electronics

MDPI

Article

Extracting Prominent Aspects of Online Customer Reviews: A Data-Driven Approach to Big Data Analytics

Noaman M. Ali [1,*], Abdullah Alshahrani [2], Ahmed M. Alghamdi [3] and Boris Novikov [4]

[1] Department of Information Systems and Technology, Port Said University, Port Fouad, Port Said 42526, Egypt
[2] Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia; asalshahrani2@uj.edu.sa
[3] Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia; amalghamdi@uj.edu.sa
[4] Department of Informatics, National Research University Higher School of Economics, 3A Kantemirovskaya St., 194100 Saint Petersburg, Russia; borisnov@acm.org
* Correspondence: no3man_mohamed@himc.psu.edu.eg

Abstract: Sentiment analysis on social media and e-markets has become an emerging trend. Extracting aspect terms for structure-free text is the primary task incorporated in the aspect-based sentiment analysis. This significance relies on the dependency of other tasks on the results it provides, which directly influences the accuracy of the final results of the sentiment analysis. In this work, we propose an aspect term extraction model to identify the prominent aspects. The model is based on clustering the word vectors generated using the pre-trained word embedding model. Dimensionality reduction was employed to improve the quality of word clusters obtained using the K-Means++ clustering algorithm. The proposed model was tested on the real datasets collected from online retailers' websites and the SemEval-14 dataset. Results show that our model outperforms the baseline models.

Keywords: aspect-based sentiment analysis; natural language processing; aspect-term extraction; word embedding; feature extraction

Citation: Ali, N.M.; Alshahrani, A.; Alghamdi, A.M.; Novikov, B. Extracting Prominent Aspects of Online Customer Reviews: A Data-Driven Approach to Big Data Analytics. *Electronics* **2022**, *11*, 2042. https://doi.org/10.3390/electronics11132042

Academic Editor: Qian Yin

Received: 20 May 2022
Accepted: 24 June 2022
Published: 29 June 2022

1. Introduction

Digital data generated every second worldwide is produced in a structured, semi-structured, and unstructured format. Unfortunately, traditional data analytics techniques cannot handle these volumes of data considering their complex structures. Therefore, big data analytics has emerged as a substantial research area and intensively researched to handle these problems. It is considered one of the most promising and rapidly developing areas in data sciences and the entire modern business. Sentiment analysis on social media and e-markets has become an emerging trend. To date, natural text processing is still challenging, with no standard procedures valid for all analysis tasks. Subsequently, developing a domain-independent aspect term extraction model is mandatory. In this work, we propose a novel method to automatically extract prominent review aspects of a product from customers' reviews.

Aspect-based sentiment analysis (ABSA) is one of the hotspots of natural language processing (NLP), which aims to mine opinions and sentiments towards specific products' aspects or services' features based on investigations of people's views obtained using their writing (e.g., reviews, blogs, comments, tweets). The process of performing an ABSA involves several subtasks that begin with the automatic identification of aspect-terms (*also called attributes or features*) of a particular target entity (e.g., Laptops), followed by the extraction of sentiment towards aspect terms. Next, the classification of extracted aspect terms into the appropriate categories (e.g., storage, price), and finally, compute the major polarity for each aspect category(*positive, negative, neutral, or conflict*) [1].

Aspect Term Extraction (ATE) from free text represents a challenging task. It is considered the primary task incorporated in aspect-based sentiment analysis. Extracting

aspect terms from the text starts with taking the free text as input and returns a set of aspects for each pre-identified entity [2,3]. The importance of this task is due to the dependence of other tasks on the results it provides, which directly influences the accuracy of the final results of the sentiment analysis. Online reviews could contain two kinds of product aspects: *explicit aspects*, which our work is interested in, and *implicit aspects* [4]. Explicit aspects refer to product features that users comment on directly within the review using exact words; for example, in the review: "Good product for the price works better with Google chrome", the aspect price has been stated explicitly. On the other hand, implicit aspects do not appear within the review but are commented on implicitly [5]; for example, in the review: "Great little computer that does what is written on the box", the user is talking about the size aspect without using an explicit word to state this aspect. Most research on developing ATE techniques strives to extract aspect terms from sentences [2,3,6–12], while in the era of big data, this will be time-consuming to inspect every single sentence individually; instead, a batch processing technique is required to extract prominent aspect terms instead [13,14].

1.1. Research Objectives

In this work, we strive to accomplish two primary objectives. First, we aim to collect review data from various online websites to build a diversified dataset suitable for validating the aspect term extraction model. Second, to develop a more reliable domain-specific aspect term extraction technique for big data, taking into account the main characteristics of big data affecting the performance of traditional methods. In other words, we aim to develop a scalable framework for extracting product aspects with stable performance.

1.2. Contributions

To fulfill these objectives, we build four datasets that contain the customers' feedback about specific products. The created datasets are collected from real and active websites.

We have investigated the potential of many natural language processing techniques for handling feature extraction problems from natural text. We have developed a domain-independent aspect term extraction model to extract the prominent aspect terms from customers' reviews text. The proposed model can extract singular aspect terms and aspect phrases as well. Word embedding technique that uses a neural network model were employed to convert prepared data to vectors. The SOMs have been employed to reduce the dimensionality of the generated vectors. On the other hand, the K-means++ clustering algorithm has been applied to extract the most prominent aspects. We have evaluated the performance of the proposed model; the obtained results demonstrate its validity and reliability.

The rest of this paper is organized as follows: In Section 2, we briefly present a discussion of the related studies. In Section 3, the basic components of our model have been described in detail. Section 4 introduces the proposed model with a description of the developed framework. Section 5 provides details of the conducted experiments, including a description of the datasets used in this study. The implementation details and an exploration of the obtained results are provided; the testing mechanism, evaluation metrics, and criteria are described in this context. In Section 6, a detailed discussion was presented to highlight our work's theoretical and practical implications. Finally, the conclusion that summarizes the results and future work.

2. Background

The key task in aspect-based sentiment analysis is extracting explicit aspects from text, more specifically online reviews–this relies on the dependency of other ABSA tasks on the results of this task [14–17]. Also, the international semantic evaluation competition (*SemEval-2014*) has attracted great attention to ABSA in recent years. The mentioned competition has involved the four primary ABSA subtasks. The first subtask was the aspect term extraction, mainly concerned with identifying the aspect terms in the given domain.

Recently, many approaches were introduced to extracting product features from online customers' reviews; for the sake of convenience, several classifications were proposed

to summarize these approaches. Authors of [18] tend to classify approaches into three categories: *machine learning-based, dependency relation-based, and lexicon-based approaches*. On the other hand, in [2], they tend to classify approaches based on the learning techniques into supervised, *semi-supervised*, and *unsupervised*, as depicted in Figure 1. Additionally, in [19], authors have classified ATE approaches into *rule-based methods, neural network-based methods, and topic modeling-based methods*. The rest of this section explores some of the most recent approaches proposed to ATE.

