark.
- **ATAE-LSTM [22]:** This is an attention-based LSTM for ABSA, where the aspect vector for a target aspect is concatenated to all word embeddings in a text. Then, the word embeddings are used as an input of an LSTM and an attention layer is located on the top of the LSTM. Because the performance of ATAE-LSTM is not reported on SemEval 2016 Task 5, we implement it ourselves for comparison with the proposed model.
- **GCAE [20]:** This is a CNN-based model for ABSA which includes the gated Tanh-ReLU units for capturing aspect-specific sentiment expressions from convolutional layers. The authors open their own code (<https://github.com/wxue004cs/GCAE>), and thus all experimental results on GCAE below are obtained using the code.
- **AMSAN w/o aspect map:** Because the aspect vector of a target aspect is fed to the fully connected layer of AMSAN, AMSAN can operate without an aspect map from AMEN. Thus, the impact of the aspect map can be judged by comparing *AMSAN without aspect map* with the proposed model.

Table 2 reports the performances of all five models on the restaurant dataset. As shown in this table, the proposed model (AMEN-AMSAN) outperforms all baselines in both cases of with and without the sentiment polarity *conflict*. When *conflict* is considered, the proposed model achieves 3% higher accuracy than UWB. Note that UWB is not an easy baseline in that it outperforms GCAE. ATAE-LSTM achieves the second-best performance with 82.699% mean accuracy. Even when the *conflict* is ignored, the proposed model achieves the best performance with an accuracy of 86.578%, which is 1% higher than that of ATAE-LSTM and almost 3% higher than that of GCAE. AMSAN without the aspect map achieves relatively good performance with accuracies of 82.309% with *conflict* and 85.394% without *conflict*. However, these accuracies are much lower than those of AMEN-AMSAN, which proves the impact of the aspect map in text-level ABSA.

Table 2. Performance comparison of the proposed model with its baselines on the restaurant dataset of SemEval-2016.

Model	Acc. w/ Conflict	Acc. w/o Conflict
UWB	81.931 (-)	-
ATAE-LSTM	82.699 (0.996)	85.496 (0.924)
GCAE	81.640 (0.621)	83.606 (0.756)
AMSAN w/o aspect map	82.309 (0.546)	85.394 (0.458)
AMEN-AMSAN	83.936 (0.496)	86.578 (0.407)

The proposed AMEN-AMSAN shows the best performance for the laptop dataset as well, as shown in Table 3. The proposed model improves UWB by 5.3% higher accuracy and outperforms both ATAE-LSTM and GCAE by 1.7% higher accuracies when *conflict* is considered. Note that all deep learning models outperform UWB for this dataset. That is, both ATAE-LSTM and GCAE achieve nearly 3.6% higher accuracies than UWB. Without *conflict*, ATAE-LSTM and GCAE achieve the accuracies of 79.685% and 80.812%, respectively. However, they do not reach the performance of the proposed model which achieves an accuracy of 82.201%, and this is higher than that of ATAE-LSTM by 2.5% and that of GCAE by 1.4%. AMSAN without the aspect map also shows comparable results in both

cases of with and without *conflict*, which implies that AMSAN is a good model for text-level ABSA. However, the fact that AMEN-AMSAN is superior to AMSAN without the aspect map proves the effectiveness of the aspect map.

Tables 2 and 3 indicate that deep learning models work well for text-level ABSA despite the relatively small datasets and exclusion of feature engineering. This is because deep learning models can produce deep embedding feature vectors and make context-aware decisions for a given target aspect through an attention-like mechanism. The proposed model improves such deep learning models by adopting the aspect map which reflects the positional information of aspect expressions.

Table 3. Performance comparison of the proposed model with its baselines on the laptop dataset of SemEval-2016.

Model	Acc. w/ Conflict	Acc. w/o Conflict
UWB	75.046 (-)	-
ATAE-LSTM	78.692 (0.747)	79.685 (1.197)
GCAE	78.659 (0.812)	80.812 (0.609)
AMSAN w/o aspect map	79.719 (0.920)	81.797 (1.100)
AMEN-AMSAN	80.399 (0.555)	82.201 (0.488)

Table 4 shows the performances on the SemEval-2014 dataset that consists of sentence-level reviews with relatively small aspects. As a result, all four models achieve similar performances over the dataset. This is because most of reviews have only one aspect. As a result, the ability to identify aspect-aware sentiment features of each model is not clearly discriminated. Nevertheless, the superiority of the proposed model can be identified in this table, since it outperforms all baselines significantly on the hard test set.

Table 4. Performance comparison of the proposed model with its baselines on the restaurant dataset of SemEval-2014.

