IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 4, 2020, accepted November 23, 2020, date of publication December 3, 2020, date of current version December 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042312

Unsupervised Semantic Approach of Aspect-Based Sentiment Analysis for Large-Scale User Reviews

SUMAIA MOHAMMED AL-GHURIBI[1,2], SHAHRUL AZMAN MOHD NOAH[1], (Member, IEEE), AND SABRINA TIUN[1]

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[2]Department of Computer Science, Faculty of Applied Sciences, Taiz University, Taiz, Yemen

Corresponding author: Shahrul Azman Mohd Noah (shahrul@ukm.edu.my)

This work was supported by the Universiti Kebangsaan Malaysia through the Prime Impact Fund under Grant DIP-2020-017.

ABSTRACT Aspect-based sentiment analysis (ABSA) has recently attracted increasing attention due to its extensive applications. Most of the existing ABSA methods been applied on small-sized labeled datasets. However, real datasets such as the Amazon and TripAdvisor contain a massive number of reviews. Thus, applying these methods on large-scale datasets may produce inefficient results. Furthermore, these existing methods extract huge number of aspects, most of which are not relevant to the domain of interest. But, on other hand, some of the infrequent relevant aspects are excluded during the extraction process. These limitations negatively affect the performance of the ABSA process. This article, therefore, aims to overcome such limitations by proposing an efficient approach that is suitable for real large-scale unlabeled datasets. The proposed approach is a combination of hybridizing a frequency-based approach (word level) and a syntactic-relation based approach (sentence level). It was enhanced further with a semantic similarity-based approach to extract aspects that are relevant to the domain, even terms (related to the aspects) are not frequently mentioned in the reviews. The extracted aspects according to the proposed approach are used to generate a total review sentiment score after estimating the weight and the rating of each extracted aspect mentioned in the review. The assignment of the weight of each extracted aspect is calculated based on a modified TF-IDF weighting scheme and the assignment of the aspect rating is calculated based on a domain-specific lexicon. Effectiveness of the extracted aspects is evaluated against two baselines available from existing literature: fixed aspect and extracted aspects. Evaluation was also performed by using a general lexicon and a domain-specific lexicon. Results in terms of F-measure and accuracy on Amazon and Yelp datasets show that the extracted aspects using the proposed approach with the domain-specific lexicon outperformed all the baselines.

INDEX TERMS Aspect, core terms, aspect extraction, aspect weight, aspect rating, domain-specific lexicon, total review score, real large-scale dataset.

I. INTRODUCTION

Recently, the sharing of experiences among customers is becoming a widespread phenomenon in social media sites. Many customers make decisions on consuming a service based on the opinions of others. Due to this phenomenon, there has been a rapid growth in the number of online opinions (i.e., user reviews), where each review expresses the customer's opinion of the used service, such as buying a product, watching a movie, or reserving a room. Such reviews are considered a valuable resource for both consumers and businesses. Despite the benefit of these reviews, the extraction of useful information from such reviews is a huge challenge due to its large scale and distinct characteristics [1]. Research to overcome such issues have been proposed by many fields, which, among others, were related to mining distinct and important information [2]–[4], extracting users' sentiments [5], [6] or summarizing user reviews [7], [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed A. Zaki Diab.

In this article, we focus on the problem of mining distinct and important information from the reviews. Specifically, the aim of this article is to develop an efficient approach for extracting aspects from reviews and exploit them for sentiment analysis. Such an approach is known as Aspect-Based Sentiment Analysis (ABSA) and is the third level of the sentiment analysis categories after the document-based and the sentence-based sentiment analyses. Aspect extraction is one of the natural language processing (NLP) tasks that use sentiment analysis (SA). There are also many NLP tasks that use SA such as word polarity disambiguation [9], polarity detection [10], and sarcasm detection [11], but we focus on aspect extraction task only. ABSA determines the user's sentiments expressed on each aspect mentioned in the reviews [12]. Aspect usually refers to a concept that represents a topic of an item in a specific domain, such as *price*, *taste*, *service,* and *cleanliness* which are relevant aspects for the *restaurant* domain. ABSA has increasingly become a trend that attracts the attention of many researchers since it helps to understand fine-grained opinion mining over coarse-grained [13]. It has an obvious effect in many applications when the aspects are extracted efficiently with their opinions' values. For example, they can be used to construct user and item profiles in decision-making processes, such as recommender systems [1], [14], or to construct a domain ontology [15]. Additionally, it has been used in many systems for different purposes such as tourism [16], online education [17], and transportation [18]. Many research projects proposed approaches to handle the ABSA problem. Most of them focus on labeled small-sized datasets as opposed to real, unlabeled and large datasets. Applying such ABSA approaches that are designed for labeled small-sized datasets to unlabeled large-scale ones would negatively affect the ABSA method and yield inefficient performance. Additionally, some of this research has limitations in terms of extraction and sentiment analysis processes. The key drawback is that these methods produce a large number of aspects but neglect the importance of the experimental domain of these aspects. Moreover, approaches based on words' frequencies during extraction lose the infrequent aspects even if they are relevant to the domain. We can sum up the other weakness of certain ABSA approaches as seen in Table 1.

TABLE 1. Limitation of Selected ABSA Approaches.

Approaches of ABSA	Limitations
Aspects are considered as some pre-defined (fixed) features for items and deal with the aspect extraction as a classification problem [19].	Aspects are limited to the pre-defined features, whereas users' reviews may contain other aspects not represented by the pre-defined features.
Aspects are extracted from reviews with some specific assumptions, such as limiting the sentence with one aspect or calculating one sentiment score for one sentence [20, 21].	The assumptions are not true in many cases, as a sentence may contain multiple aspects and multiple sentiment scores.
Comparative opinions are not considered during the extraction of aspects [21, 22].	Comparative opinion words provide valuable information between two items about a specific aspect.

The aforementioned limitations foresee the need to develop an efficient ABSA approach for unlabeled large-scale datasets. Thus, the main aim of this research is to propose an efficient approach that is able to extract aspects relevant to the domain of interest and enhance the ABSA process for unlabeled large-scale datasets. The proposed approach aims to fulfill four tasks of ABSA, without making assumptions for the aspects or their sentiments words, as follows:

- Extracting aspects
  For the aspect extraction task, a novel method is proposed consisting of a hybrid approach, including a frequency-based approach (word level) and a syntactic-relation based approach (sentence level), that works in parallell and followed by a semantic similarity-based approach.
- Estimating aspects' weight
  For estimating an aspect's weight task, three weighting methods are explored and the one that will give the best performance for the ABSA process will be chosen. The methods are the conventional Term Frequency–Inverse Document Frequency (TF-IDF), and two modified TF-IDF weighting schemes proposed by Zhu *et al.* [23] and Ngoc *et al.* [2].
- Inferring aspects' rating
  For this task, an algorithm is proposed to extract the aspect sentiment pairs, assign sentiment scores for the sentiment words of the aspects using the domain-specific lexicon developed by Al-Ghuribi *et al.* [24].
- Calculating total review score
  For calculating total review score task, an algorithm is proposed to calculate the total review sentiment score based on the work of [2], [4], [25]. The algorithm takes the results of the previous three tasks as inputs (i.e., the extracted aspects with the core terms, aspects' weights, and the domain-specific lexicon for calculating the aspects' ratings). The output of this algorithm is the total review sentiment score (i.e. overall rating) for each review.

The rest of the paper is organized as follows: section two introduces related works in the area, while section three presents the methodology adopted in developing the proposed approach. Section four provides the experimental results of the four tasks of ABSA and is followed by the evaluation in section five. Finally, we present the conclusion that can be drawn from this research work in section six.

II. RELATED WORK

The ABSA problem was first defined by Hu and Liu [8] and, since then, it has received great attention from many researchers. Additionally, many surveys discussed the main challenges and issues related to this field. The recent survey by Nazir *et al.* [26], identifies two main tasks for ABSA: aspect extraction and aspect sentiment analysis.

The tasks in ABSA are classified into three categories: supervised, semi-supervised and unsupervised. Supervised approaches require labeled aspects, whereas the unsupervised approaches do not require such label datasets for the aspect extraction process. Unlike the two previous approaches, the semi-supervised approaches require both labeled and unlabeled data for the extraction process [27]. These categories of approaches have been applied in many research. For example the work of [28]–[31] represent techniques that utilized the supervised approach. While the work of [32]–[35] applied the semi-supervised approach in the ABSA. The research work presented in this article focuses on unsupervised methods as supervised learning requires data annotation which is time consuming [36]–[38] and suffers from domain adaptation problems [39]. The unsupervised method has been adopted to avoid depending on labeled data, since there is no need to separately perform extraction and categorization to obtain the aspects [40].

The unsupervised methods for ABSA can be classified into four categories [41]: vocabulary-based, frequency-based, syntactic relation-based, and topic model-based methods. In the vocabulary-based method, a fixed pre-defined list for aspects is used. Few researchers rely only on the pre-defined list for identifying and extracting aspects, such as Aciar *et al.* [42], while others use it to extract other aspects that are related to the elements in the list [43]. It was claimed that learned aspects (i.e., aspects extracted from the reviews) generate better overall sentiment results as compared to the fixed pre-defined aspects. This is due to the fact that the number of the aspects in the fixed pre-defined list is limited, and there is no guarantee that these aspects will occur in the users' reviews [1], [44]. As a result, few researchers use the vocabulary-based method.

The most used method for extracting the learned aspects is the frequency-based method [27], [45]. Despite its simplicity, it is very effective and used by many researchers [41]. The main idea of this method is to extract high occurrences words in reviews (i.e. words that are frequently mentioned by users to express their opinions). The candidate words for aspects are the noun and noun phrase [3], [8], [12], [20], [44], [46]. If the frequency of the candidate word exceeds some threshold value, the word is considered as an aspect. Once the aspect is extracted, the aspect's sentiment word is selected based on the nearest adjective to the specific aspect. Finally, the selected sentiment word is assigned a polarity value (i.e. a score) based on some lexicons. The work of Caputo *et al.* [44] and Mubarok *et al.* [20] are among the recent works that use the frequency-based method.

Caputo *et al.* [44] proposed a system for opinion retrieval called a sentiment aspect based retrieval engine (SABRE) which consists of four tasks: extract aspects and its sub-aspects; find the opinion associated with each aspect; detect the polarity for each sentiment opinion word; and retrieve documents for a given opinion. The extraction of aspects is based on term-frequency probabilities and a model that calculates the difference distribution for a word between a specific domain and a general corpus using the non-symmetric Kullback-Leibler divergence technique. For opinion word extraction and its polarity, a lexicon-based approach is applied using the AFINN wordlist. Finally, for document retrieval, the TF-IDF weighting scheme is used to retrieve the top-N documents with the highest opinion scores. The experiment of the proposed method on the TripAdvisor dataset containing 167,780 reviews outperformed the conventional term frequency method in terms of F-measure.

On the other hand, Mubarok *et al.* [20] proposed an approach for extracting sentiment polarity based on specific aspects of product reviews. The model creates two bag-of-words models where one of the models contains the aspects (i.e., nouns), and the other contains the sentiment words (i.e., adjectives or adverbs). The words in both lists are selected using a chi-square test by choosing the words with the highest relevance for each opinion. The Naive Bayes classifier is then used to classify the sentiment polarity of each aspect. The model is evaluated using the SemEval-2014 dataset which contains 3,618 reviews for the *restaurant* domain consisting of five aspects (i.e., *price*, *food*, *ambience*, *service*, and *miscellaneous*). The model is compared against 17 baselines. The proposed method received the seventh highest F-measure among the 17 compared baselines with an F-measure = 78.12%.

Although the frequency-based method is an efficient one, it has obvious limitations. One of the limitations is that the approach may select words that are not aspects (i.e., pick up many words that do not contain any subjectivity) because it relies only on word frequencies. Furthermore, aspects that are not frequently mentioned will not be detected using this method. However, the syntactic relation-based and topic model-based methods can address such a limitation.