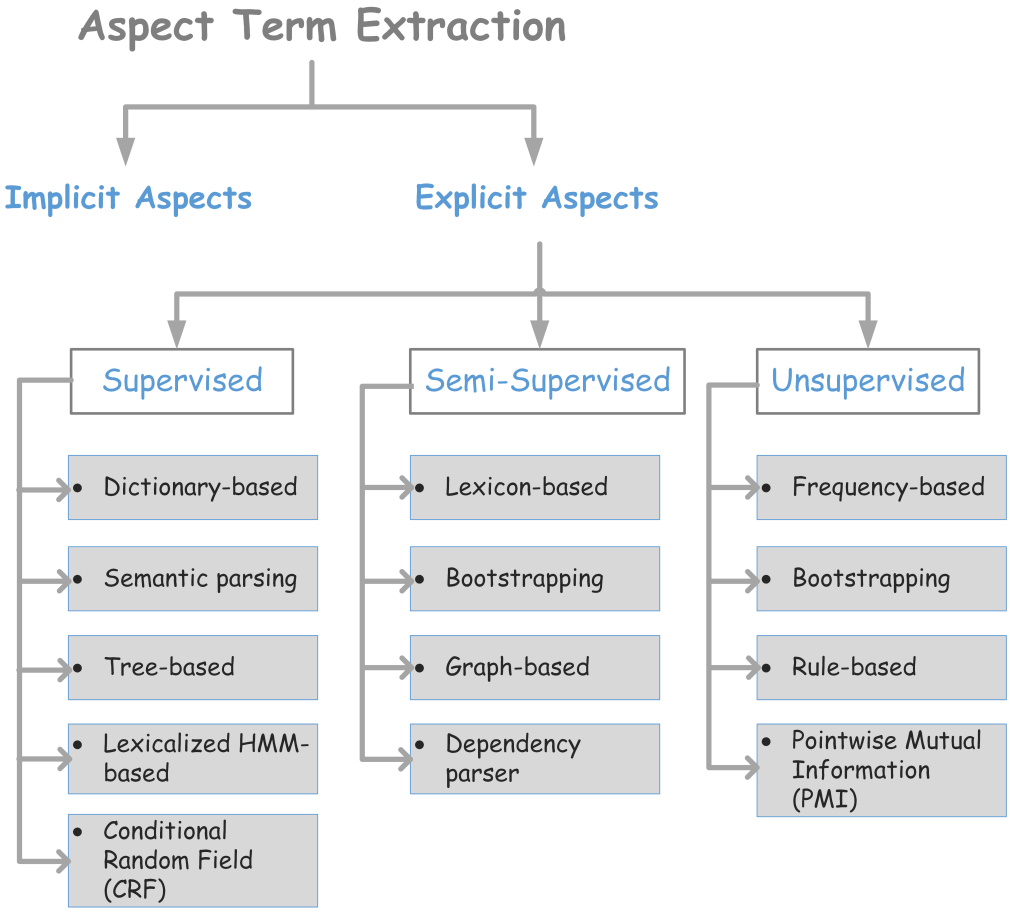

Figure 1. Classification of aspect term extraction techniques.

Shi Li et al. [20] introduced a frequency-based model for feature extraction with the help of similarity measures using the web. The proposed model represents an extension of the RCut text classification algorithm. It starts with selecting the candidate aspects, then measuring the semantic similarity with target entities. The RCut algorithm is used in the aspect selection process to learn the threshold. Experiment results on the Chinese language prove the model's ability to handle diverse domains and data sizes. Zhijun Yan et al. [18] have introduced the integration of the PageRank algorithm, implicit feature inference, and synonym expansion to automatically identify product aspects, which is called EXPRS (An EXtended PageRank algorithm enhanced by a Synonym lexicon). The data preparation process includes sentence-level segmentation, and POS tagging. The ATE tasks include extracting dependency relations and ranking words by the proposed NodeRank method to identify candidate aspects. Finally, it generates the final aspect set after getting word synonyms for candidate aspects and inferring implicit features. Xin Wang et al. [21] introduced an extension of dependency-based word embeddings for feature identification. In this work, the authors present an improved method for word representation based on the Recurrent Neural Networks (RNNs). The proposed model is language-independent, which means its ability to handle several languages. Shufeng

Xiong and Donghong Ji [22] proposed a semantically relevance-based model for extracting aspect-phrase. The authors have introduced a pipeline model of word embedding and clustering. Word embedding was applied to generate a weighted context representation of data. On the other hand, flexible-constrained clustering was introduced that built on the k-means clustering algorithm to obtain aspect-phrases cluster. Wei Xue et al. [23] introduced neural networks based model for aspect terms identification and classifying them into the proper category, respectively, called MTNA. The proposed multi-task learning model is built on the recurrent neural networks and convolutional neural networks (CNNs) as well, as they have combined the BiLSTM for feature extraction and CNNs for aspect category classification in a multi-task framework. Xin Li et al. [24] introduced a model for capturing aspect detection history and opinion summary based on LSTM. It consists of two primary parts. The first is Truncated History-Attention (THA) which is responsible for identifying aspects' history. The second is the Selective Transformation Network (STN) which works on capturing the opinion summary. The LSTMs produce the initial word representation that inputs the result to the two parts. Authors have concluded that the joint extraction sacrifices the accuracy of aspect prediction. Chuhan Wu et al. [25] introduced a hybrid unsupervised algorithm for aspect term extraction and opinion target extraction. The proposed model presents a combination of rule-based and machine learning techniques. The linguistic rules at the chunk level were used to extract opinion targets and features. Next, domain correlation-based filtering was applied to exclude irrelevant items. Results of these steps are finally used to train a deep-gated recurrent unit (GRU) network. Yanmin Xiang et al. [26] proposed a Multi-Feature Embedding (MFE) clustering method for extracting aspect terms in the ABSA, which is based on the Conditional Random Field (CRF). Text representation was done using MFE to extract extensive semantic information from data. Next, the K-means++ clustering algorithm was employed to generate word clusters to support the position features of the CRF. Finally, MFE clusters with word embedding were used to train the model. Zhiyi Luo et al. [19] introduced a data-driven-based approach for extracting prominent aspects from textual reviews called ExtRa (Extraction of Prominent Review Aspects), which can extract aspect words and phrases. The proposed method employs knowledge sources like WordNet and Probase to build aspect graphs to limit the aspect space. The aspect spaces are then divided into clusters to extract the prominent K aspect. Md Shad Akhtar et al. [27] proposed a pipeline framework for extracting and classifying aspect terms. The proposed multi-task learning framework begins with applying a Bi-directional Long Short Term Memory (Bi-LSTM) followed by a self-attention mechanism on sentences to extract aspect terms. Next, it employs the convolutional neural networks to predict the sentiments of the extracted product aspects. Experiments were conducted on English and Hindi and proved the proposed model's validity in handling both languages.

3. The Basic Procedures

Generally, the proposed model ensembles several sub-tasks, including Word embedding, multi-dimensional reduction, clustering, and feature selection. As follows, we will discuss the primary procedures of our model with different variants of specific tasks for performance tuning.

3.1. Word Embedding

In the last few years, researchers have attempted to improve the performance of text representation techniques, which positively affect several NLP tasks. Many Word Embedding models were introduced based on statistical models, machine learning, and deep learning (e.g., Word2Vec, GloVe, BERT) [9,21,22,28,29]. Such models produce word vectors of real numbers representing the context of the given words, which can be used as input features for further processing tasks. There are several levels of text representation using embedding techniques in addition to word-level, including char-based embeddings (e.g., Embeddings from Language Models "ELMo" [30], and Flair [31]), byte-level embeddings [32], and N-gram embeddings [33]. Concerning the word level, researchers have introduced several models that started from the old-fashioned embedding techniques

"Bag-of-Words" that built on simple machine learning algorithms like Term Frequency-Inverse Document Frequency (TF-IDF). Subsequently, the next-generation models were based on deep learning techniques like Word2Vec. Recently, cutting-edge word embedding models have appeared, introducing the language models used with transfer learning from attention-based transformers like BERT. The capabilities of word embedding of capturing the word's meaning in a specific context considering similarities to other words (e.g., semantically and syntactically and other relations) have made it one of the most significant technologies contributing to NLP. We examined several well-established word embedding techniques to evaluate how the performance of different models changes over various word embeddings.

- **Word2Vec:** Word2Vec is one of the most popular techniques of text embedding introduced by Mikolov et al. in 2013 [34] and accompanied by a C package; it is considered a protoplast model of any neural network word embedding. A pre-trained version exists, which is trained on Google News [35]. Other popular models in the same category include Facebook's "FastText" model and Stanford's "GloVe". Returning to the Word2Vec, the model consists of two-layer neural networks, and it takes a text corpus as input and generates a set of vectors representing a set of words respectively as output. A well-trained model groups similar word vectors nearby in one space. Training the Word2Vec model has two main alternatives: Hierarchical Softmax or Negative Sampling. The first approach is Continuous Bag of Words (CBOW), which uses the context to predict a target word. On the other hand, the second approach is the Skip-gram; unlike CBOW, it uses the word to predict the target context. Commonly, practices proved that the skip-gram method could perform better than the CBOW method.
- **BERT**: Dynamic Word Embeddings have appeared and developed rapidly, known as Language Models or Contextualized Word Embeddings. These models defeat the most significant limitation of the classic Word Embedding approaches: representing a single word with different meanings by one vector only and polysemy disambiguation. BERT (Bidirectional Encoder Representations from Transformers) represents one of the most well-known language models built based on Transformer introduced by Google's team, Jacob Devlin et al., in 2018 [36]. The Transformer is an attention mechanism that identifies contextual relations in the text between words or sub-words; it incorporates two separate mechanisms, encoder, and decoder. The first reads the input, and the second produces a prediction for the task. The BERT model randomly substitutes a subset of words in a sentence with a mask token, and next, it predicts the masked word based on the surrounding unmasked words through a transformer-based architecture. Unlike classical directional models, which read the input text sequentially (right-to-left or left-to-right), the prediction is generated considering word neighbors from both sides, the left and right, due to the transformer encoder mechanism that reads the complete sequence of words at once. BERT generates a word vector that is a function of the entire sentence. Accordingly, based on the context of a word, it can have different vectors. The model consists of four dimensions, [# layers, # batches, # tokens, # features]. The model is a deep neural network with 12 layers in the "Base" or 24 in the "Large" version. The batch number refers to the number of input sentences to the model, and the token numbers refer to the number of words in input sentences. Finally, the feature number of the hidden units refers to the number of dimensions representing each token, 768 in the "Base" or 1024 in the "Large" version.