Model	Acc. on Whole Test	Acc. on Hard Test
ATAE-LSTM	74.87 (1.46)	49.89 (1.88)
GCAE	74.13 (0.72)	46.29 (1.93)
AMSAN w/o aspect map	74.59 (0.78)	46.52 (1.28)
AMEN-AMSAN	75.59 (0.51)	53.26 (0.41)

5.4. Quality of Aspect Map

To see how the aspect map is improved by the L_1 regularizer $\|\mathcal{A}^t\|_1$ in Equation (7), two aspect maps for three aspects are shown in Figure 8 when a restaurant review is given. The three aspects are *FOOD#QUALITY*, *AMBIENCE#GENERAL* and *FOOD#PRICES*. First, let us see the aspect maps in Figure 8a which are not regularized yet by the proposed L_1 regularizer. In the aspect map for *FOOD#QUALITY*, the phrases “restaurant,” “The pizza is,” and “the atmosphere. But the pizza is” are strongly highlighted. Among them, the expression of “the pizza is” is relevant to the target aspect *FOOD#QUALITY*, but the other expressions such that “restaurant” and “the atmosphere.” are not much related to the target aspect. In the aspect map for *AMBIENCE#GENERAL*, the phrases of “a fun restaurant” and “the atmosphere. But” are correctly specified as the aspect expressions of *AMBIENCE#GENERAL*, while “yummy” is also activated though it is not related to *AMBIENCE#GENERAL*. Similarly, the aspect map for *FOOD#PRICES* focuses well on the expression of “way too expensive,” but it also focuses badly on the expression of “is yummy.” As seen in these examples, the aspect maps basically find true aspect expressions for their target aspects, but false aspect expressions are also often highlighted by the aspect maps. It is obvious that such noises prevent AMSAN from classifying sentiment of a target aspect correctly.

Figure 8. Aspect maps for three aspect categories over a restaurant review.

On the other hand, Figure 8b shows the aspect expressions of the aspect maps penalized by the proposed regularizer. The expressions in this figure are activated more sparsely but more precisely than those in Figure 8a. The aspect map for *FOOD#QUALITY* no longer highlights the expressions “restaurant” and “the atmosphere”, but newly highlights the expression “yummy.” In the aspect map for *AMBIENCE#GENERAL*, it can be observed that the activation of the expression “yummy” is weakened compared to the original aspect map in Figure 8a. Finally, “is yummy,” the strongly specified expression in the original aspect map for *FOOD#PRICES*, is also deactivated in the regularized aspect map. Due to the refined aspect maps, AMSAN is able to classify the sentiment polarity of a text correctly for a given target aspect. For instance, the expression “the atmosphere. But” helps AMSAN classify the review in Figure 8 as *positive* for the target aspect *AMBIENCE#GENERAL*, because it appears near the word “like” which AMSAN concentrates on.

Another conspicuous characteristic of the aspect map is that it tends to highlight the sentiment expressions as well as aspect expressions. For instance, “the atmosphere” and “fun” are highlighted by the aspect map for *AMBIENCE#GENERAL*, where “fun” is a sentiment expression and “the atmosphere” is an aspect expression. The review in this figure has a positive polarity toward ambience because it has a clause “I like the atmosphere.” This positive polarity about ambience can also be inferred from the expression “fun” on which the aspect map focuses. Similarly, “yummy,” and “expensive” are also sentiment expressions highlighted by the aspect maps for *FOOD#QUALITY* and *FOOD#PRICES* respectively. As a result, it can be concluded that the aspect map helps AMSAN classify the aspect-based sentiment polarity of a text accurately because sentiment expressions are double-checked by both AMEN and AMSAN.

In summary, the proposed model achieves good results in text-level ABSA tasks. To the best of our knowledge, the results in Tables 2 and 3 are state-of-the-art performances of text-level subtasks of SemEval 2016 Task 5. These good performances are accomplished by exploiting aspect maps for target aspects. In addition, the proposed model localizes the aspect expressions without additional annotation on training data because the aspect maps are obtained in a weakly supervised manner. Nevertheless, the quality of the aspect map is reliable since the map is regularized for ABSA.

6. Conclusions and Future Work

This paper proposes a novel neural model that utilizes the positional information of aspect expressions for ABSA. The proposed model consists of two CNNs, where one CNN is for extracting an aspect map for a given text and a target aspect and the other is for classifying the sentiment polarity of the text based on the extracted aspect map. Because sentiment expressions tend to appear near the aspect expressions of a target aspect, the proposed aspect map plays an important role in aspect-based sentiment classification. In the experiments on text-level SemEval 2016 ABSA tasks, the proposed model achieves state-of-the-art performances.

The proposed model still has room to improve. Since it deals a whole text as a sequence of words, both left and right sentiment expressions of a highlighted aspect expression have the same contribution to the final classification. To deal with this concern, we can exploit the hierarchical structure of a text which allows the proposed model to restrict the alignment of aspect expressions and sentiment expressions in the same sentence. Furthermore, it is believed that sentiment maps can be extracted as

well as aspect maps. Then, learning to align two kinds of maps becomes another research direction of ours. We remain these ideas as our future work.