The syntactic relation-based method (also called the rule-based method) aims to analyze the syntactic structure of the sentence and the relations among the words to identify the aspect's sentiment words. A well-known algorithm that uses this method is the Double Propagation (DP) proposed by Qiu *et al.* [47]. The algorithm describes the syntactic relation between nouns or noun phrases with adjectives using dependency grammar. This method has been used as a baseline for many other methods, while others, such as Poria *et al.* [48], tried to improve it by expanding the rules for the relations extraction. Other research use dependency parsing to define the relation among the words such as the works of [6], [49]–[53]. Following are description of some prominent works based on this method as follows:

Chen and Yao [51] presented an approach defining the opinion words' relations using both dependency parsing and shallow semantic analysis, then built an ontology and a collocation (i.e., the most frequently co-existing topic and sentiment pairs') database. The method is applied on two datasets used in [8], [54] containing of 500 and 2500 reviews, respectively. Both datasets are mainly from product domains such as laptops, cameras, printers, and DVDs. The proposed method outperforms two baselines, a naive baseline in which the final polarity is calculated by the major number of positive or negative opinion words in the sentence, and [8] as baseline 2, in terms of accuracy. Mukherjee and Bhattacharyya [6] proposed a method for identifying the features and their corresponding opinions in the product reviews using dependency parsing to define the short and long dependencies between words. An experiment is performed using a Chinese corpus on the car domain collected from Internet product reviews. The results of the proposed approach outperform the compared baselines (of closest-pair and dependency parsing) in terms of precision, recall, and F-measure. Hai *et al.* [50] used three syntactic dependency rules with two domain corpus to extract the aspects from two domains (i.e., cellphones and hotels) of Chinese reviews. The candidate aspects are firstly extracted using the three rules then these aspects are filtered based on their relevance to the domain using two relevance measures, namely intrinsic-domain relevance and extrinsic-domain relevance. The F-measure results of their conducted experiments on the Chinese reviews were 63.6% and 52.2% for the two domains, respectively. Nejad *et al.* [53] is one of the recent research that employed an unsupervised approach for detecting explicit features in Persian language for hotel domain. Their methodology consists of three steps, text preprocessing, sentimental vocabulary construction, and aspect extraction. A directed weighted graph is constructed based on frequent pattern identification from the sentences of their Persian corpus. The paths within the constructed graph are determined based on some developed rules to extract multi-word aspects. The proposed approach is evaluated and compared with some existing approaches that works on Persian language, and it gives the best F-measure value.

Similar to the frequency-based method, the syntactic relation-based method produces noise in terms of non-related aspects due to only focusing on the subjective expressions (i.e., sentence structures) and ignoring the words' semantics. In addition, not all the used rules are effective for extraction and not all the extraction pattern rules are explored [28].

Topic model-based method addresses this problem by focusing on the semantics of the words. This is because the method reveals topics from a large collection of texts whereby words are grouped into aspects or topics. For example, users talk about *price* using words like *money*, *budget*, and *cost*, which should not be regarded as different aspects. The topic modeling is based on two basic models, Latent Dirichlet Allocation (LDA) [55] and Probabilistic Latent Semantic Analysis (PLSA) [56], for learning latent topics (i.e., the local topics) that have a direct correlation with the aspects (i.e., the general topic). It is used by many researches such as [4], [36], [57]–[61]. For example, in the work of Brody and Elhadad [59], a local LDA is applied on restaurant reviews dataset by assuming that each sentence in the reviews is a separate document in order to extract low frequent aspects. The work of Lin and He [61] and Moghaddam and Ester [60] extend the standard LDA. In the work of [61], an additional sentiment layer is added to the basic LDA model to develop a Joint Sentiment Topic Model, in which, the topic word is not separated from the sentiment word. The work of [60] extend the LDA to Interdependent Latent Dirichlet Allocation based on the assumption that there is an interdependency relation between the aspect and the sentiment words. In other work, McAuley *et al.* [62] proposed a probabilistic model that benefited from the ratings associated with the reviews to learn words that correlate with aspects or specific ratings. For instance, the word *appearance* may be used to represent the *look* aspect, and the word *delicious* may refer to a high rating. In order to build this model, three learning methods are used: supervised, semi-supervised, and unsupervised. They introduced a new dataset consisting of 5,000,000 reviews, where each user provides ratings for each aspect of a product. The model is evaluated on three prediction tasks: determining the parts of the review that discuss the rated aspects, finding the sentences that determine the user's rating, and predicting the non-rating user's aspects. The authors claimed that their model is suitable for real datasets and has the ability to determine the reviews' parts that relate to each aspect and select the sentence that best summarizes the review.

In recent years, deep learning approaches received special attention in SA studies generally and in ABSA particularly [63]. LDA is combined with deep learning techniques to enhance the aspect extraction process such as the works of [36], [64]–[68]. We illustrate the topic modeling approach with deep learning by referring to the work of Garcia *et al.* [36], and Chauhan *et al.* [67].

García-Pablos *et al.* [36] proposed an unsupervised approach called W2VLDA which is based on topic modelling combined with continuous word embeddings and a maximum entropy classifier. The approach consists of three subtasks: aspect classification, sentiment classification, and aspect/opinion word separation. The performance of the approach is evaluated in the multilingual SemEval-2016 task 5 dataset [69]. It is tested for three domains, electronic devices, restaurants, and hotels and for four languages, English, French, Spanish and Dutch. Therefore, it outperforms two of the conventional LDA-based approaches. Chauhan *et al.* [67] integrated rule based method with Bidirectional Long-Short-Term-Memory (Bi-LSTM) model to extract the aspects. The rule-based method is used to extract the candidate aspects from noun and noun phrases follows by the Bi-LSTM model to filter the candidate aspects and select the correct ones. Similar to García-Pablos *et al.* [36] work, the SemEval-2016 dataset is used for evaluation involving the restaurant and laptop domains. The results of the approach missed many aspects during the extraction process because it only consider the nouns and noun phrases [28]. Some recent works combine both Conditional Random Field (CRF) and (Bi-LSTM) models in the aspect extraction process such as Liang *et al.* [68] and Gandhi and Attar [70]. The former use SemEval 2014 and 2015 [71] datasets and the latter use Hindi dataset to evaluate their approaches. While in other work an extension of LSTM is proposed such as Ma *et al.* [72]. They proposed two methods, the first is a Sentic LSTM that contains a separate output gate that interpolates both concept-level input and token-level memory. The second method is an extension of the Sentic LSTM, it merges the LSTM and a recurrent additive network that simulates sentic patterns. The performance of the proposed methods is evaluated on both SentiHood dataset and SemEval 2015 dataset. Results show the effectiveness of the proposed methods in both aspect categorization and aspect-based sentiment classification tasks.

The topic model-based method has two main limitations [2], [21]. First, it's restriction of use in real-life sentiment analysis applications, because the method will not achieve reasonable and efficient results if the size of the data is small. This makes such a method unsuitable for many practical sentiment analysis applications. The second limitation is that the result of LDA contains more global topics than local ones because it is designed for the document-level, thus using it at the aspect-level is not a wise decision.

Another unsupervised method for ABSA is based on the bootstrap technique [73] which is used for statistical inference without depending on many assumptions. Many researchers adopted this method, thus, we consider it as the fifth method for ABSA such as in [2], [74], [75]. The following is an explanation of one of the recent research that uses this method for ABSA.

Ngoc *et al.* [2] proposed a method that combines the conditional probabilistic model and a bootstrap technique to extract product's aspects. For each extracted aspect, an inferred rating is calculated by dealing with it as a multi-label classification using the Naive Bayes classifier. Each aspect's weight is predicted based on the occurrences in which the user discusses the aspect within the reviews. This method does not use the overall rating in the aspect rating and weight calculation, but only uses the review text. Experiments were carried out for the domain of *hotel* [4], *beer* [62], and *coffee*. Each domain involved seven, five, and four aspects, respectively. Precision is used to evaluate aspect extraction, mean square error measure is used to evaluate aspect rating, and for the evaluation of aspect's weight, the overall rating from reviews' text is calculated and compared with the overall rating given by the user. The proposed method outperformed Long's method [76] which is used as the baseline.

In a nutshell, there have been active studies into the handling of ABSA, of which we have mentioned a few significant and influential ones relevant to our scope of work. Through this study, we intend to introduce a hybrid approach for unlabeled large-scale datasets, whereas most of the existing methods concentrate on labeled small-scale datasets and whose implementation in large-scale datasets would yield inefficient results. Additionally, the proposed method aims to overcome the two previously discussed limitations:

- Most of the existing approaches extract a large number of aspects but not all of the extracted aspects relevant to the domain of the reviews, and, thus, negatively affect the performance of ABSA [77].
- Most of the existing approaches rely on word frequencies, resulting in many infrequent words that are relevant to the domain being ignored.

In this research, we aim to handle these limitations by proposing a semantically enhanced aspect extraction approach. As mentioned earlier, the proposed approaches consist of four task: extracting aspects; estimating aspects' weights; estimating aspects' ratings and calculating total review scores.

An example illustrating the previous four tasks is as follows.

Assume a user gives the following review:

"*This is a charming version of the classic Dicken's tale. Henry Winkler makes a good showing as the "Scrooge" character. The casting is excellent and the music old but very relevant.*"

The proposed approach should be able to extract five different aspects from the above review using a semantically enhanced aspect extraction method. The extracted aspects are *version*, *tale*, *showing*, *casting,* and *music.* The extraction process is then followed by assigning a weight for each of the extracted aspects using a modified TF-IDF weighting scheme. Using an algorithm proposed in this article, all the sentiment words associated with the previous five extracted aspects can be extricated. For example, the word *charming* is the sentiment word for the aspect *version*, and the words *very relevant* are the sentiment words for the aspect *music*. After that, the sentiment score of each aspect is calculated using a domain-specific lexicon. Finally, the score of each aspect with its weight are aggregated to generate the total review sentiment score. For the above review, the overall review sentiment score is greater than zero which indicates its positiveness. This review is taken from the Amazon dataset, where each row of the dataset contains the review text and the overall rating. The overall rating of the aforementioned example is five, thus, it shows that our approach works by giving it a positive score. Table 2 illustrates the details of the given example.

Details of the proposed approach is discussed in the following section.

TABLE 2. **Applying the Proposed ABSA Approach on a Sample of Amazon Review.**

Aspect	Aspect Weight	Aspect Rating		Aspect Sentiment Pair's Score (weight*score)	
		Sentiment Words	Sentiment Scores		
version	0.3801	charming	0.0310	0.01178	
tale	0.1332	classic	0.2444	0.03255	
showing	0.0736	makes good	0.2582	0.01900	
casting	0.0703	excellent	0.2640	0.01856	
music	0.3163	old	0.1105	0.03495	
music	0.3163	very relevant	0.0107	0.00338	
Total Review Sentiment Score				0.12022	Positive Review

III. THE PROPOSED APPROACH

As mentioned earlier, the proposed approach is divided into four tasks: extracting aspects, estimating aspect weight, inferring aspect rating, and calculating the total review score. Each task is further explained in the following sections.

A. ASPECT EXTRACTION

The aspect extraction method consists of three main tasks: extraction of noun and noun-adjective pairs, creation of aspect dictionary, and generation of aspects and core terms. We called this method Semantically Enhanced Aspect Extraction (SEAE) whereby the aim is to extract aspects that are relevant to the domain. Figure 1 illustrates the process of the proposed SEAE method which is further described in the following sub-sections.

1) EXTRACTION OF NOUN AND NOUN-ADJECTIVE PAIRS

The purpose of this task is to extract noun and noun-adjective pairs from the textual reviews as both sets of words have the potential to be classified as aspects [20], [44]. In this task, two approaches are used: the frequency-based and the syntactic-relation based approaches. As both approaches run in parallel, we called it the Parallel Hybridization Approach (PHA).

The Frequency-Based Approach (FBA) extracts nouns from the dataset, calculates their occurrences, then stores the extracted nouns with their frequencies in blocks (i.e., clusters) using the blocking technique discussed in [78]. Most of the existing approaches merely present a fixed list of aspects, and there is no guarantee that the listed aspects are presented in the user-generated review texts. Thus, aspects should be learned and subsequently extracted in such a way that they are sufficiently represented in the user reviews. The words that represent aspects are mostly nouns as proven by many studies [3], [20], [44]. As a result, this phase mainly focuses on calculating the occurrence of nouns (other categories of words that may refer to aspects will be taken care in the third task of SEAE). To summarize, this approach aims to take