3.2. Multi-Dimensional Reduction

Word embeddings are multidimensional; employing a word embedding technique to represent a corpus of text will generate a set of vectors, each representing only one word. The number of values of one vector refers to the number of dimensions that represent the word. For example, in Word2Vec models, the size parameters refer to the number of dimensions used to represent each word, directly proportional to the quality of the resulting vectors. Practically, the preferred values for the dimensionality size of word vectors range from 300 to 400, which depends on the input data size. On the other hand, in BERT, words are represented by 768 dimensions in the Base model or 1024 in the Large version. Increasing the dimensionality of data imposes difficulties in data interpretation and visualization. Several approaches are introduced for nonlinear dimensionality reduction, like PCA, SOM, and t-SNE. Such models provide a mapping from the high-dimensional space to the low-dimensional embedding.

- **SOM**: The Self-Organizing Maps (SOM) have been introduced by Teuvo Kohonen in 1982 [37–40], also known as Kohonen's Self Organizing Feature Maps, or Kohonen network. SOMs are an unsupervised machine learning technique that maps high-dimensional vectors onto a low-dimensional (typically two-dimensional) space; they learn to classify data without supervision. It is a type of ANN that can handle datasets with few references. Commonly, SOMs are used for data clustering, and visualization purposes [41]. SOM could reduce the dimensionality of vectors since it provides a way of representing high-dimensional data in low-dimensional spaces "2-D". One of the most significant aspects of SOMs distinguishing them from other clustering techniques is preserving the original data's topology because the distances in the 2-D space reflect those in the high-dimensional space. Kohonen's networks store information to maintain topological relationships within the input datasets. The internal learning mechanism continuously modifies network weights until input and target vectors become consistent. On the other hand, other clustering techniques like K-means generate a representation that is hard to visualize because it is not in a convenient 2-D format.
- **t-SNE**: The t-Distributed Stochastic Neighbor Embedding is an effective probabilistic technique for dimensionality reduction that can be used for visualizing multi-dimensional datasets [42–45]. Given complex datasets with numerous dimensions, t-SNE projects this data and the underlying relationships between vectors in a lower-dimensional space and represents it in 2D or 3D while preserving the original dataset's structure. In addition to data visualization, the power of t-SNE to cluster data in an unsupervised gives it the ability to aid machine learning algorithms in prediction and classification. The working mechanism of the t-SNE starts with measuring the similarity between every possible pair of data points towards generating a similarity distribution of the high-dimensions input data. Next, it creates a pairwise similarity distribution of the corresponding low-dimensional points in the embedding–finally, its works on minimizing the divergence between these two distributions. Several approaches use t-SNE in large-scale datasets; one approach ensembles it with another technique like PCA; or implements t-SNE via approximation technique like the Barnes-Hut approximations.

3.3. Clustering Feature

Clustering algorithms are used to separate a set of objects and group them according to the similarity of features measured through their attributes and proximity in the vector space. Various fields could employ clustering techniques, including statistical analysis, computer vision, genomics, machine learning, and data science. Generally, clustering algorithms can be classified into six main categories:

1. Hierarchical clustering "Connectivity-based" (e.g., BIRCH, ROCK, DIANA)
2. Partitioning clustering "Centroids-based" (e.g., K-means, k-modes)
3. Density-based clustering (e.g., DBSCAN, DENCAST)
4. Distribution-based clustering (e.g., Gaussian Mixed Models, DBCLASD)
5. Fuzzy Clustering (e.g., Fuzzy C means, Rough k-means)
6. Constraint-Based (e.g., Decision Trees, Random Forest, Gradient Boosting)

Despite the comprehensive efforts by researchers to improve the clustering techniques, handling high-dimensional feature vectors is still challenging in any clustering task. Therefore, this work ensembles the stepped clustering of feature vectors; the first phase handles the high-dimensional problem and generates a 2-D representation of vector space. On the other hand, the second phase produces high-quality clusters of features. For the second phase, we have selected the K-means++ clustering algorithm.

- **K-Means++:** David Arthur and Sergei Vassilvitskii introduced the k-means++ clustering algorithm in 2007 to overcome the drawback posed by the k-means algorithm [46]. They have presented Monte Carlo simulation results proving that k-means++ is faster and performs better than the standard version. It is an unsupervised clustering algorithm that collects and separates data into k number of clusters. The mechanism work of k-means++ is choosing the initial center points and starting the standard k-means; this is what k-means++ does to avoid the poor clustering found by classical k-means algorithms. K-Means++ centers are distributed across data and are less costly than random initialization of centroids. Initialization in k-means++ is performed by randomly allocating one cluster center; given the first center, it searches for other centers. Therefore, the algorithm is guaranteed to find the optimal result instead of getting the local optimum achieved by the old one.

3.4. Feature Selection

Concerning the set of vectors' clusters representing words and phrases produced from the clustering phase, the purpose is to filter out irrelevant tokens and extract the list of prominent aspects. Algorithm 1 presents the proposed features selection procedure. It starts with counting the frequency distribution for nouns, adjectives, and verbs only using Part-of-Speech tags since these word classes synthesize most aspect terms. The next step is computing the distance space for each cluster and extracting the distance from the cluster's centroid for each item. Then it ascendingly sorts items in each cluster and selects the top central items, "Candidate Aspects List". Converting vectors into textual equivalent tokens is the next step; selected items are words and phrases as well. Next is filtering the list by accepting only the mentioned word classes for both single words and parts in phrases. Finally, extracting frequencies for single words and phrases from the frequency distribution mentioned early and excluding items that did not pass a certain threshold–consequently, the filtered items constitute the list of prominent aspects.

Algorithm 1 Feature Selection

Input: (LIST: *Vectors_Clusters*)
(LIST: *Tagged_Sentences*)
Output: (LIST: *Prominent_Aspects*)
Initialization :
Extract ← *Tagged_Sentences* ⇒ "*Nouns*, *Verbs*, *Adjectives* : *NVA* "
Count ← *NVA* ⇒ "*Freq_Dist*"
Extract ← *Vectors_Clusters* ⇒ (*LIST* : *Cluster_Labels*)
Calculate ← *Vectors_Clusters* ⇒ "*Distance_Space*"
Create ← (*LIST* : {*Label* : (*LIST* : *distance*)}) ⇒ "*Cluster_Distance*"
for each Label ∈ *Cluster_Labels* **do**
    Extract ← *Distance_Space* ⇒ "*distance*"
    Append ← *distance* ⇒ "*Cluster_Distance*[*Label*]"
end for
threshold = *min*
Create ← (*LIST* : *Prominent_Aspects*)
for each Label ∈ *Cluster_Distance* **do**
    Sort ← *items* ∈ "*Cluster_Distance*[*Label*]" ⇒ "*sorted_items*"
    Select ← *sorted_items* ⇒ "*top_items*"
    for each item ∈ *top_items* **do**
        Convert ← *Vector: item* ⇒ "*word_token*"
        token_freq = *Zero*
        if (`Length(`*word_token*`) > One`) **then**
            for each part ∈ *word_token* **do**
                Get ← *Freq_Dist* ⇒ "*Freq_Dist*[*part*]"
                token_freq = *token_freq* + *Freq_Dist*[*part*]
            end for
        else
            Get ← *Freq_Dist* ⇒ "*Freq_Dist*[*word_token*]"
            token_freq = *Freq_Dist*[*word_token*]
        end if
        if (*token_freq* >= *threshold*) **then**
            Append ← *word_token* ⇒ "*Prominent_Aspects*"
        end if
    end for
end for
return *Prominent_Aspects*

4. The Proposed Model for ATE

This work's primary goal is to extract the most prominent aspect terms from the customers' review text. The proposed model aims to generate a numerical representation of the text "Word vectors" and then apply a dimensionality reduction technique to improve the quality of word clusters and finally filter extracted aspects to obtain the leading aspects of the subject of reviews. As follows, we will discuss the workflow of our model and how it operates.