It is worthy to note that recent deep learning approaches are inexplicable generally for the decisions they make [40]. The proposed model also has such inherent weakness in explainability, though it is somewhat interpretable by looking over the aspect map. Developing a model re-traceable as well as explainable is a major concern in the community and is also our future work.

Author Contributions: Conceptualization, Y.N. and S.-B.P.; Funding acquisition, S.P.; Methodology, Y.N.; Project administration, S.P.; Supervision, S.P. and S.-B.P.; Validation, Y.N. and S.-B.P.; Visualization, Y.N.; Writing—original draft, Y.N.; Writing—review & editing, S.P. and S.-B.P.

Funding: This research received no external funding.

Acknowledgments: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform).

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

References

1. Petz, G.; Karpowicz, M.; Fürschuß, H.; Auinger, A.; Winkler, S.M.; Schaller, S.; Holzinger, A. On text preprocessing for opinion mining outside of laboratory environments. In *International Conference on Active Media Technology*; Springer: Berlin, Germany, 2012; pp. 618–629.
2. 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; pp. 168–177.
3. Turney, P.D. Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA, 7–12 July 2002; pp. 417–424.
4. Kim, S.; Hovy, E.H. Determining the Sentiment of Opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 23–27 August 2004.
5. Ding, X.; Liu, B.; Yu, P.S. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining*, Palo Alto, CA, USA, 11–12 February 2008; pp. 231–240.
6. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, USA, 6–7 July 2002.
7. Dave, K.; Lawrence, S.; Pennock, D.M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungary, 20–24 May 2003; pp. 519–528.
8. Mullen, T.; Collier, N. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 15 January 2004; pp. 412–418.
9. Jo, Y.; Oh, A.H. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining*, Hong Kong, China, 9–12 February 2011; pp. 815–824.
10. Kim, Y. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
11. Socher, R.; Pennington, J.; Huang, E.H.; Ng, A.Y.; Manning, C.D. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Scotland, UK, 27–31 July 2011; pp. 151–161.
12. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.

13. Glorot, X.; Bordes, A.; Bengio, Y. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In Proceedings of the 28th International Conference on Machine Learning, Omnipress, Bellevue, WA, USA, 28 June–2 July 2011; pp. 513–520.
14. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.J.; Hovy, E.H. Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
15. Feldman, R. Techniques and applications for sentiment analysis. *Commun. ACM* **2013**, *56*, 82–89. [[CrossRef](#)]
16. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P.P. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
17. Tang, D.; Qin, B.; Liu, T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 17–21 September 2015; pp. 1422–1432.
18. Guan, Z.; Chen, L.; Zhao, W.; Zheng, Y.; Tan, S.; Cai, D. Weakly-Supervised Deep Learning for Customer Review Sentiment Classification. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 3719–3725.
19. Tang, D.; Qin, B.; Liu, T. Aspect Level Sentiment Classification with Deep Memory Network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 214–224.
20. Xue, W.; Li, T. Aspect Based Sentiment Analysis with Gated Convolutional Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 2514–2523.
21. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive Attention Networks for Aspect-Level Sentiment Classification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4068–4074.
22. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
23. Cheng, J.; Zhao, S.; Zhang, J.; King, I.; Zhang, X.; Wang, H. Aspect-level Sentiment Classification with HEAT (HiERarchical ATtention) Network. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 97–106.
24. Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
25. Ruder, S.; Ghaffari, P.; Breslin, J.G. INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 330–336.
26. Lee, G.; Jeong, J.; Seo, S.; Kim, C.; Kang, P. Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. *Knowl.-Based Syst.* **2018**, *152*, 70–82. [[CrossRef](#)]
27. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
28. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 27–35.
29. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; Clercq, O.D.; et al. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
30. Chen, P.; Sun, Z.; Bing, L.; Yang, W. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 452–461.
31. Goodfellow, I.J.; Bengio, Y.; Courville, A.C. *Deep Learning*; Adaptive Computation and Machine Learning; MIT Press: Cambridge, MA, USA, 2016.

32. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014; NIPS: Grenada, Spain, 2014*; pp. 3320–3328.
33. Mou, L.; Meng, Z.; Yan, R.; Li, G.; Xu, Y.; Zhang, L.; Jin, Z. How Transferable are Neural Networks in NLP Applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, 1–5 November 2016; pp. 479–489.
34. 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*, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
35. He, R.; McAuley, J. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web*, Montreal, QC, Canada, 11–15 April 2016; pp. 507–517.
36. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
37. Duchi, J.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
38. Chollet, F. Keras. 2015. Available online: <https://github.com/fchollet/keras> (accessed on 7 August 2019).
39. Hercig, T.; Brychcín, T.; Svoboda, L.; Konkol, M. UWB at SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, San Diego, CA, USA, 16–17 June 2016; pp. 342–349.
40. Holzinger, A. From machine learning to explainable AI. In *Proceedings of the 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, Košice, Slovakia, 23–25 August 2018; pp. 55–66.

© 2019 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/>).