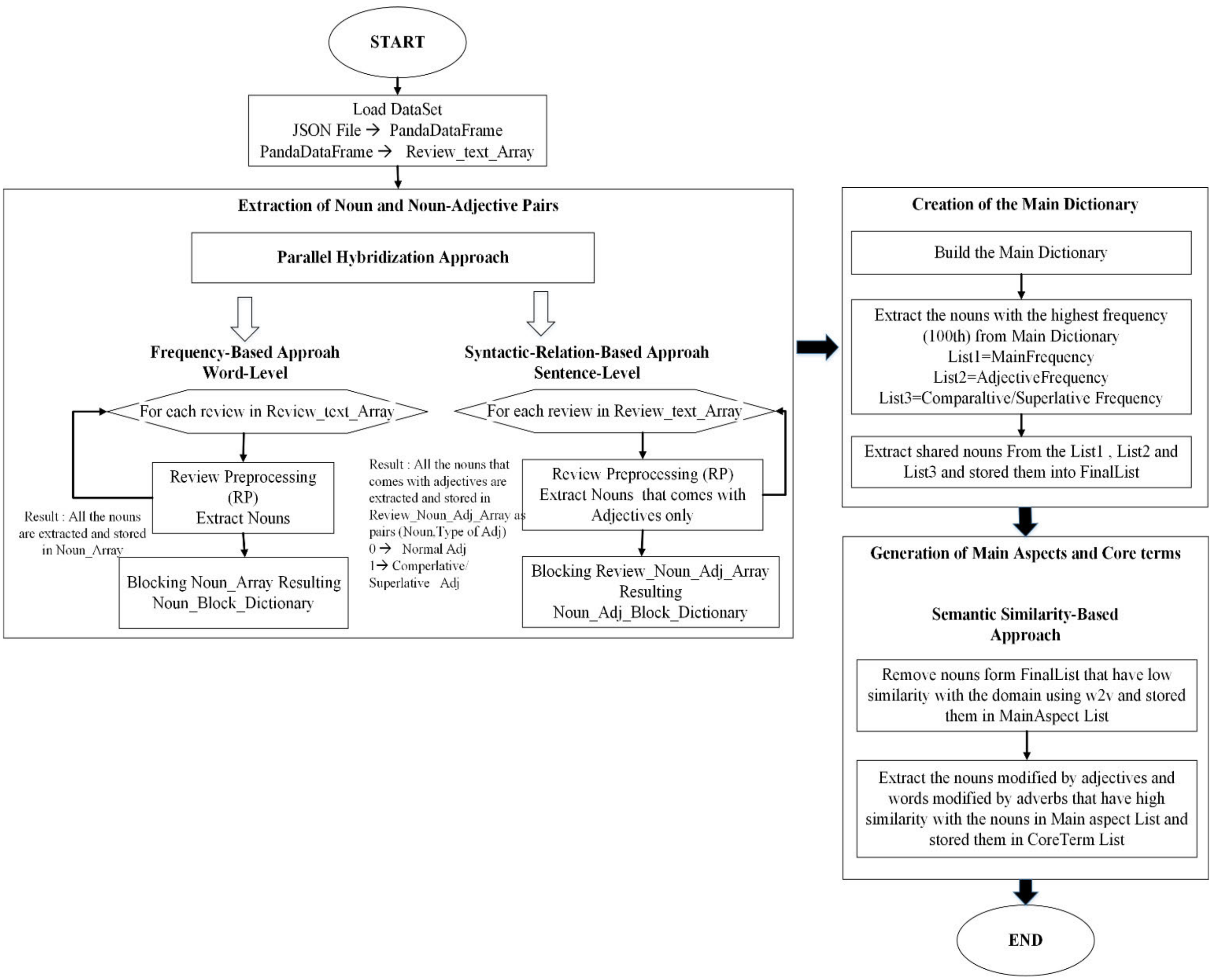

FIGURE 1. The general approach of the semantically enhanced aspect extraction method.

```
Function Frequency-Based is    #Word_Level
 Input: Review_text_Array
 Output: Noun_Block_Dictionary
  #Initalize the Noun_Block_Dictionary (27 Blocks Named A, B, C to Z)
 For each block_no in Noun_Block_Dictionary do
     Noun_Block_Dictionary[block_no] ['Word']←[]
     Noun_Block_Dictionary[block_no] ['Frequency']←[]

 Total_Extracted_Nouns  ← 0
 For each review_text in Review_text_Array do
    Word_Tokenized ← Split the review_text into tokens using NLTK
    Clean the Word_Tokenized list by Remove Numbers, Remove Special Characters, and Split Words from Special Characters
    Tagged_list ← NLTK.POS_TAG (Word_Tokenized)
    Words, Tagger ← Zip (* Tagged_list)
    For each word in Words do
      IF (Tagger (Word) → Noun)
          IF (len (Word)>2 and Not StopWord)
             Candidate_Aspect ← Lemma (word)
             Total_Extracted_Nouns + ← 1   #To present how many extracted word
             First_Letter ← Candidate_Aspect [0]  #first letter of the word
             Ascii_Letter ← Ascii (Capitalize (First_Letter)) #Capitalize convert letter from small to capital one
          Block_no ←Ascii_Letter-65   # Ascii no of A is 65
          flag_findword=0, word_index ← 0
          For each word_inside_Block in Noun_Block_Dictionary [Block_no] ['Word'] do
             word_index ← word_index+1
             IF word_inside_Block = Candidate_Aspect     #Check if the Candidate_Aspect is inserted before to the block
                  flag_findword=1 #The Noun_Word is previously added
                   #Increase the Noun frequency by one
                  word_frequency ← Noun_Block _Dictionary [ Block_no] ['Frequency'] [word_index]
                  Noun_Block _Dictionary [ Block_no] ['Frequency'] [word_index] = word_frequency+1
                  Break        #Do not go through the rest of the words in the block
             IF flag_findword=0   #The noun is not added before to the block
                   #add the noun to the block and put its frequency equal to 1
                Noun_Block _Dictionary [ Block_no] ['Word']. append (Candidate_Aspect)
                Noun_Block _Dictionary [ Block_no] ['Frequency']. append (1)
```

FIGURE 2. Frequency-based approach.

the user-generated reviews (i.e., text), extract nouns from all the reviews in the dataset, and calculate the frequencies of each extracted nouns. Then, the nouns, together with their frequencies, are stored in blocks using the blocking technique mentioned in [78].

The main purpose of using the blocking technique is to arrange words in blocks in order to accelerate the searching part in our large-scale experimental datasets by reducing the number of required search comparisons. There are 27 blocks, named from A to Z, and numbered from 0 to 26. Each block is specialized for storing the nouns that begin with the same letter of the block name with their frequencies. For example, Block A contains all the extracted nouns beginning with letter A, together with their frequencies. In more detail, when a noun is extracted from a user review, the block is first identified through some simple steps: first, the first letter of the noun is converted into a capital letter, then the relevant ASCII number it is found. Suppose the extracted noun is *movie*. Thus, the associated ASCII number to M (77) will be identified. The blocks are numbered from 0 to 26, the M block needs to be determined in order to add the word *movie* to it. This is done by subtracting the ASCII number of the first letter from the ASCII number of letter A (65) which is used as the first block. Therefore, the block number for the word *movie* will be $77 - 65 = 12$. After the block for the extracted word has been identified, the processing of the word can be performed in the identified block. In the case of the word *movie*, the approach will search only Block 12 for the word *movie*. If the word is already stored in the block, its frequency will be updated. However, if the word does not exist, the word *movie* will be added to the block. Figure 2 illustrates the algorithm of the FBA approach which takes all the reviews' text as input and produces the *Noun_Block_Dictionary* as output.

The Syntactic-Relation-Based Approach (SRBA), on other hand, aims to extract the noun-adjective pairs using grammatical roles and calculate the frequencies of each noun that comes with the adjectives based on the adjectives' degree (i.e., positive adjective or comparative/superlative adjective). Then, it stores the extracted nouns with their frequencies in the blocks. The syntactic relations or grammatical roles refer to functional relationships between constituents in a phrase or a clause. In this approach, we focus on the relation between the noun and adjective using the Java StanfordCoreNLP

```
Function Syntactic-Relation Based is   #Sentence_Level
 Input: Review_text_Array
 Output: Noun_Adj_Block_Dictionary
  #Initalize the Noun_Adj_Block_Dictionary (27 Blocks Named A, B, C to Z)
 For each block_no in Noun_Adj_Block_Dictionary do
     Noun_Adj_Block_Dictionary [block_no] ['Word']←[]
     Noun_Adj_Block_Dictionary [block_no] ['Adj_Frequency']←[]
     Noun_Adj_Block_Dictionary [block_no] ['Comp_Sup_Frequency']←[]
  #Java StanfordCoreNLP library is used so we must run the StanfordCoreNLP Server
 Total_Extracted_Nouns  ← 0   Total_Extracted_Nouns2  ← 0
 For each review_text in Review_text_Array do
   Sentence_Tokenized ← Split the review_text into Sentences
   For each Sentence in Sentence_Tokenized do
     Word_Tokenized ← Split the review_text into tokens using NLTK
     Clean the Word_Tokenized list by Remove Numbers, Remove Special Characters, and Split Words from Special Characters
                                 then append them into Final_Sentence
     IF len (Final Sentence > 1)   # >1 means Sentence is not empty
      Output ← Use 'depparse' annotator to annotate Final Sentence
      For each Relation in Output ['sentences'] ['enhancedDependencies'] do
        IF (Relation['dep'] = 'amod')   # amod: Adjective modifier
          Noun = Relation ['governorGloss']
          Adjective = Relation ['dependentGloss']
          Noun_Adj ←' '
          IF (len (Noun)>2 and Not StopWord)
            Total_Extracted_Nouns + ← 1
            IF (Adjective ['pos'] = 'JJR' OR 'JJS') # come with comparison
                  Noun_ Adj ← (lemma (Noun),1)
            Else
                  Noun_Adj ←(lemma (Noun),0)
            Total_Extracted_Nouns2 + ← 1
            Ascii_Letter ← Ascii (Capitalize (Noun_Adj [0][0]))
            Block_no ←Ascii_Letter-65
            flag_findword  ← 0, word_index ← 0
            For each word in Noun_Adj_Block_Dictionary [Block_no] ['Word'] do
                  #Check if the Noun_Word is inserted before to the block or not
               word_index ← word_index+1
               IF word = Noun_Adj
                  flag_findword=1 #The Noun_Word is previously added
                   #Increase the Total frequency by one and Comp_Sup_Frequency based on Noun_Adj[1]
                  F1←Noun_Adj_Block_Dictionary [ block_no] [ 'Adj_Frequency'] [word_index]
                  Noun_Adj_Block_Dictionary [ block_no] [ 'Adj_Frequency'][word_index]← F1+1
                  F2←Noun_Adj_Block_Dictionary [ block_no] ['Comp_Sup_Frequency'] [word_index]
                  Noun_Adj_Block_Dictionary [ block_no] ['Comp_Sup_Frequency'] [word_index] ← F2 +(Noun_Adj [1])
                  Break        #Do not go through the rest of the words in the block
            IF flag_findword=0   #The noun is not added before to the block
                   #add the noun to the block and put its total frequency and conp_sup_frequency
                  Noun_Adj_Block_Dictionary [ Block_no] ['Word']. append (Noun_Adj)
                  Noun_Adj_Block_Dictionary [ Block_no] ['Adj_Frequency']. append (1)
                  Noun_Adj_Block_Dictionary[block_no] ['Comp_Sup_Frequency']. Append (Noun_Adj [1])
```

FIGURE 3. Syntactic relation-based approach.

adjectival modifier. As mentioned before, aspects are mostly represented by nouns and the nouns that come with the adjectives have more probabilities to be chosen as aspects. For this reason, this approach focuses on the adjectives as modifiers. For example, the sentence "*The company produces a good movie*"; the *<movie, good>* pair results from the adjectival modifier, where *good* is the adjective that serves to modify the meaning of the noun *movie*. This approach aims to extract such adjective phrases and calculate its frequencies. It is known that the adjectives have three degrees: positive, comparative, and superlative [79]. The positive adjectives describe people, places, and things in a positive way, unlike the comparative and superlative adjectives that are used to compare two comparative or superlative entities. Current works [21], [22] do not use comparative and superlative adjectives in their extraction process, despite the fact that such a comparison of words is usually made between items in a specific aspect. Thus, apart from the frequencies of the nouns that come with all degrees of adjectives, the frequencies of nouns that come with the comparative and superlative adjectives are also being stored. Similar to FBA, the extracted nouns with their frequencies are stored in blocks using the blocking technique mentioned in [78] and consisting of information in the form of (<*Block Number*> <*Word*> <$freq_{Adjective}$> <$freq_{comparative/Superlative\_Adjective}$>). As a summary, this task aims to take the user-generated reviews (i.e., text), extract all the nouns that come with adjectives from all the reviews in the dataset, and calculate the two specified frequencies of each extracted noun. Figure 3 shows the algorithm of this approach which produces the *Noun_Adj_Block_Dictionary* as output.

2) CREATION OF THE MAIN DICTIONARY

The two processes in the previous task resulted in the creation of two dictionaries, the *Noun_Block_Dictionary* and the *Noun_Adj_Block_Dictionary* which result from the FBA and SRBA processes, respectively. The creation of the main dictionary step aims to merge the two dictionaries into a main dictionary named *Main_Dictionary* in which every word in the dictionary contains three different frequencies. The first frequency describes how many times the word appears as a noun in all of the reviews; this frequency is taken from the *Noun_Block_Dictionary*. The second frequency describes how many times the noun appears with the adjectives and the third frequency describes how many times the noun appears with the comparative or superlative adjectives; both are taken from the *Noun_Adj_Block_Dictionary*. The algorithm of building the main dictionary is as described in Figure 4.

Each row in the *Main_Dictionary* consists of four columns as shown in Figure 5: word, total frequency, adjective frequency, and comparative/superlative frequency. In this step, the top 100 most frequent words in the three columns are extracted, merged, and stored in a list called the *SharedWords* list.

The previous two tasks of SEAE (i.e., the extraction of noun and noun-adjective pairs and the creation of the main dictionary) were implemented for three domains: *movie*, *book*, and *restaurant*. Consequently, after implementing the previous two SEAE tasks for the three domains, we will obtain three *SharedWords* lists, one list for each domain. Finally, the last step of this task is a filtering step aimed at filtering the words in the *SharedWords* list by keeping only the words relevant to the domain and removing common words (i.e., words which are common in many domains).