The Framework Description

Generally, the proposed model incorporates several subtasks starting from collecting the customers' reviews, followed by a series of data processing tasks, ending with providing a list of prominent aspects of the target entity, as shown in Figure 2. Data collection and preprocessing is the first step toward achieving the desired task [47,48]. Clustering features accept only numerical data; therefore, getting a numerical representation of textual data is imperative. Word embedding is the most popular technique used to achieve this task. So, the next step is to extract the list of words because word embedding requires the input data to be formatted as a list of words. Since our original dataset consists of the customers' reviews, each review could be divided into sentences, "list of sentences", and

each sentence would be a list of tokens. Therefore, we need to convert data into a "list of lists" format to be accepted input to the word embedding techniques. In this work, we have investigated two different techniques for word embedding that are widely used, namely "Word2Vec and BERT", and analyzed their effect on the performance of our model. The extraction of aspect phrases and aspect terms has also been considered; therefore, we have employed a Multi-Feature Embedding (MFE) to capture aspect phrases. Results from several levels of word embedding were then combined. Since generated vectors are of high dimension, which hinders any trials for clustering these vectors by negatively affecting the quality of resulting clusters, an intermediate clustering process is necessary to reduce the dimensionality of the input vectors while preserving the original data topology. Here we have investigated two different techniques to accomplish this task, namely "SOM and t-SNE". Clustering of generated 2-D vectors is the next step in our model. The selection of the K-Means++ clustering algorithm ensembled in the proposed model relies on its efficiency and appropriateness to our datasets and the analysis purpose. Generated clusters of vectors are the input to the proposed feature selection algorithms discussed previously, which extract the list of prominent aspects.

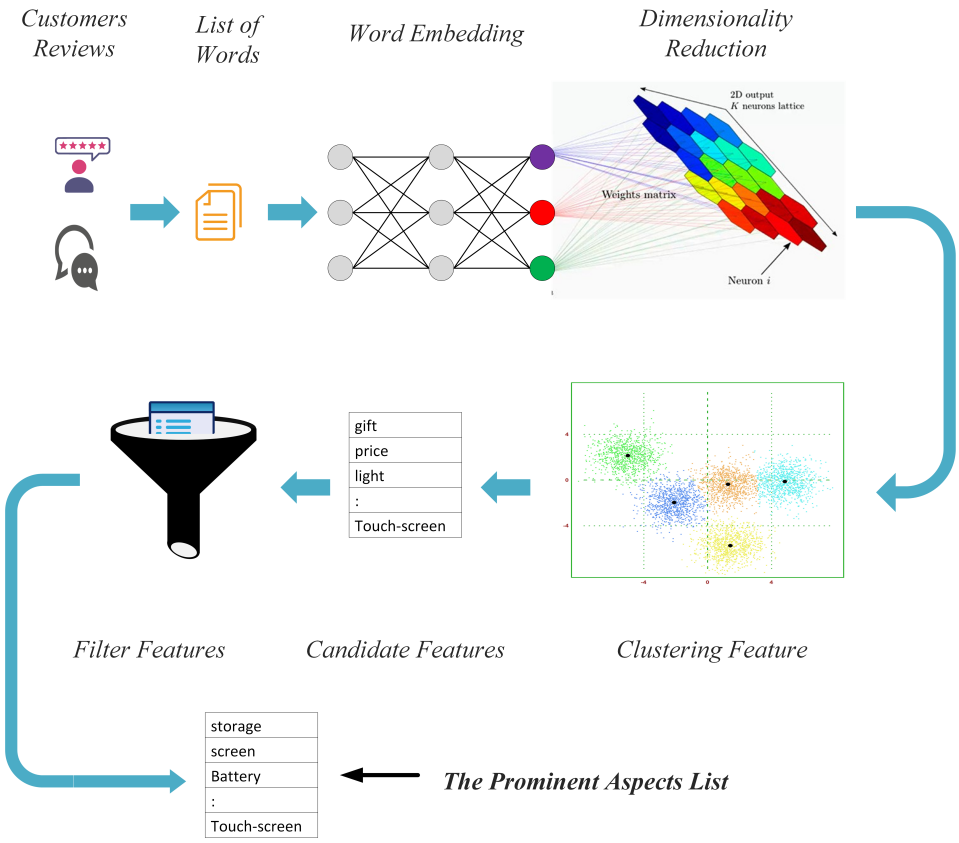

Figure 2. The experimental framework of the proposed aspect term extraction model.

5. Experiments and Results

To examine the efficiency of our model, we have tested and evaluated it against several available and publicly accessed datasets. The following is a description of the datasets used in the experiment–next is a presentation of the experimental setup of the ensembled procedures. Following is a review of the evaluation metrics and benchmarks used in our experiment–finally, a discussion of the achieved results.

5.1. Datasets

The experiments were conducted using two groups of datasets. The first group consists of online customers' reviews that we have collected from online markets using

a developed web scraper. On the other hand, the second group consists of benchmark datasets that researchers widely use to compare with baseline models. The datasets used in the experiments are as follows:

5.1.1. Collected Datasets

- **Amazon:** The dataset contains 28,125 online customer reviews regarding 45 different laptop products.
- **BestBuy:** The dataset contains 47,905 online customer reviews regarding 33 different laptop products.
- **Dell:** The dataset contains 44,175 online customer reviews regarding 19 different laptop products.
- **Lenovo:** The dataset contains 85,711 online customer reviews regarding 21 different laptop products.

5.1.2. Benchmark Datasets

- **SemEval-14:** The organizers of the International Workshop on Semantic Evaluation (SemEval-2014) held in Dublin, Ireland, have announced a competition of semantic evaluation tasks for two domain-specific datasets; we are interested in the fourth task that is concerned with the aspect-based sentiment analysis and, more precisely, with the first subtask that concerned to aspect term extraction. The datasets include customers' reviews from two different domains, restaurants and laptops. Table 1 shows the main characteristics of the mentioned datasets. Figure 3 shows an illustrative example that presents the format of one annotated sentence of the laptop's trial dataset.

Table 1. The main characteristics of the SemEval-14 datasets.

Dataset		Laptops			Restaurants		
		Trial	**Test**	**Total**	**Trial**	**Test**	**Total**
Sentence Count		3048	800	3848	3044	800	3844
Aspect Term	With Repetition	2373	654	3027	3699	1134	4833
	Unique Items	1048	419	1315	1295	555	1657

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sentences>
<sentence id="1630">
        <text>I took it back for an Asus and same thing- blue screen which
        required me to remove the battery to reset.</text>
        <aspectTerms>
            <aspectTerm term="battery" polarity="neutral" from="87" to="94"/>
        </aspectTerms>
    </sentence>
</sentences>
```

Figure 3. A sample of the laptop's trial dataset shows an annotated sentence using the XML tags.

5.2. Experiment Setup

We experimented with various approaches for involved techniques in our ensemble. As follows, we will explore experiment settings regarding alternatives available for each task.

5.2.1. Word Embedding

To extract aspect terms and aspect phrases as well, we have extracted word sequences that occur together using n-gram models. Frequent word sequences only were considered. Subsequently, these phrases substituted singular counterpart words in sentences before inputting them to the word embedding model to handle them as a single term.

Word2Vec: Several runs were performed to choose the most fitting hyper-parameters from these runs on our datasets as follows: the dimensionality of word vectors equal

to 300, the maximum distance between the target word and its neighboring word equal to 30 tokens, the minimum word frequency equal to 100, training the model using the Skip-Gram algorithm through the negative sampling technique, the number of worker threads used to train the model equals 8.

BERT: We used 2 different pre-trained word embeddings models. The first is the Base uncased model, consisting of a deep neural network with 12 layers; the dimensions representing each token are 768 features. On the other hand, the second model is the Large uncased version, consisting of a deep neural network with 24 layers; the dimensions representing each token are 1024 features. Tokenizing the sentence was done using the "BERT Tokenizer" model that adds the special tokens '[CLS]' indicating the start of the text and '[SEP]' to separate adjacent sentences and for padding & truncating all sentences. The max_length is set to 64; padding equals the 'max_length'; returning attention mask property is set to "True" to construct the attention masks; and return tensors is set to 'pt', to return PyTorch tensors. The model consists of four dimensions, [# layers, # batches, # tokens, # features]. For getting word embeddings, we have removed the batches dimension and rearranged the remaining dimensions to be as follows: [#tokens,# layers,# features]; for each token, we have concatenated the vectors from the last four layers.

5.2.2. Dimensionality Reduction

SOM: Many trials were executed towards selecting the most fitting hyper-parameters from these trials on our datasets as follows: the grid size of a SOM is set to 80 for both numbers of rows and columns as well; the selected working mode is the unsupervised mode; initialization of the unsupervised mode is set to "random"; the number of iterations is set equal to 50,000; the training mode is set to "Batch Mode"; the selected neighborhood mode is the "linear" mode; the chosen learning mode of the unsupervised SOM is "min"; the distance metric for the comparison on feature level is the "Euclidean Distance"; the starting and ending learning rate is set to 0.5 and 0.05 respectively; the neighborhood distance weight mode is set to "pseudo gaussian"; the number of jobs to run in parallel is 3; the random number generator of the NumPy "random" method is used for the random state property; the verbosity controllers is set to zero.

t-SNE: We have used the following parameters' settings which are fitted appropriately to our dataset: the number of dimensions of the embedded space is set to 2-D; the number of nearest neighbors "perplexity" is set to 40; the learning rate equals 300; the early exaggeration property is set to 24; the maximum number of iterations for the optimization is set to 2500; aborting the optimization will occur after 500 unprogressive iterations; the minimum gradient norm threshold for halting the optimization is set to be under 1e-7; the distance metrics between instances in a feature array are chosen to be "Euclidian"; the initialization of embedding is set to "PCA"; the random number generator is set to 25; setting the square distances property to "legacy" to square distance values.

5.2.3. Clustering

K-Measns++: Depending on the results of numerous trials, we have used the following parameters settings: the preferred number of clusters ranges from 50 to 250 depending on the size of the dataset; the initialization method is set to "k-means++"; the maximum number of iterations for a single run is equal to 400; the number of runs with different centroid seeds is set to 15; the relative tolerance property is set to 0.0001; the verbosity mode is set to zero; the random number generation for centroid initialization is set to Zero for deterministic randomness; the "copy_x" property is set to "True" to avoid modifying original data; the K-means algorithm to use is selected to be "Elkan".

5.3. Testing and Evaluation Metrics

5.3.1. Evaluation Metrics

The results of the conducted experiments were evaluated using the following metrics: Precision (P), Recall (R), and F-measure (F), to assess the effectiveness of the proposed model. Regarding a list of extracted aspects, precision is the proportion of correct extracted

features "cor(asp)" to the total extracted aspects "pre(asp)". The recall is the proportion of correct extracted aspects to all correct features "true(asp)". On the other hand, F-score indicates the overall performance of precision and recall; it is a harmonic average of precision and recall. The formulas for Precision, Recall, and F-Score are as follows:

$$Precision(P) = \frac{cor(asp)}{pre(asp)} \tag{1}$$

$$Recall(R) = \frac{cor(asp)}{true(asp)} \tag{2}$$

$$F - Score = \frac{2 * P * R}{P + R} \tag{3}$$

5.3.2. Baseline Methods

To validate the performance of our proposed model, a comparison of our model against multiple baselines was made for this task, including the following models:

- **DTBCSNN+F** [49]: A convolutional stacked neural network model built on the dependency parse tree towards capturing the syntactic features.
- **MFE-CRF** [26]: Based on the Conditional Random Field, Multi-Feature Embedding clustering was proposed to capture implicit contextual information.
- **Wo-BiLSTM-CRF** [28]: Using pre-trained word embedding model "Glove.42B", Bi-directional long short-term memory with conditional random fields was proposed to enhance the feature representation and extraction.
- **WoCh-BiLSTM-CRF** [28]: Using the character-level word embedding along with word-level embedding through employing pre-trained word embeddings "Glove.42B", to enhance the performance of Bi-directional long short-term memory with the optional conditional random field.

5.4. Result Analysis and Discussion