This step aims to remove the words that occur in all of the three *SharedWords* lists to differentiate between the common words and the specific words for each domain and to subsequently store the relevant words for each domain in a *Finallist* list. Words that commonly appear in many domains have little content-bearing and limited discrimination capabilities. Thus, they will not give positive results in ABSA.

3) GENERATION OF MAIN ASPECTS AND CORE TERMS

The generation of the final main aspects and core terms (i.e., words that have high similarity values with the main aspects) is based on the semantic similarity-based approach which aims to overcome two inherent problems in ABSA: problems related to a large number of aspects being extracted but not all of them being relevant to the domain [77] and problems relating to infrequent nouns (non-popular nouns) being ignored despite their importance to the domains of interest [77]. In other words, due to the nature of user-generated reviews, different users write their reviews using different words without referring to any standards. Some users use different words but with similar intentions and meanings.

```
Function Main_Dictionary_Building is
 Input: Noun_Block_Dictionary and Noun_Adj_Block_Dictionary #NBD and NABD
 Output: Main_Dictionary   #MD will be used for shortening
 #Initalize the Main_Dictionary (27 Blocks Named A, B, C to Z)
 For each block_no in MD do
      MD [block_no] ['Word']←[]
      MD [block_no] ['Total_Frequency']←[]
      MD [block_no] ['Adj_Frequency']←[]
      MD [block_no] ['Comp_Sup_Frequency']←[]
 For each blockno in MD do
     For each word_in_block in NBD [ blockno] ['Word']
        MD [blockno]['Word']. append (NBD [blockno] ['Word'])
        MD [blockno]['TotalFrequency']. append (NBD [blockno] ['Frequency'])
        Found_NABD=0
        For each word in NABD [ blockno] ['Word'] do
         IF word_in_block = word   #The word is occurred in both NBD and NABD dictionaries
           Found_NABD = 1
           MD [blockno] ['Adj_Frequency']. append (NABD [blockno] [ 'Adj_Frequency']
           MD [blockno] ['Comp_Sup_Frequency']. append (NABD [blockno][ 'Comp_Sup_Frequency']
           Break
         IF Found_NABD = 0 # The word does not occur on NABD
           MD [blockno] ['Adj_Frequency']. append (0)
           MD [blockno] ['Comp_Sup_Frequency']. append (0)
```

FIGURE 4. Steps for building the ***Main_Dictionary***.

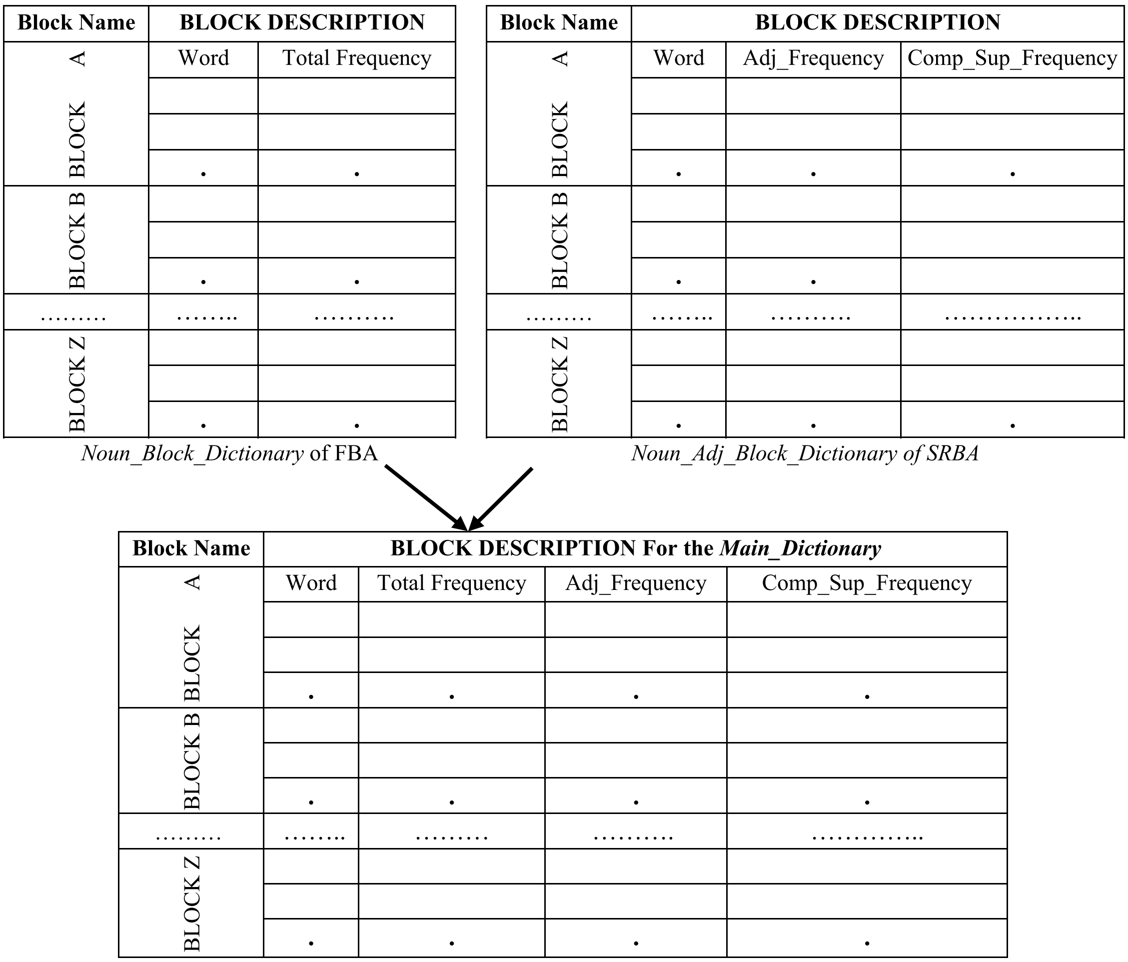

FIGURE 5. **Blocks representation that illustrates the creation of the *Main_Dictionary*.**

Restricting the aspects of only specific words and ignoring words that have similar meanings will negatively affect the performance of the ABSA. For example, in the *movie* domain, the word *story* is one of the words that occur in the *SharedWords* list but the words *article*, *narrative,* and *tale* do not appear in the *SharedWords* list despite the fact that they have a similar meaning to the word *story*. In order to handle the two issues, non-relevant words need to be removed and words that are semantically similar to the aspects need to be extracted as well.

The removal of non-relevant words is achieved through examining the similarity of the words with the domain using the pre-trained Google's Word2vec model [80]. Word2vec is one of the word embedding algorithms [80] that gives accurate values and can calculate the similarity values between words that are written incorrectly or misspelled which is the case of some words in the user reviews. If the words do not meet a specific threshold value, the words will be considered as not relevant. By performing this step, there is a higher chance that all of the words chosen as aspects are relevant to the domain. Finally, these aspects are stored in a *Main_Aspects* list.

The extraction of words that are strongly related (semantically) to the main aspects are concerned with the need to identify words that are low in frequency but still relevant to the domain. As the focus of this work is on ABSA, there is a strong desire to extract aspects that can enhance the performance of sentiment analysis. As such, words that appear with adjectives and adverbs are extracted from all the reviews texts because they meet more priorities to qualify as core terms. Earlier, the focus on the nouns was only in choosing the aspects. In this phase, we extract the adjective phrases and adverb phrases. The extraction of semantically similar words with the main aspects is achieved based on the following three steps:

Step 1: Extract all the Nouns that are Modified by Adjectives (NMAdj) in all of the adjective phrases mentioned in the user reviews and that have similarity values greater than a specific threshold to any of the aspects in the *Main_Aspects*. As the number of extracted nouns can be very large due to the large size of the datasets, only nouns with a frequency equal to and greater than 30 are extracted. The similarity of these nouns with the main aspects in the *Main_Aspects* list are measured using the Google pre-trained

Word2vec model. If the similarity value ≥ 0.5, the nouns are included as core terms.

Step 2: Extract all Words that are Modified by Adverbs (WMAdv) in all of the adverb phrases mentioned in the user reviews and that have similarity values greater than a specific threshold with any aspect of the *Main_Aspects*. The adverb phrases mostly contain adverbs which modify either verbs, adjectives or even adverbs. Similar to Step 1, the words that appear along with the adverbs and their frequencies in a count of at least 30 are selected. The similarity values of each aspect in the *Main_Aspects* list with the selected words are calculated. If the similarity value between the aspect and the word is greater than the threshold, then the word is selected as a core term.

Step 3: Merge the core terms gained as results from Step 1 and Step 2 and save the unique words in the *core_terms* list.

After the three tasks of the SEAE method are achieved and result in generating both the *Main_Aspects* list and the *core_terms* list, we apply a confirmation step. This step aims to prove whether the words generated for both aspects and core terms are comprehensive words that reflect the main words in all of the reviews. A summary field in the Amazon dataset is used for achieving this step, which is a very short text written by the user to show their opinion regarding the item. Figure 6 is a sample review from the dataset used to show the difference between the reviewText field that is used in the three previous tasks of the SEAE method and the summary field that is used in this confirmation step. Most of the available studies conducted in the past do not use the summary field and suffice to only use the reviewText field. This research is the first to use both fields for the aspect extraction process. To achieve this step, the same procedures of Step 1 are done, and all of the NMAdj in all of the adjective phrases that are mentioned in all the summary texts are extracted. All of the extracted nouns that have frequency values equal to or greater than 30 and have similarity values with any one of the main aspects greater than a specific threshold which are stored in a *Summary* list.

B. WEIGHTING OF ASPECTS

Aspect weight is a measure to evaluate the importance of aspects to users and a number of frequency-based weighting schemes have been proposed [81].

However, in this article, we focus on and evaluate three approaches which are based on the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme: the conventional TF-IDF, and two modified TF-IDF weighting schemes as proposed by Zhu *et al.* [23] and Ngoc *et al.* [2].

The following sub-sections include the explanation of each method.

1) TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY

TF-IDF is a weighting scheme intended to measure how important a word is to a specific document (in our case, the user review) in a collection (or corpus) of documents. It is


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns.  The music  is
at times hard to read because we think the book was published for
singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


FIGURE 6. **Sample of a review from the Amazon dataset.**

widely used in information retrieval and summarization. The equation of TF-IDF is as follows:

$$TF - IDF = TF\left(t, r\right).IDF\left(t\right) \tag{1}$$

where Term Frequency (TF) measures how frequently a term occurs in a review. It is the number of times a term t appears in a review r, divided by the total number of terms in the review r (equation 2). The Inverse Document Frequency (IDF) measures how important a term t is in all of the reviews (equation 3) where R is the total number of reviews and r_t is the number of reviews that contains the term t.

$$TF(t, r) = \frac{c(t_r)}{T_r} \tag{2}$$

$$IDF\left(t\right) = log(\frac{R}{r_t}) \tag{3}$$

2) MODIFIED TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY

We consider two versions of the modified TF-IDF, proposed by Zhu *et al.* [23] and Ngoc *et al.* [2]. The method proposed by Zhu *et al.* [23] adds an impact factor (IF) to the original equation based on the assumption that each word has different documents or different classes. The subsequent equation is as follows:

$$Modified - TFIDF\left(t, r\right) = TF - IDF\left(t, r\right).IF(t) \tag{4}$$

The calculation of TF and IDF is similar to the equations of (2) and (3), respectively. In our case, the calculation of the IF is based on the same ideas of Zhu *et al.* [23] with changes to the values of the parameters in order to suit our requirements. Thus, the equation for calculating the impact factor for the term t is as follows:

$$IF\left(t\right) = \begin{cases} X\left(t\right) + 3 & if\ t\ is\ the\ main\ aspect \\ X\left(t\right) + 1 & if\ t\ is\ the\ core\ terms \end{cases} \tag{5}$$

where:

$$X\left(t\right) = \sqrt{\frac{1}{C}\sum\nolimits_{i=1}^{C}\left(eq\left(t, C_i\right) - \frac{1}{C}\right)^2},$$

```
Function Aspect_SentimentWords_Pair is
  Input:
      #Read the Extracted Aspect
      Aspect_list ← Append Aspects to the list
      Review_text  #The user review
  Output: AspectSentimentWordsPair list

  Sentence_Tokenized ← Split the Review_text into Sentences
  For each Sentence in Sentence_Tokenized do
    NER_Array ← []
    IF (len (Sentence) > 1)
      Word_Tokenized ← Split the review_text into tokens using NLTK
       #Preprocessing:  Remove Numbers, Remove Special Characters, and Split Words from Special Characters then append the words
                                                   into Clean_Sentence
      Clean_Sentence ← Preprocessing (Word_Tokenized)
      Output ← Use 'depparse' and 'NER' annotators to annotate Clean_Sentence
        # NER is Name Entity Recognizer, depparser is Dependency Parser
        # Step 1: Collect the Words in the Sentence that has special NER
        For each NER in Output ['sentences'] ['tokens'] do
             IF NER in ['PERSON', 'MISC', 'TITLE', 'LOCATION']
                NER_Array. append (NER ['word'])
        # Step 2: Collect the Words' pair in the Sentence that has special Dependencies
        For each Relation in Output ['sentences'] ['enhancedDependencies'] do
            IF (Relation['dep'] in [ 'amod', 'advmod', 'nmod', 'agent', 'dobj', 'nsubj', 'nsubjpass', 'pobj', 'xcomp', 'neg', 'case',
                                                   'nmod:like', 'dep' ]
               Noun ← Relation (['governorGloss'])
               Adjective ← Relation ([' dependentGloss '])
                   Add the new pair (Noun + ' ' +Adjective) to the Phrase_list after checking the following conditions:
                 1. IF The pair is not existing before  → Add #Some pairs come from many relations
                   2. IF the last pair that was Added to the list has similar Noun or Adjective of the current pair → Merge
                                                        them
                  # ex. very good movie .it will be analyzed very movie and good movie; I want to merge → very good movie
         #Step3: Merge step 1 and 2 by adding indictors to the pair that has any of the specified NER & Remove StopWords
        For each pair in Phrase_list do
           Word_Tokenized ← Split the review_text into tokens using NLTK
            For each word in Word_Tokenized do
               IF word in NER_Array
                   Phrase_list1.append (NewPair, 1)
                   Break
            IF no word occurs in NER_Array
               Phrase_list1.append (NewPair, 0)
         #Step4 Arrange the pair in the Phrase_list1& Extract the pairs that contain aspects or contain any words from NER list
        For each Pair in Phrase_list1
          IF All the Current Pair Words exist in the Previous Pair Or vice versa
              Ignore the Pair with the small size
          Else
              IF the Pair does not contain any aspect from Aspect_list or Pair [1] equal to zero
                 Go to the Next Pair
              Else
                 AspectSentimentWordsPair.append (Pair)
```

FIGURE 7. Extracting the aspect sentiment words pair function.

C is the total number of class; and $eq(t, C_i)$ is the number of reviews exist in class C_i in which term t occurs, divided by the total number of reviews.

The reviews in the dataset used in our study are divided into five classes based on the value of ratings given to each review. In other words, since the ratings in the Amazon dataset are between 1 and 5, five classes corresponding to each rating are created.

The second variation of the TF-IDF is based on the work of Ngoc *et al.* [2], which is concerned with the reliability of the TF-IDF scheme being applied to short texts, such as reviews. It was found that TF–IDF performed well when it is used with long documents. However, this result does not apply for short texts [82]. As a result, inspired by the work of Ngoc *et al.* [2], the equation for TF-IDF is the same as in equation 1, with the only difference being calculating the *TF*. The *TF* is modified

```
Function Sentiment_Score is
 Input: Sentiment_Words_list
        #Read the developed domain-specific  lexicon
        Lexicon_Word_list ← Read the domain-specific  lexicon (Word, Sentiment Score) from file
        # Store Lexicon_Word_list as blocks (Blocking) < Block Name, Word, Sentiment Score >
        Lexicon_Dictionary ← Blocking (Lexicon_Word_list)
 Output: SentimentScore for the input list
 SentScore ← 0
 For each Word in Sentiment_Words_list do
     First_Letter ← Word [0]   #Determine the block for the Word
     Ascii_Letter ← Ascii (Capitalize (First_Letter)) #Capitalize convert letter from small to capital one
     Block_no ←Ascii_Letter-65
     wordscore ← 0,  WordIndex ← 0
     For each Word_in_Block in Lexicon_Dictionary do
         IF (Word = Word_in_Block)
             Wordscore = Lexicon_Dictionary [Block_no] ['SentimentScore'] [WordIndex]
             Break
         WordIndex ← WordIndex +1
     SentScore ← SentScore + Wordscore
```

FIGURE 8. Assigning sentiment score for words function.

as illustrated in the following equation:

$$TF(t, r) = \frac{c(t_r)}{T_r} \tag{6}$$

Here, the TF is the number of times the term t appears in the review's sentences, divided by the total number of sentences in the review.

3) ASPECT RATING INFERENCE

This task aims to calculate the rating of each extracted aspect, where the rating is assumed to reflect a user's opinion on the aspect of the item being reviewed. The inferring of aspects' ratings involved two steps: the first step is extracting the sentiment words for each aspect (i.e., aspect sentiment words pair), and the second step is assigning a sentiment score (i.e., a polarity) for the aspect's sentiment words. The syntactic dependency parser and the Named Entity Recognition (NER) of the Java StanfordCoreNLP library are used to extract the sentiment words of each aspect.

The algorithm for the extraction process is illustrated in Figure 7. The inputs for this function are the extracted aspects and their core terms and the review text. The function extracts all of the aspects' sentiment words pairs that are mentioned in the review text and stores them in a list. The difference between our extraction function and other researchers' is that we do not make any assumptions of the number of sentiment words for an aspect (i.e., many studies determine only one sentiment word for each aspect [21]) nor determine that a specific type of word is to be a sentiment word, as most of the past studies restrict the sentiment words to adjectives only.

To assign a score for the aspects' sentiment words, we use the *movie* domain-specific lexicon as discussed in our earlier work [24]. The generation of the lexicon is based on Labille *et al.*'s method in [5], but with different preprocessing approaches. The domain-specific lexicon outperformed the general-based lexicon in all of the experiments. This encouraged us to use it in this step to get an accurate score for each sentiment word. Figure 8 shows the assigning sentiment score function that calculates scores for the input words.

4) TOTAL REVIEW SCORE CALCULATION ALGORITHM

The conventional way for calculating the total review sentiment score is to extract all of the sentiment words that are mentioned in the review text and calculate their scores based on a specific lexicon. The summation of all the scores of the extracted sentiment words is the total review score. Another method to calculate the total review sentiment score is based on the aspect sentiment words pair, which is based on the hypothesis that a total review score is equal to the weighted sum of the user's opinion (i.e., their rating) on multiple aspects. Such a hypothesis has shown its reliability in many experiments such as [2], [4], [25]. The total review score based on the aspect sentiment words pair is as follows:

$$O_i = \sum\nolimits_{i=1}^{k} R_{ik} W_{ik} \tag{7}$$

where R_{ik} and W_{ik} refer to the rating and weight for aspect k in review i, respectively.

In this research, we propose an algorithm to implement the previous equation through using the results of the previous three tasks of our proposed approach as inputs to the algorithm (i.e., the extracted aspects with the core terms, aspects' weights, and the domain-specific lexicon for calculating the aspects' ratings). The output of this algorithm is the total review sentiment score (i.e., overall rating) for each review in the Testing Data using equation 7. The algorithm is explained step by step in Figure 9.

```
Algorithm ReviewScore_AspectBased is
Input:
    Aspect_Weight ← Read the aspects and their weights        #Read the Extracted Aspect with their weights
     # Store the aspects with their weights into blocks (Blocking) < Block Name, Aspect, Weight >
    Aspect_Weight_Dictionary ← Blocking (Aspect_Weight)
        # Read the Testing Data, each row consists of a Review Text
    TestData ← Read TestingData from file
Output: Sentiment Score for each review in TestData Based on Aspect and its Sentiment word pair

For each Row in TestData do
  Review_text ← Row
   #call the Aspect_SentimentWords_Pair function to extract the aspect sentiment words pair and store them in Phrase_list2
  Phrase_list2 ← Aspect_SentimentWords_Pair (Review_text)
   #calculate the pair score
  Final_Phrase_list ← 0
  For each Pair in Phrase_list2
        IF Pair Contain aspect
            Aspect ← add the aspect that exist in the pair
              #find the aspect weight from the Aspect_Weight_Dictionary
            Block_no ← Ascii (Capitalize (Aspect [0])) -65
            Aspectweight← 0 , WIndex ← 0
            For each Word_in_Block in Aspect_Weight_Dictionary [Block_no] do
                IF (Aspect = Word_in_Block)
                  Aspectweight= Aspect_Weight_Dictionary [Block_no] ['Weight'] [WIndex]
                  Break
                Windex ← Windex +1
        Else # the pair does not contain aspect but contain words from NER_Array
                Aspectweight← 1
        PairSentimentWord ← Add all the reminder pair words except the aspect into list
          # Call the Sentiment_Score Function to calculate the score for the pair sentiment words
        Score ← Sentiment_Score (PairSentimentWord)
        PairScore ← Score * Aspectweight
        Final_Phrase_list. append (Pair, PairScore)
      For each Pair in Final_Phrase_list   #calculate the total review sentiment score based on equation 7

        ReviewSentimentScore+ ← Pair [1] #summation of all pair scores
```

FIGURE 9. Total review sentiment score calculation algorithm.

IV. EXPERIMENT AND RESULTS

To implement the previous four tasks of our proposed approach, we use two standard datasets, the Amazon dataset[1] [83] and the Yelp dataset.[2] We implemented the SEAE task on the domains of *book*, *movie*, and *restaurant*. The datasets of the *book* and *movie* domains are from the Amazon dataset, whereas the *restaurant* domain is from the Yelp dataset. The number of user reviews used in the experiments are 1,500,000, 1,300,000, and 1,000,000 for the *book*, *movie*, and *restaurant* domains, respectively. The results for each task mentioned in the methodology section are presented and discussed in the following sections.

A. ASPECT EXTRACTION RESULTS

The results of the three SEAE tasks which were previously discussed in section III.A are presented in this section.

1) EXTRACTION OF NOUN AND NOUN-ADJECTIVE PAIRS

In the PHA approach, both FBA and SRBE were implemented in parallel, producing the *Noun_Block_Dictionary* and *Noun_Adj_Block_Dictionary*, respectively. Table 3 presents the PHA result for the *book*, *movie*, and *restaurant* domains for both FBA and SRBE. As seen, out of the 1,500,000 reviews in the *book* domain, only 1,499,845 are valid reviews and the remaining 155 are empty reviews. Similarly, for the *movie* domain, there are 50 empty reviews.

The unique nouns generated using the frequency-based approach (FBA) are 125,730, 139,877, and 80,160 for the *book*, *movie*, and *restaurant* domains, respectively. The syntactic-relation based approach (SRBA) produced fewer nouns as it mainly focused on nouns that come with the adjectives based on the adjectives' degree. Table 3 shows the number of all of the nouns extracted that appear with adjectives and the number of nouns that appear with comparative and superlative adjectives.

2) CREATION OF THE MAIN DICTIONARY

In this task, both two dictionaries resulted from FBA and SRBA are merged to create the *Main_Dictionary* using the steps explained in Figure 4. Each noun in the *Main_Dictionary* has three different frequency types

[1] http://jmcauley.ucsd.edu/data/amazon/links.html
[2] https://www.yelp.com/dataset

TABLE 3. Numbers of Noun Extracted using the PHA Approach for the Domains of Book, Movie and Restaurant.

Domain	Number of Reviews	Extracted Reviews		Nouns Extracted based on the Frequency Based Approach (FBA)	Size of FBA Dictionary	Nouns Extracted based on the Syntactic-Relation Based Approach (SRBA)		Size of SRBA Dictionary
		Read	Empty			All Nouns	Comparative/ Superlative Nouns	
Book	1,500,000	1,499,845	155	61,802,842	125,730	19,055,664	1,045,387	59,325
Movie	1,300,000	1,299,953	50	56,191,348	139,877	15,913,946	719,606	66,711
Restaurant	1,000,000	1,000,000	0	22,236,439	80,160	6,982,793	433,290	37,055

TABLE 4. Total Number of the Finallist List for the Domains of Book, Movie and Restaurant.

Domain	Book	Movie	Restaurant
Number of words	82	71	69

(i.e., total frequency, adjective frequency, and comparative/superlative frequency) as illustrated in Figure 5. The top 100 most frequent nouns of the three frequency types were merged to form a *SharedWords* list.

Finally, upon obtaining three *SharedWords* lists, each related to one of the three domains, we removed the common nouns (i.e., 15 words) that appear in the three *SharedWords* lists to keep the domain nouns only and store them in the *Finallist* list for each domain. Table 4 shows the total number of nouns in the *Finallist* list for the three domains.

3) GENERATION OF THE MAIN ASPECTS AND CORE TERMS

To achieve this task, two sub-processes were performed: removing words that are not relevant to the domain and extracting words semantically similar to the main aspects.

To remove non-relevant words related to the domain, words that occur in *Finallist* whose similarity values (using the pre-trained Google's Word2vec model) with the domain are less than a specific threshold are removed. The threshold values used are 0.16, 0.14, and 0.15 for the *book*, *movie*, and *restaurant* domains, respectively. Thus, only words that are strongly related to the domains and which are considered as the main aspects are retained in the *Finallist* and subsequently listed in the *Main_Aspects* list. Figure 10 lists the main aspects of each domain. To better clarify the aspects, the aspects are grouped into related topics.