In this section, the proposed models were tested to verify our work's viability for implementation; also compared with the baseline models to evaluate our models' efficiency and assure their quality. We have proposed several models' ensembles that use two different pre-trained word embeddings techniques, two different dimensionality reduction methods, and one clustering algorithm as follows:

1. W-S-K: Word2Vec → SOM → K-Means++
2. W-t-K: Word2Vec → t-SNE → K-Means++
3. B(B)-S-K: BERT (Base) → SOM → K-Means++
4. B(L)-S-K: BERT (Large) → SOM → K-Means++
5. B(B)-t-K: BERT (Base) → t-SNE → K-Means++
6. B(L)-t-K: BERT (Large) → t-SNE → K-Means++

5.4.1. Word2vec vs. Bert

Results appear in Figure 4 show that the overall performance of models employing BERT word embeddings acting over other those employing the Word2Vec word embeddings; this may rely on the well trained that BERT various models built on and also the representation of words that involves semantic and contextual information of words from two sides producing higher quality than those produced by Word2Vec.

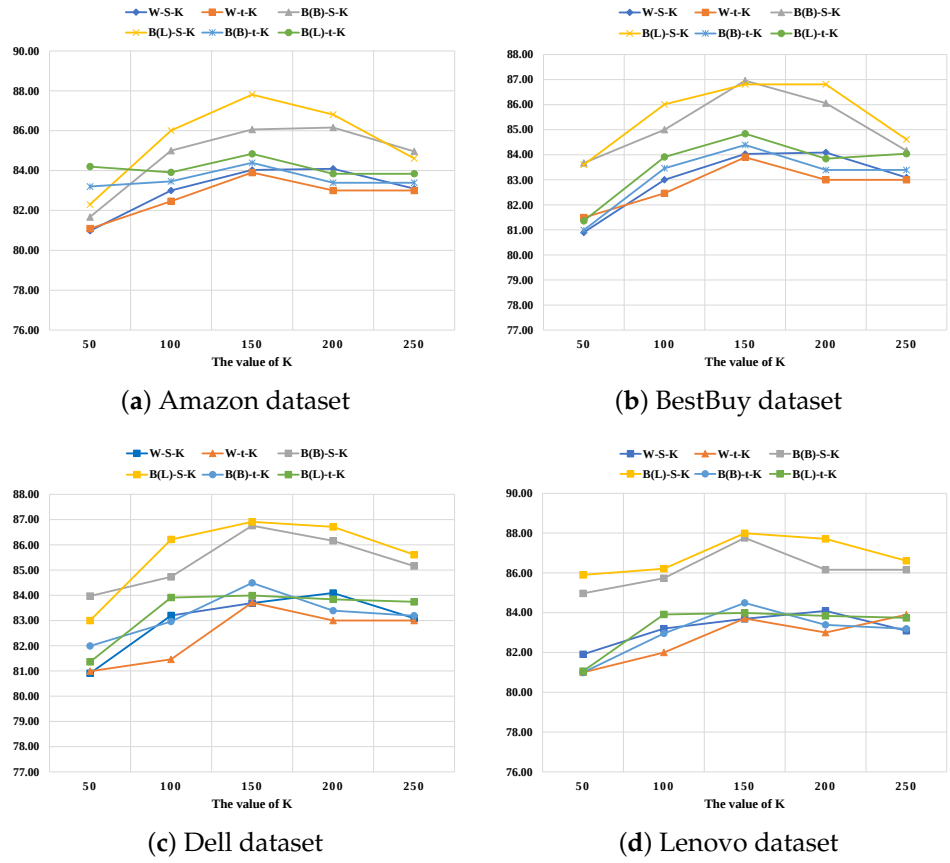

Figure 4. The effect of K value on the experimental results in terms of percentage of the precision.

5.4.2. SOM vs. t-SNE

The experimental results show that SOM and t-SNE act similarly with low volumes of data with a relative advantage to t-SNE, while when the volume of data is going on, SOM remarkably performed better with huge volumes of data.

5.4.3. The Evaluation of K Value

Numerous runs were performed using different sequential values of K; the results show that the best performance could be obtained with values in the range of 50 to 250. Figure 4 shows the results of different model combinations in terms of precision on each dataset separately; we have limited the results to the preferred values range exclusively for clarity. We can observe that the highest precision achieved was when k = 150.

5.4.4. Performance Analysis of the Proposed Models

The comparison between proposed models presented in Figure 5 shows that the third model "B(B)-S-K" and fourth model "B(L)-S-K", respectively, outperform other models with a slight superiority to the fourth model. This situation was proved previously by comparing the contributed procedures where BERT and SOM contributions provide significantly better performance than the Word2Vec and t-SNE, respectively.

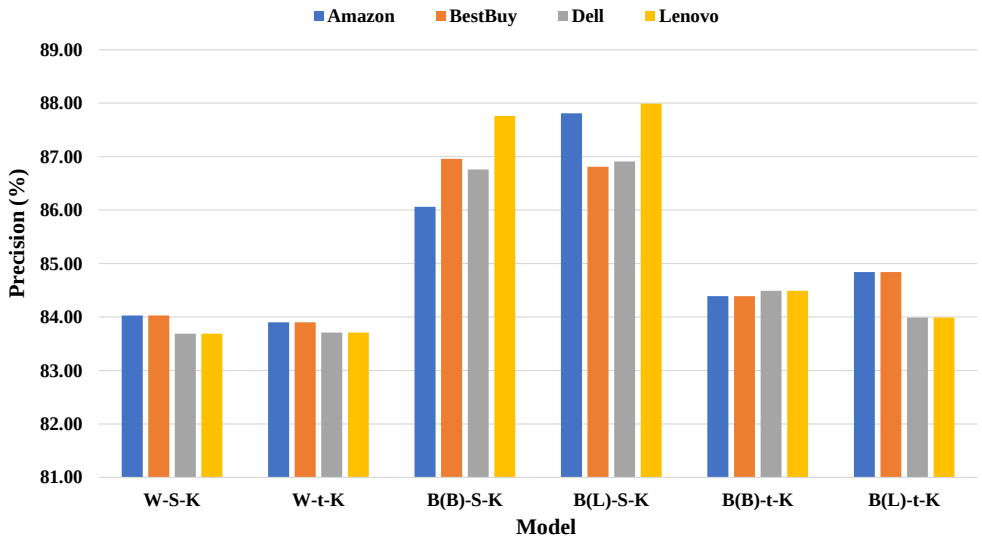

Figure 5. The comparison between the achieved results from applying proposed models variants on different datasets, where K = 150.

5.4.5. The Comparison with Baseline Methods

To compare our models against the baseline methods, it is mandatory to test them on benchmark datasets that researchers widely use. Therefore, we have tested the proposed models' combinations on the SemEval-2014 datasets. Table 2 shows the experimental results of applying these combinations on the restaurants' dataset using a K value equal to 200 and laptops datasets using the K value equal to 150. Results show that the same situation occurred against our collected datasets, where the third and fourth models provide better performance than other combinations. The main remarks that their performance is stable for both domain-specific datasets, which proves the efficiency of our models and their ability to handle various domains separately, as depicted in Figure 6. Furthermore, we can say that handling multi-domain datasets are still a challenge.

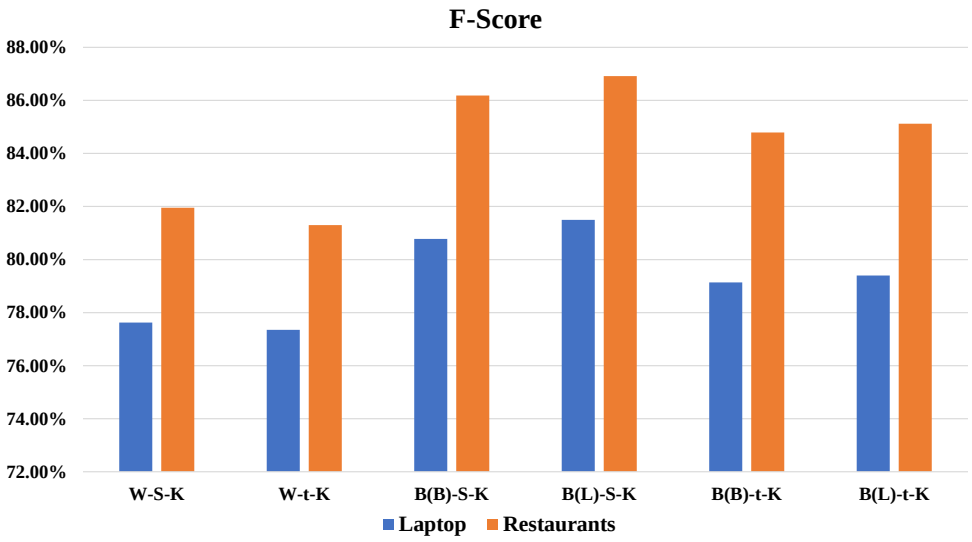

Figure 6. Comparison of various model combinations in terms of F-score results.

Table 2. The results of applying different aspect term extraction models on the SemEval-14 datasets, using the K value equal to 150 with the Laptops dataset and 200 with the restaurants' dataset.

	Laptops			Restaurants		
Model	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
W-S-K	83.98%	72.17%	77.63%	85.86%	78.39%	81.96%
W-t-K	84.37%	71.41%	77.35%	84.99%	77.91%	81.30%
B(B)-S-K	88.28%	74.45%	80.78%	88.54%	83.95%	86.18%
B(L)-S-K	88.72%	75.36%	81.50%	89.83%	84.19%	86.92%
B(B)-t-K	85.25%	73.84%	79.14%	86.55%	83.10%	84.79%
B(L)-t-K	85.45%	74.14%	79.40%	87.11%	83.22%	85.12%

Regarding the evaluation of our models, we have compared the results of F-Score achieved by the fourth model of proposed combinations "B(L)-S-K," which provides the best performance, against the state-of-the-art models as depicted in Table 3. Although the WoCh-BiLSTM-CRF model provides good performance with a relative advantage over the Wo-BiLSTM-CRF, it employs a combination of word and character embedding. However, the performances of these models that use traditional GloVe embedding vectors are generally inferior to our model that is built on BERT. This performance demonstrates the effectiveness of employing deep learning models and their superiority over linear CRF-based models, see Figure 7. We can observe that our model outperforms the baseline methods. It acts well with a relative advantage on the laptops' dataset, while it provides significant improvements over other models with the restaurants' dataset.

Table 3. Comparison of F-score results of baseline methods for SemEval-14.

Model	**Laptops**	**Restaurants**
B(L)-S-K	81.50%	86.92%
DTBCSNN+F	75.66%	83.97%
MFE-CRF	76.53%	84.33%
Wo-BiLSTM-CRF-Glove.42B	81.08%	84.97%
WoCh-BiLSTM-CRF-Glove.42B	79.21%	86.05%

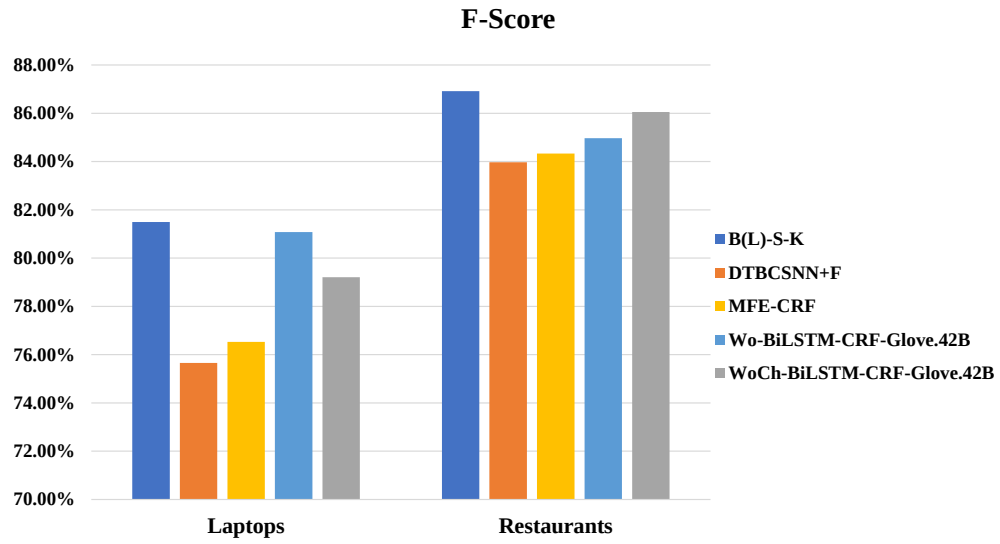

Figure 7. Comparisons of F-score between the baseline methods and our model for the SemEval-14 datasets.

6. Discussion

Undoubtedly, the web is the primary resource for data; most available online data are in text form. Various methods have been proposed in the data sciences literature to analyze the raw textual data to extract aspect terms from free-structure text. Several websites, including e-markets and vendors, widely employ such techniques to improve their capabilities of providing users with the most relevant information to their preferences. Regarding the business intelligence domain, the primary goal is represented in two main objects: predicting the consumer's needs and realizing satisfying interactions for customers and vendors.

Generally, companies seek to collect and analyze customers' feedback for several commercial purposes. The analysis of this information could help companies in many areas, including planning promotional campaigns, product redesign, advertising, customer satisfaction, loyalty, providing customized offers, etc. Although online feedback has been identified as a significant resource for analyzing the market, studies that endeavor to develop scalable techniques that can handle a vast amount of data automatically are rarely seen in the literature regarding the nature and characteristics of big data that imposes additional challenges.