The results have shown the capabilities of the proposed approach to extract aspects from user reviews for various domains. For extracting high similar words with the main aspects, we focus only on one domain, the *movie* domain. Given that multiple domains are required only in the previous processes to differentiate between the domain aspects and the common ones, as a result, in the remaining process, only the *movie* domain will be considered.

The results of the four steps of this process are explained in detail as follows:

TABLE 5. Results for the Adjective Phrases Extraction Step.

Range of Reviews	Total Reviews	No of NMAdj	Unique Nouns
0-100000	100,000	1,176,386	33,338
100000-200000	100,000	1,189,577	34,309
200000-300000	100,000	1,207,484	35,568
300000-600000	300,000	3,699,152	67,967
600000-950000	350,000	4,530,559	77,908
950000-1300000	350,000	4,110,788	72,407
Total	**1,300,000**	**15,913,946**	**159,446**

TABLE 6. Results for the Adjective Phrases Filtering Process.

Total Number of Nouns	159,446
Nouns with Frequency >=30	16,306
Nouns not in the Word2vec vocabulary	2,509
Nouns in the Word2vec vocabulary	13,797

Step 1: Extract all of the nouns that are modified by adjectives (NMAdj) from the user reviews. To expedite the process, the dataset is divided into six parts, where all of the adjective phrases are extracted and the nouns that are being modified by adjectives are merged. The number of extracted nouns is 159,446 as shown in Table 5. Of the 159,446 nouns, only nouns with frequency ≥ 30 are used.

These nouns are then further filtered by removing those that do not exist in the Word2vec vocabulary. Such nouns are mostly those with incorrect spelling or proper nouns. The results of the extraction and filtering are summarized in Table 6.

Finally, the similarity values of the main aspects stored in the *Main_Aspects* list with the 13,797 nouns are calculated. Nouns that have similarity values greater than 0.5 are considered to be core terms. The total number of core terms is 488.

Step 2: Extract all of the words that are modified by adverbs (WMAdv) appearing in the adverb phrases of the user reviews. This process is similar to Step 1, but the focus is on adverb phrases instead of adjective phrases. Table 7 shows the result of Step 2.

Similar to the previous step, only words with a frequency greater than or equal to 30 were used and then further filtered

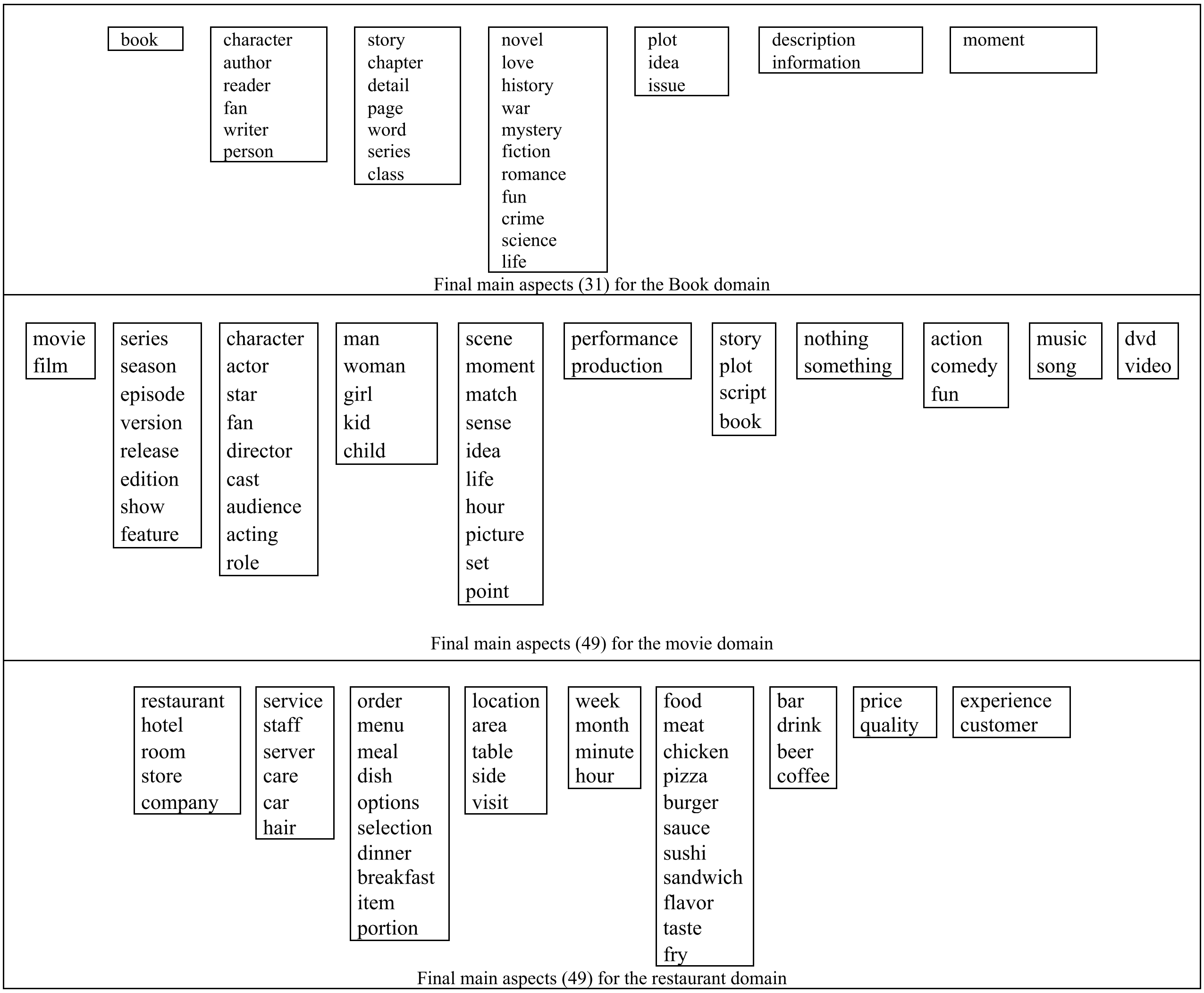

FIGURE 10. The final main aspects of the three domains.

TABLE 7. Results for the Adverb Phrases Extraction Step.

Range of Reviews	Total Reviews	Number of WMAdv	Unique Words
0-300000	300,000	2,803,213	52,397
300000-350000	50,000	459,044	23,828
350000-650000	300,000	3,075,926	55,795
650000-700000	50,000	538,906	25,089
700000-1000000	300,000	3,083,689	56,783
100000-1300000	300,000	2,857,544	51,864
Total	**1,300,000**	**12,818,322**	**112,715**

by removing words that do not exist in the Word2vec vocabulary. The result of this is shown in Table 8.

Finally, the similarity values of the main aspects stored in the *Main_Aspects* list with the 14,943 words are calculated. The number of the extracted core terms where the similarity values with the main aspects are at least 0.5 is 380.

TABLE 8. Results for the Adverb Phrases Filtering Process.

Total Number of WMAdv	112715
WMAdv with Frequency >=30	15,424
WMAdv not in the Word2vec vocabulary	481
WMAdv in Word2vec vocabulary	14,943

Step 3: Extract all of the nouns that are modified by adjectives (NMAdj) from the summary field. The dataset is divided into three parts, all of the adjective phrases are extracted from each part, and then all of the extracted nouns are merged. After this, the redundant nouns are removed, and finally, we get 20,631 unique nouns from this extraction step as depicted in Table 9.

Similar to the previous two steps, the extracted nouns are filtered as shown in Table 10, and the similarity values of the main aspects that stored in the *Main_Aspects* list with the 1,793 nouns are calculated. This produces 208 words that are considered to be core terms and have similarity values with the domain of greater than 0.5.

TABLE 9. Results of the Adjective Modifiers Extraction Step from the Summary Field.

Range of Reviews	No of Reviews	Empty Reviews	No of NMAdj	Unique Nouns
0-500000	500,000	8	291,329	12,035
500000-1000000	500,000	5	285,611	12,802
1000000-1300000	300,000	2	168,139	9,437
Total	**1,300,000**	**15**	**745,079**	**20,631**

TABLE 10. Results of the Adjective Modifiers Filtering Process from the Summary Field.

Total Number of Nouns	20,631
Nouns with Frequency >=30	1,825
Nouns not in Word2vec vocabulary	32
Nouns in Word2vec vocabulary	1,793

We can notice that, in the summary field, the total number of unique nouns (i.e., nouns in Word2vec vocabulary that are candidates to be core terms) is very small (1,793) as compared to the number of the resulted words after Step 1 (13,797) and Step 2 (14,943).

On the other hand, the total number of nouns that have high similarity values with the main aspects that are stored in the *Main_Aspects* list and chosen as core terms in the summary field is 208.

This number is considered a big number whereas the number of words in both Steps 1 and 2 are 488 and 380 words, respectively. Additionally, we can conclude that this big number indicates that the summary field contains valuable words that are related to the aspects. Thus, it supports our decision to select the summary field in the aspect extraction process.

Step 4: The last step of this task is merging the results of both Steps 1 and 2. After removing redundant words, the *core_terms* list now contains 481 words. We noticed that all of the 208 words that resulted from Step 3 exist in the *core_terms* list, and this proves that the extracted words are comprehensive words (i.e., main words) extracted from all of the reviews. The effect of these extracted words in the sentiment analysis will be studied in the next section to prove whether the extracted aspects have a positive effect on the sentiment analysis process.

Table 11 presents samples of the extracted core terms with their similarity values with the main aspects. As seen in Table 11, the words *edition* and *performance* were wrongly spelled (*editon*, *peformance,* and *performace)*, but have high similarity values with the main aspect. This is one of the reasons for choosing the Word2vec model. Ignoring all of the words that are similar to the main aspects but incorrectly written by the users will affect the ABSA process negatively, but by using the Word2vec model, this problem can be eliminated because it has the ability to calculate the similarity values for such words.

TABLE 11. Samples of similarity values between the extracted core terms and main aspects.

Core Terms	Main Aspects	Similarity Value
actress	**actor**	0.7930
comedic	**comedy**	0.7498
lyric	**song**	0.7426
superstar	**star**	0.7341
coordinator	**director**	0.7340
editon	**edition**	0.7332
peformance	**performance**	0.7114
performace	**performance**	0.6889
tale	**story**	0.6853
jazz	**music**	0.6835
concept	**idea**	0.6744
cd	**dvd**	0.6546
paperback	**book**	0.6471
photograph	**picture**	0.6404
documentary	**film**	0.6345
remix	**song**	0.6332
biography	**book**	0.6156
soundtrack	**song**	0.6155
costar	**actor**	0.5978

B. TOTAL REVIEW SCORE CALCULATION ALGORITHM

In this part, the results for the remaining tasks of our proposed approach will be presented. The dataset that will be used in this section is the Amazon dataset. Each review has an overall rating range from 1 to 5 (i.e., ratings 1 and 2 as negative, 3 as neutral, and 4 and 5 positive) [84]. We use the positive and negative reviews only, and ignore the neutral reviews as Labille *et al.*'s method used in [5]. As a result, the new size for the dataset is 1,196,941 reviews, of which 1,035,299 are positive reviews and 161,642 are negative reviews. The dataset is divided into 80% to build the lexicon and 20% to test our algorithm (i.e., calculating the total review score based on the extracted aspects using the SEAE approach).

The algorithm requires three inputs: the aspects and core terms, the aspects' weights, and the aspects' ratings. First, for the *movie* domain-specific lexicon, we use the *movie* domain-specific lexicon in [24], consisting of 123,178 sentiment words, whereby 117,657 of them are positive and the remaining are negative. This lexicon is built for an unbalanced big-sized dataset and proves its efficiency in the sentiment analysis process.

Second, to choose the efficient aspect weight's method among the three methods mentioned in section III.B, we apply the *ReviewScore_AspectBased* algorithm that is explained in Figure 9 for each of the three aspect weight's methods, separately. The *ReviewScore_AspectBased* algorithm takes the reviews' texts as input and generates the total review sentiment score for each review based on equation 7. The resultant total review sentiment score is a value ranging between -1 and 1. To evaluate the performance of our algorithm, we compare the resultant score with the overall rating value that comes with the review in the Amazon dataset.

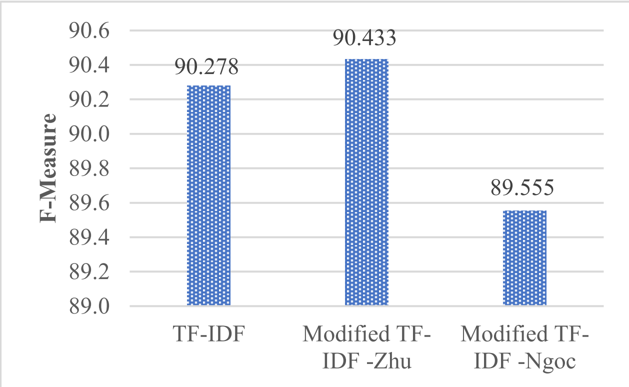

FIGURE 11. The F-measure of the three methods of aspects' weighting in calculating total review score.

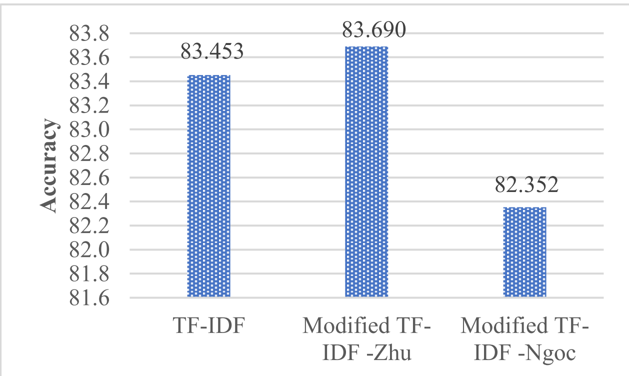

FIGURE 12. The accuracy of the three methods of aspects' weighting in calculating total review score.

If the overall rating is 4 or 5 and the resulting sentiment score is positive, it is considered as a correct calculation for the total review sentiment score, i.e., True Positive (TP). However, if the resultant sentiment score is negative, it is considered as a False Positive (FP). Similar classification is applied for the True Negative (TN) and False Negative (FN) for the negative scores. Two standard performance measures are used, the F-measure and the accuracy, which the equations depict are as follows:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (11)$$

The following sub-sections describe the results in detail

1) WEIGHTING OF ASPECTS

The results of applying the three aspects' weights methods in the proposed algorithm explained in Figure 9 are as shown in Figure 11 for the F-measure, and in Figure 12 for the accuracy measure.

TABLE 12. Details of the Generated Lexicons.

Lexicon	Threshold (O)	Lexicon Details		
		Total Size	Positive Words	Negative Words
Adj-Lexicon(5)	5	46,825	45,682	1,143
Adj-Lexicon(1)	1	205,615	202,795	2,820
Lexicon(10)	10	123,178	117,657	55,21

It is clear that the Modified TF-IDF proposed by Zhu *et al.* [23] generates the best F-measure and accuracy compared to the other two methods. As a result, it will be chosen to estimate the aspects' weights.

2) TYPE OF WORDS IN LEXICON

The lexicon that was built in our previous research [24] consists of words of all types (i.e., noun, verb, adjective, and adverb) and it proves its efficiency in the sentiment analysis process and outperforms the performance of the lexicon that contains only adjectives. In this part, we also assess whether the performance of the lexicon that contains all types of words outperforms the performance of the lexicon that contains only adjectives in our algorithm described in Figure 9. We build three lexicons using the same method explained in [24], the exception being choosing the words' type during the building of the lexicon, if we choose all of the words' types or the adjectives only. The number of the processed reviews is 1,196,941, of which 1,035,299 are positive reviews and 161,642 are negative reviews. The following table shows the detail of the generated lexicons.

Table 12 presents the details of the three generated lexicons and illustrates the size of the generated lexicon and the total number of both positive and negative words. The Adj-Lexicon(5) and Adj-Lexicon(1) are adjective lexicons with different thresholds whereas Lexicon(10) contains all of the types of words (i.e., noun, verb, adjective, and adverb). The chosen words for building the lexicon are selected based on a specific threshold (O). The O threshold refers to the minimum number of reviews in which the word is mentioned. For example, in the Adj-Lexicon(5), the threshold value is 5, which means that only the words that are mentioned in at least five reviews are appending to the lexicon.

We implemented our algorithm by using the extracted aspects with the core terms, the modified TF-IDF proposed by Zhu *et al.* [23] for aspects' weights, and the three generated lexicons presented in Table 12 to select the suitable lexicon. The results are shown in Figure 13.

We can observe that Lexicon(10) outperforms the other two lexicons. This proves that all of the words' types, and not only adjectives, can be sentiment words. In the end, we can conclude that the Modified TF-IDF proposed by Zhu *et al.* [23] and Lexicon(10) are the suitable aspect's weight method and the efficient lexicon for aspects' ratings, respectively.

In the next section, the extracted elements, i.e., the extracted aspects and the *movie* domain-specific Lexicon(10), are evaluated with other baselines.

TABLE 13. Details of the Four Test Cases.

Case Number	Aspect Type	Algorithm	Aspect's Rating (Lexicon Type)	Aspect's Weight
Case #1 (Fixed-General)	fixed aspects given by [41]	algorithm proposed by [41] to determine the sentiment words for each aspect	general-based lexicon (SentiWordNet [85])	Modified TF-IDF proposed by Zhu [23]
Case #2 (SABRE-General)	extracted aspects using Caputo et al.'s method [44]			
Case #3 (SEAE-General)	the proposed extracted aspects	review_score aspect_based algorithm which proposed in this research		
Case #4 (SEAE-Domain)			movie domain-specific Lexicon(10)	

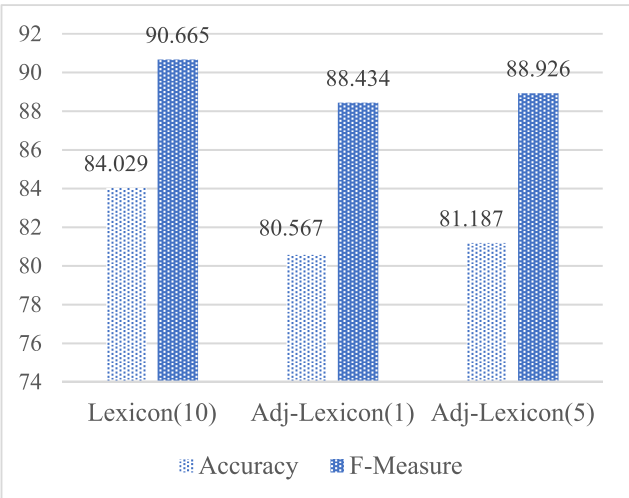

FIGURE 13. The accuracy and F-measures of the proposed approach on the three lexicons.

V. EVALUATION

We compare our extracted aspects with two aspects' types, the first is the fixed aspects given by Hernández-Rubio *et al.* [41] for the Amazon *movie* domain, where the total number of the main fixed aspects is 23 and the total number of the core terms is 271. The second compared aspect's type is the extracted aspects using the proposed method by Caputo *et al.* [44], where the total number of the extracted aspects for the *movie* domain of the Amazon dataset is 223 words.

It is important to note that the aspect extraction process is done for two main purposes: for sentiment analysis and for summarization. In this research, we focus on conducting sentiment analysis. As a result, we will check the efficiency of the extracted aspects in calculating the total review sentiment score based on aspects as stated in equation 7. Four different cases will be tested to evaluate their performances in calculating the total review sentiment score based on the aspects with their weights and ratings. The detail for each element (i.e., aspect type, aspect's weight, aspect's rating, and algorithm for extracting the aspect's sentiment words) in the four tested cases is explained in Table 13.

There are two points must be clarified regarding Table 13:

- The algorithm proposed by Hernández-Rubio *et al.* [41] for extracting the aspect's sentiment words is quite similar to our algorithm, but they do not use NER and the number of our chosen modifiers is more than that used in their algorithm.
- The only difference in case three and four is the lexicon's type. The aim of this difference is to evaluate the efficiency of the extract aspects without affecting the domain lexicon.

We apply 5-fold cross validation for the *movie* domain of the Amazon dataset and in each fold, the dataset is divided into 80% for building the lexicon in case we need it as in case 4, and the remaining 20% is used for calculating the reviews' total sentiment scores in the four cases.

Three performance measures are tested: the F-measure, accuracy, and coverage. The coverage performance measure represents how many times our algorithm does not fail to assign a score for a review, in other words, how many reviews resulted with total sentiment scores not equal to zero, or with a positive or negative score. This measure shows the coverage of the extracted aspects, in other words, whether the extracted aspects are comprehensive for all the reviews and efficient in the sentiment analysis process.

The average results of the 5-folds of the four tested cases are shown in Figure 14. It is clear from the figure that both SEAE-General and SEAE-Domain outperformed the fixed aspects and the extracted aspects observed by Caputo *et al.* [44] in the three performance measures. Also, our extracted aspects work with the domain-specific Lexicon(10) and performed better than the general-based lexicon, which proves the efficiency of the Lexicon(10) and shows that it is more suitable than the general lexicon in the sentiment analysis process.

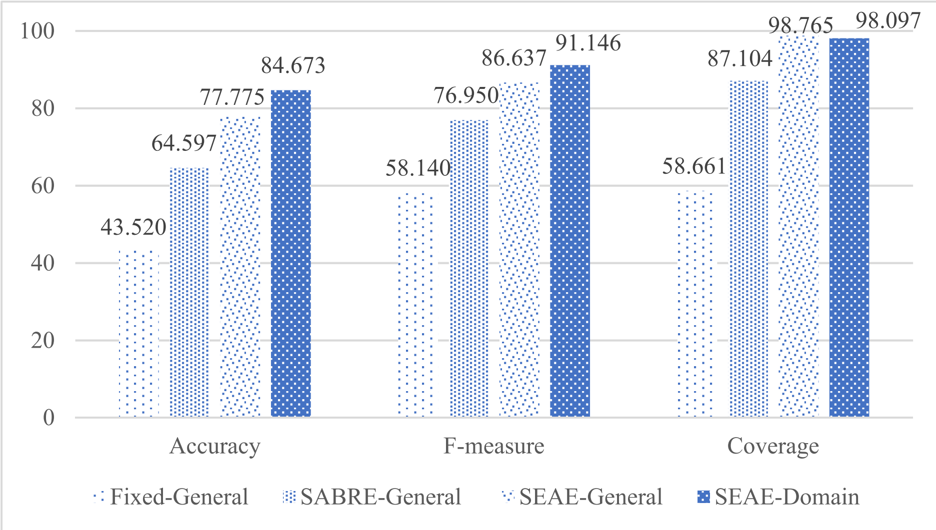

FIGURE 14. Results of the four test cases in terms of average accuracy, F-measure and coverage.

Additionally, SEAE-General outperforms both Fixed-General and SABRE-General despite that the three cases using the same aspects' weights method and the same lexicon. This proves that the reason behind the high performance of SEAE-General is our extracted aspects that work efficiently in the sentiment analysis process.

Finally, the coverage performance measure of SEAE-General outperformed SEAE-Domain with a small difference. The reason behind this is that the number of the words in the SentiWordNet is bigger than the built domain-specific lexicon (i.e., SentiWordNet contains more than 200,000 entries depicting sentiment scores for various senses of words and phrases [85]). On the other hand, the accuracy and the F-measure of SEAE-Domain are 7% and 4.5% higher than SEAE-General, respectively. This means that big-sized lexicons do not always reflect high performance.

VI. CONCLUSION

In conclusion, this article described an efficient approach for aspect-based sentiment analysis process designed for unbalanced large-scale reviews and implemented on a real Amazon dataset. The approach consists of four various tasks, the first task is extracting the main aspects from three domains, *movies*, *books*, and *restaurants*, through a hybrid approach. This was followed by a semantic similarity-based approach to extract the core terms for each main aspect. The second task is estimating a weight for each aspect that reflects the importance of the aspect for users within all the reviews by using a modified version of TF-IDF, followed by the third task that aims to assign a rating for the extracted aspect using a domain-specific lexicon. The last task uses the aspects and their weights with the support of the domain lexicon as inputs for an algorithm that is specialized in calculating the total review sentiment score based on the aspects. The proposed approach is evaluated by comparing the generated total review sentiment scores with other three cases using fixed aspects and learned aspects with either a domain-specific or general-based lexicon. The proposed approach outperformed the three compared cases in terms of F-measure, accuracy, and coverage. In the near future, we plan to increase the efficiency of the proposed approach by using the co-occurrence relations during the extraction of the core terms.

While the co-occurrence relations can predict coherent knowledge among words, our approach does not use such relations during the extraction of the core terms for each aspect and depends only on the semantic and dependency relations. Thus, some terms that co-occur with some opinion words may remain unextracted. This issue will be put forward as a future work.

Additionally, we plan to use the extracted aspects in profiling user and item profiles for applications of recommender system.

ACKNOWLEDGMENT

The authors acknowledge the support of the Organization for Women in Science for the Developing World (OWSD) and Swedish International Development Cooperation Agency (Sida).

REFERENCES

[1] S. M. Al-Ghuribi and S. A. Mohd Noah, "Multi-criteria review-based recommender system–The state of the art," *IEEE Access*, vol. 7, pp. 169446–169468, 2019.

[2] T. N. T. Ngoc, H. N. T. Thu, and V. A. Nguyen, "Mining aspects of customer's review on the social network," *J. Big Data*, vol. 6, no. 1, p. 22, 2019.