Several approaches have been introduced to extract aspect terms; most of them are domain-dependent, which affects the performance of these approaches in terms of low precision of the system in different domains. On the other hand, aspect terms are not necessarily to be single words; extracting aspect phrases besides single terms is still a challenge. Also, filtering irrelevant terms is fine-tuned task where selecting the most significant aspects is mandatory. Building an incremental model able to handle the increasing amounts of data is also essential.

The study has double implications, theoretical and practical. Theoretically, this work presents good results in ATE tasks. The model is based on clustering the word vectors generated using the pre-trained word embedding model. Results show that the model outperforms the baseline models. Basically, this work contributes to the competition of the international semantic evaluation competition (SemEval-2014). The competition has involved the four primary ABSA subtasks. The first subtask was the aspect term extraction, mainly concerned with identifying the aspect terms in the given domain. Also, our study contributes to the literature on text classification, pattern recognition, text summarization, and information extraction by introducing an unconventional approach that can automatically extract the prominent aspect terms from a large volume of data.

We built diverse datasets to serve as an experimental environment for the proposed model. The created datasets were collected from real and active websites, which comprise customers' reviews of products and services. Then a preliminary data preprocessing has been performed to clean and filter data to be suitable to apply the proposed model. Also, we have used the benchmark datasets to compare our results with the baseline models to test and demonstrate our model validity against other models.

The proposed model extracts aspect terms from review text based on a data-driven approach. It extracts single words and can also extract phrases as prominent aspects. Various word embedding techniques were considered to get a numerical representation of each word. Given the high dimension of the generated word vectors, the Self Organizing Maps (SOM) technique was applied to reduce the dimensionality and preserve the original data characteristics. Next, the K-means++ clustering technique was applied to extract the most prominent aspect terms with the help of pre-defined filtering criteria. We conducted experiments to evaluate the performance of the proposed model and examine its capabilities in extract aspect terms to validate our approach. The presented results and the comparison with baseline models demonstrate its validity and reliability. Furthermore, the experimental results demonstrate the effectiveness of K-Means++, which is guaranteed to find the optimal result instead of getting the local optimum achieved by the old one.

In addition to extracting the prominent aspect terms from a large volume of online review data, the results obtained from implementing the developed algorithm are of prac-

tical importance and can be used in different analyzing tasks of natural language text, specifically in social media and e-marketing. Also, it has good prospects for prospective development, including incorporating other systems like web personalization, user profiling, and recommender systems.

7. Conclusions

The primary task incorporated in the aspect-based sentiment analysis is the Aspect Term Extraction from free text. This significance relies on the dependency of other tasks on the results it provides, which directly influences the accuracy of the final results of the sentiment analysis. In this paper, we have proposed six different models' combinations for aspect term extraction built based on clustering the word vectors generated using the pre-trained BERT model with the help of dimensionality reduction techniques "SOM" to improve the quality of word clusters that could be obtained using the K-Means++ clustering algorithm. We have conducted extensive experiments to ensure our models' implementation validity and verify their effectiveness. The results show that all the proposed models act well, but one performs better with higher precision than the others. To assure its efficiency, we have tested our best model combination on the SemEval-14 datasets and compared it against the baseline methods. Results show that our model outperforms them in terms of F-Score results. This work has good prospects for future development; we plan to adjust our model to be applied to multi-domain datasets. Additionally, the application of our model to another language, "Arabic", is also considered in the future.

Author Contributions: Conceptualization, N.M.A.; methodology, N.M.A., A.A., A.M.A. and B.N.; software, N.M.A., A.A., A.M.A. and B.N.; validation, N.M.A., A.A., A.M.A. and B.N.; formal analysis, N.M.A., A.A. and A.M.A.; investigation, N.M.A., A.A., A.M.A. and B.N.; resources, N.M.A., A.A., A.M.A. and B.N.; data curation, N.M.A., A.A., A.M.A. and B.N.; writing—original draft preparation, N.M.A. and A.A.; writing—review and editing, N.M.A., A.M.A. and B.N.; visualization, N.M.A., A.A. and B.N.; supervision, B.N.; project administration, N.M.A., A.A., A.M.A. and B.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The researcher [Noaman M. Ali] is funded by a scholarship [EGY-6428/17] under the Joint Executive Program between the Arab Republic of Egypt and the Russian Federation.

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

References

1. Zhang, L.; Liu, B.; Lim, S.H.; O'Brien-Strain, E. Extracting and Ranking Product Features in Opinion Documents. In Proceedings of the 23rd International Conference on Computational Linguistics, Posters, Beijing, China, 23–27 August 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010. [CrossRef]
2. Rana, T.A.; Cheah, Y.N. Aspect Extraction in Sentiment Analysis: Comparative Analysis and Survey. *Artif. Intell. Rev.* **2016**, *46*, 459–483. [CrossRef]
3. Dragoni, M.; Federici, M.; Rexha, A. An Unsupervised Aspect Extraction Strategy for Monitoring Real-Time Reviews Stream. *Inf. Process. Manag.* **2019**, *56*, 1103–1118. [CrossRef]
4. Tubishat, M.; Idris, N.; Abushariah, M.A.M. Implicit Aspect Extraction in Sentiment Analysis: Review, Taxonomy, Oppportunities, and Open Challenges. *Inf. Process. Manag.* **2018**, *54*, 545–563. [CrossRef]
5. Rana, T.A.; Cheah, Y.N.; Rana, T. Multi-Level Knowledge-Based Approach for Implicit Aspect Identification. *Appl. Intell.* **2020**, *50*, 4616–4630. [CrossRef]
6. Ren, F.; Sohrab, M.G. Class-Indexing-Based Term Weighting for Automatic Text Classification. *Inf. Sci.* **2013**, *236*, 109–125. [CrossRef]
7. Akhtar, M.S.; Gupta, D.; Ekbal, A.; Bhattacharyya, P. Feature Selection and Ensemble Construction: A two-Step Method for Aspect Based Sentiment Analysis. *Knowl.-Based Syst.* **2017**, *125*, 116–135. [CrossRef]

8. Al-Smadi, M.; Al-Ayyoub, M.; Jararweh, Y.; Qawasmeh, O. Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels' Reviews Using Morphological, Syntactic and Semantic Features. *Inf. Process. Manag.* **2019**, *56*, 308–319. [CrossRef]
9. Song, M.; Park, H.; Shin, K.s. Attention-Based Long Short-Term Memory Network Using Sentiment Lexicon Embedding for Aspect-Level Sentiment Analysis in Korean. *Inf. Process. Manag.* **2019**, *56*, 637–653. [CrossRef]
10. Fu, Y.; Liao, J.; Li, Y.; Wang, S.; Li, D.; Li, X. Multiple Perspective Attention Based on Double BiLSTM for Aspect and Sentiment Pair Extract. *Neurocomputing* **2021**, *438*, 302–311. [CrossRef]
11. Zhao, H.; Liu, Z.; Yao, X.; Yang, Q. A Machine Learning-Based Sentiment Analysis of Online Product Reviews with A Novel Term Weighting and Feature Selection Approach. *Inf. Process. Manag.* **2021**, *58*, 102656. [CrossRef]
12. Wan, C.; Peng, Y.; Xiao, K.; Liu, X.; Jiang, T.; Liu, D. An Association-Constrained LDA Model for Joint Extraction of Product Aspects and Opinions. *Inf. Sci.* **2020**, *519*, 243–259. [CrossRef]
13. Khan, M.T.; Durrani, M.; Khalid, S.; Aziz, F. Lifelong Aspect Extraction from Big Data: Knowledge Engineering. *Complex Adapt. Syst. Model.* **2016**, *4*, 15. [CrossRef]
14. Ali, N.M.; Novikov, B.A. Big Data: Analytical Solutions, Research Challenges and Trends. *Proc. Inst. Syst. Program. Russ. Acad. Sci.* **2020**, *32*, 181–204. [CrossRef]
15. Ali, N.M. Aspect-Oriented Analytics of Big Data. In Proceedings of the 14th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2020), Tallinn, Estonia, 16–19 June 2020; Volume 2620, pp. 41–48.
16. Peng, Y.; Xiao, T.; Yuan, H. Cooperative Gating Network Based on A Single BERT Encoder for Aspect Term Sentiment Analysis. *Appl. Intell.* **2021**, *5*, 5867–5879. [CrossRef]
17. He, J.; Li, L.; Wang, Y.; Wu, X. Targeted Aspects Oriented Topic Modeling for Short Texts. *Appl. Intell.* **2020**, *50*, 2384–2399. [CrossRef]
18. Yan, Z.; Xing, M.; Zhang, D.; Ma, B. EXPRS: An Extended Pagerank Method for Product Feature Extraction from Online Consumer Reviews. *Inf. Manag.* **2015**, *52*, 850–858. [CrossRef]
19. Luo, Z.; Huang, S.; Zhu, K.Q. Knowledge Empowered Prominent Aspect Extraction from Product Reviews. *Inf. Process. Manag.* **2019**, *56*, 408–423. [CrossRef]
20. Li, S.; Zhou, L.; Li, Y. Improving Aspect Extraction by Augmenting A Frequency-Based Method With Web-Based Similarity Measures. *Inf. Process. Manag.* **2015**, *51*, 58–67. [CrossRef]
21. Wang, X.; Liu, Y.; Sun, C.; Liu, M.; Wang, X., Extended Dependency-Based Word Embeddings for Aspect Extraction. In Proceedings of the International Conference on Neural Information Processing ICONIP, Neural Information Processing , Kyoto, Japan, 16–21 October 2016. [CrossRef]
22. Xiong, S.; Ji, D. Exploiting Flexible-Constrained K-Means Clustering with Word Embedding for Aspect-Phrase Grouping. *Inf. Sci.* **2016**, *367–368*, 689–699. [CrossRef]
23. Xue, W.; Zhou, W.; Li, T.; Wang, Q. MTNA: A Neural Multi-task Model for Aspect Category Classification and Aspect Term Extraction On Restaurant Reviews. In Proceedings of the 8th International Joint Conference on Natural Language Processing, Taipei, Taiwan, 27 November–1 December 2017; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2017; pp. 151–156.
24. Li, X.; Bing, L.; Li, P.; Lam, W.; Yang, Z. Aspect Term Extraction with History Attention and Selective Transformation. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; pp. 4194–4200. [CrossRef]
25. Wu, C.; Wu, F.; Wu, S.; Yuan, Z.; Huang, Y. A Hybrid Unsupervised Method for Aspect Term and Opinion Target Extraction. *Knowl.-Based Syst.* **2018**, *148*, 66–73. [CrossRef]
26. Xiang, Y.; He, H.; Zheng, J. Aspect Term Extraction Based on MFE-CRF. *Information* **2018**, *9*, 198. [CrossRef]
27. Akhtar, M.S.; Garg, T.; Ekbal, A. Multi-Task Learning for Aspect Term Extraction and Aspect Sentiment Classification. *Neurocomputing* **2020**, *398*, 247–256. [CrossRef]
28. Augustyniak, Ł.; Kajdanowicz, T.; Kazienko, P. Comprehensive Analysis of Aspect Term Extraction Methods using Various Text Embeddings. *Comput. Speech Lang.* **2021**, *69*, 101217. [CrossRef]
29. Park, H.j.; Song, M.; Shin, K.S. Deep Learning Models and Datasets for Aspect Term Sentiment Classification: Implementing Holistic Recurrent Attention on Target-Dependent Memories. *Knowl.-Based Syst.* **2020**, *187*, 104825. [CrossRef]
30. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237. [CrossRef]
31. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual String Embeddings for Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1638–1649.
32. Kenter, T.; Jones, L.; Hewlett, D. Byte-Level Machine Reading Across Morphologically Varied Languages. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 5820–5827.
33. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
34. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781v3.
35. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 22–24 June 2014; Volume 32, pp. 1188–1196.

36. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186. [CrossRef]
37. Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. *Biol. Cybern.* **1982**, *43*, 59–69. [CrossRef]
38. Kohonen, T. The Self-Organizing Map. *Proc. IEEE* **1990**, *78*, 1464–1480. [CrossRef]
39. Kohonen, T. *Self-Organizing Maps*, 1st ed.; Springer Series in Information Sciences; Springer: Berlin/Heidelberg, Germany, 1995; Volume 30. [CrossRef]
40. Kohonen, T. Essentials of the Self-Organizing Map. *Neural Netw.* **2013**, *37*, 52–65. [CrossRef]
41. Riese, F.M.; Keller, S.; Hinz, S. Supervised and Semi-Supervised Self-Organizing Maps for Regression and Classification Focusing on Hyperspectral Data. *Remote Sens.* **2020**, *12*, 7. [CrossRef]
42. Maaten, L.v.d. Learning a Parametric Embedding by Preserving Local Structure. In Proceedings of the 2009 Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009; pp. 384–391.
43. Maaten, L.V.D. Accelerating t-SNE using Tree-Based Algorithms. *JMLR* **2014**, *15*, 3221–3245.
44. Maaten, L.V.D.; Hinton, G. Visualizing Data using t-SNE. *JMLR* **2008**, *9*, 2579–2605.
45. Maaten, L.V.D.; Hinton, G. Visualizing Non-Metric Similarities in Multiple Maps. *Mach. Learn.* **2012**, *87*, 33–55. [CrossRef]
46. Arthur, D.; Vassilvitskii, S. k-means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; Volume 1, pp. 1027–1035. [CrossRef]
47. Ali, N.M.; Novikov, B. A Multi-Source Big Data Framework for Capturing and Analyzing Customer Feedback. In Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), Moscow, Russia, 29 January–1 February 2018. [CrossRef]
48. Ali, N.M.; Gadallah, A.M.; Hefny, H.A.; Novikov, B. An Integrated Framework for Web Data Preprocessing Towards Modeling User Behavior. In Proceedings of the 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 6–9 October 2020; pp. 1–8. [CrossRef]
49. Ye, H.; Yan, Z.; Luo, Z.; Chao, W. Dependency-Tree Based Convolutional Neural Networks for Aspect Term Extraction. In Proceedings of the 21st Pacific-Asia Conference, Jeju, South Korea, 23–26 May 2017; pp. 350–362. [CrossRef]