[3] S. Moghaddam and M. Ester, "Opinion digger: An unsupervised opinion miner from unstructured product reviews," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1825–1828.

[4] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: A rating regression approach," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 783–792.

[5] K. Labille, S. Gauch, and S. Alfarhood, "Creating domain-specific sentiment lexicons via text mining," in *Proc. Workshop Issues Sentiment Discovery Opinion Mining (WISDOM)*, 2017, pp. 1–8.
[6] S. Mukherjee and P. Bhattacharyya, "Feature specific sentiment analysis for product reviews," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*, 2012, pp. 475–487.
[7] C. Musto, G. Rossiello, M. de Gemmis, P. Lops, and G. Semeraro, "Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations," in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 383–387.
[8] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 168–177.
[9] Y. Xia, E. Cambria, A. Hussain, and H. Zhao, "Word polarity disambiguation using Bayesian model and opinion-level features," *Cognit. Comput.*, vol. 7, no. 3, pp. 369–380, Jun. 2015.
[10] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.
[11] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," 2016, *arXiv:1610.08815*. [Online]. Available: http://arxiv.org/abs/1610.08815
[12] M. Eirinaki, S. Pisal, and J. Singh, "Feature-based opinion mining and ranking," *J. Comput. Syst. Sci.*, vol. 78, no. 4, pp. 1175–1184, Jul. 2012.
[13] H. H. Do, P. Prasad, A. Maag, and A. Alsadoon, "Deep learning for aspect-based sentiment analysis: A comparative review," *Expert Syst. Appl.*, vol. 118, pp. 272–299, Mar. 2019.
[14] N. A. Osman, "Contextual sentiment based recommender system to provide recommendation in the electronic products domain," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 4, pp. 425–431, Aug. 2019.
[15] F. N. Al-Aswadi, H. Y. Chan, and K. H. Gan, "Automatic ontology construction from text: A review from shallow to deep learning trend," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 3901–3928, Aug. 2020.
[16] A. R. Alaei, S. Becken, and B. Stantic, "Sentiment analysis in tourism: Capitalizing on big data," *J. Travel Res.*, vol. 58, no. 2, pp. 175–191, Feb. 2019.
[17] Z. Kastrati, A. S. Imran, and A. Kurti, "Weakly supervised framework for aspect-based sentiment analysis on students' reviews of MOOCs," *IEEE Access*, vol. 8, pp. 106799–106810, 2020.
[18] F. Ali, D. Kwak, P. Khan, S. El-Sappagh, A. Ali, S. Ullah, K. H. Kim, and K.-S. Kwak, "Transportation sentiment analysis using word embedding and ontology-based topic modeling," *Knowl.-Based Syst.*, vol. 174, pp. 27–42, Jun. 2019.
[19] L. Sun, J. Guo, and Y. Zhu, "Applying uncertainty theory into the restaurant recommender system based on sentiment analysis of online Chinese reviews," in *Proc. World Wide Web*, 2018, pp. 1–18.
[20] M. S. Mubarok, Adiwijaya, and M. D. Aldhi, "Aspect-based sentiment analysis to review products using Naïve Bayes," in *Proc. AIP Conf.*, 2017, Art. no. 020060.
[21] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.
[22] Z. Madhoushi, A. R. Hamdan, and S. Zainudin, "Sentiment analysis techniques in recent works," in *Proc. Sci. Inf. Conf. (SAI)*, Jul. 2015, pp. 288–291.
[23] L. Zhu, G. Wang, and X. Zou, "A study of Chinese document representation and classification with Word2vec," in *Proc. 9th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2016, pp. 298–302.
[24] S. M. Al-Ghuribi, S. A. Noah, and S. Tiun, "Various pre-processing strategies for domain-based sentiment analysis of unbalanced large-scale reviews," in *Proc. 6th Int. Conf. Adv. Intell. Syst. Inform. (AISI)*, Cairo, Egypt, 2020, pp. 204–214.
[25] J. Yu, Z.-J. Zha, M. Wang, and T.-S. Chua, "Aspect ranking: Identifying important product aspects from online consumer reviews," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2011, pp. 1496–1505.
[26] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Trans. Affect. Comput.*, early access, Jan. 30, 2020, doi: 10.1109/TAFFC.2020.2970399.
[27] A. Hussain and E. Cambria, "Semi-supervised learning for big social data analysis," *Neurocomputing*, vol. 275, pp. 1662–1673, Jan. 2018.
[28] M. Tubishat, N. Idris, and M. Abushariah, "Explicit aspects extraction in sentiment analysis using optimal rules combination," *Future Gener. Comput. Syst.*, vol. 114, pp. 448–480, Jan. 2021.
[29] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016.
[30] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, New Orleans, LA, USA, 2018, pp. 5876–5883.
[31] A. Da'u, N. Salim, I. Rabiu, and A. Osman, "Weighted aspect-based opinion mining using deep learning for recommender system," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112871.
[32] G. Ansari, C. Saxena, T. Ahmad, and M. N. Doja, "Aspect term extraction using graph-based semi-supervised learning," *Procedia Comput. Sci.*, vol. 167, pp. 2080–2090, Jan. 2020.
[33] D. Anand and D. Naorem, "Semi-supervised aspect based sentiment analysis for movies using review filtering," *Procedia Comput. Sci.*, vol. 84, pp. 86–93, Jan. 2016.
[34] N. Li, C.-Y. Chow, and J.-D. Zhang, "SEML: A semi-supervised multi-task learning framework for aspect-based sentiment analysis," *IEEE Access*, vol. 8, pp. 189287–189297, 2020.
[35] Z. Yan, M. Xing, D. Zhang, and B. Ma, "EXPRS: An extended pagerank method for product feature extraction from online consumer reviews," *Inf. Manage.*, vol. 52, no. 7, pp. 850–858, Nov. 2015.
[36] A. García-Pablos, M. Cuadros, and G. Rigau, "W2 VLDA: Almost unsupervised system for aspect based sentiment analysis," *Expert Syst. Appl.*, vol. 91, pp. 127–137, Jan. 2018.
[37] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, "Sentence compression for aspect-based sentiment analysis," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2111–2124, Dec. 2015.
[38] M. Shams, N. Khoshavi, and A. Baraani-Dastjerdi, "LISA: Language-independent method for aspect-based sentiment analysis," *IEEE Access*, vol. 8, pp. 31034–31044, 2020.
[39] M. Darwich, S. A. M. Noah, and N. Omar, "Deriving the sentiment polarity of term senses using dual-step context-aware in-gloss matching," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102273.
[40] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proc. 55th Annu. Meeting Assoc. for Comput. Linguistics, Long Papers*, vol. 1, 2017, pp. 388–397.
[41] M. Hernández-Rubio, I. Cantador, and A. Bellogín, "A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews," *User Model. User-Adapted Interact.*, vol. 29, no. 2, pp. 381–441, Apr. 2019.
[42] S. Aciar, D. Zhang, S. Simoff, and J. Debenham, "Informed recommender: Basing recommendations on consumer product reviews," *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 39–47, May 2007.
[43] M. Siering, A. V. Deokar, and C. Janze, "Disentangling consumer recommendations: Explaining and predicting airline recommendations based on online reviews," *Decis. Support Syst.*, vol. 107, pp. 52–63, Mar. 2018.
[44] A. Caputo, P. Basile, M. de Gemmis, P. Lops, G. Semeraro, and G. Rossiello, "SABRE: A sentiment aspect-based retrieval engine," in *Information Filtering and Retrieval*. Cham, Switzerland: Springer, 2017, pp. 63–78.
[45] Z. M. Zohreh Madhoushi, A. R. Hamdan, and S. Zainudin, "Aspect-based sentiment analysis methods in recent years," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 8, no. 1, pp. 79–96, Jun. 2019.
[46] M. E. Mowlaei, M. Saniee Abadeh, and H. Keshavarz, "Aspect-based sentiment analysis using adaptive aspect-based lexicons," *Expert Syst. Appl.*, vol. 148, Jun. 2020, Art. no. 113234.
[47] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, Mar. 2011.
[48] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proc. 2nd Workshop Natural Lang. Process. Social Media (SocialNLP)*, 2014, pp. 28–37.
[49] Z. Luo, S. Huang, and K. Q. Zhu, "Knowledge empowered prominent aspect extraction from product reviews," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 408–423, May 2019.
[50] Z. Hai, K. Chang, J.-J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 623–634, Mar. 2014.
[51] M. Chen and T. Yao, "Combining dependency parsing with shallow semantic analysis for Chinese opinion-element relation identification," in *Proc. 4th Int. Universal Commun. Symp. (IUCS)*, Oct. 2010, pp. 299–305.

[52] M. Dragoni, M. Federici, and A. Rexha, "An unsupervised aspect extraction strategy for monitoring real-time reviews stream," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 1103–1118, May 2019.

[53] S. Jamshidi Nejad, F. Ahmadi-Abkenari, and P. Bayat, "A combination of frequent pattern mining and graph traversal approaches for aspect elicitation in customer reviews," *IEEE Access*, vol. 8, pp. 151908–151925, 2020.

[54] H. Lakkaraju, C. Bhattacharyya, I. Bhattacharya, and S. Merugu, "Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2011, pp. 498–509.

[55] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[56] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1, pp. 177–196, Jan. 2001.

[57] Y. Seroussi, F. Bohnert, and I. Zukerman, "Personalised rating prediction for new users using latent factor models," in *Proc. 22nd ACM Conf. Hypertext Hypermedia*, 2011, pp. 47–56.

[58] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 639–648.

[59] S. Brody and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2010, pp. 804–812.

[60] S. Moghaddam and M. Ester, "ILDA: Interdependent LDA model for learning latent aspects and their ratings from online product reviews," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2011, pp. 665–674.

[61] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 375–384.

[62] J. McAuley, J. Leskovec, and D. Jurafsky, "Learning attitudes and attributes from multi-aspect reviews," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 1020–1025.

[63] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, "Deep learning for aspect-level sentiment classification: Survey, vision, and challenges," *IEEE Access*, vol. 7, pp. 78454–78483, 2019.

[64] O. Alqaryouti, N. Siyam, A. A. Monem, and K. Shaalan, "Aspect based sentiment analysis using smart government review data," *Appl. Comput. Inform.*, 2019, doi: 10.1016/j.aci.2019.11.003.

[65] R. Kumar, H. S. Pannu, and A. K. Malhi, "Aspect-based sentiment analysis using deep networks and stochastic optimization," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3221–3235, Apr. 2020.

[66] E. Ekinci and S. İ. Omurca, "Concept-LDA: Incorporating babelfy into LDA for aspect extraction," *J. Inf. Sci.*, vol. 46, no. 3, pp. 406–418, Jun. 2020.

[67] G. S. Chauhan, Y. K. Meena, D. Gopalani, and R. Nahta, "A two-step hybrid unsupervised model with attention mechanism for aspect extraction," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113673.

[68] T. Liang, W. Wang, and F. Lv, "Weakly-supervised domain adaptation for aspect extraction via multi-level interaction transfer," 2020, *arXiv:2006.09235*. [Online]. Available: http://arxiv.org/abs/2006.09235

[69] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. de Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. María Jiménez-Zafra, and G. Eryigit, "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 1–13.

[70] H. Gandhi and V. Attar, "Extracting aspect terms using CRF and bi-LSTM models," *Procedia Comput. Sci.*, vol. 167, pp. 2486–2495, Jan. 2020.

[71] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 486–495.

[72] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis," *Cognit. Comput.*, vol. 10, no. 4, pp. 639–650, Aug. 2018.

[73] T. A. Rana and Y.-N. Cheah, "Aspect extraction in sentiment analysis: Comparative analysis and survey," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 459–483, Dec. 2016.

[74] A. Bagheri, M. Saraee, and F. de Jong, "An unsupervised aspect detection model for sentiment analysis of reviews," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.*, 2013, pp. 140–151.

[75] Y. Li, Z. Qin, W. Xu, and J. Guo, "A holistic model of mining product aspects and associated sentiments from online reviews," *Multimedia Tools Appl.*, vol. 74, no. 23, pp. 10177–10194, Dec. 2015.

[76] C. Long, J. Zhang, and X. Zhu, "A review selection approach for accurate feature rating estimation," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 766–774.

[77] T. A. Rana and Y.-N. Cheah, "Improving aspect extraction using aspect frequency and semantic similarity-based approach for aspect-based sentiment analysis," in *Proc. Int. Conf. Comput. Inf. Technol.*, 2017, pp. 317–326.

[78] K. Thabit and S. M. Al-Ghuribi, "A new search algorithm for documents using blocks and words prefixes," *Sci. Res. Essays*, vol. 8, no. 16, pp. 640–648, Apr. 2013.

[79] E. Klein, "A semantics for positive and comparative adjectives," *Linguistics Philosophy*, vol. 4, no. 1, pp. 1–45, Mar. 1980.

[80] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[81] G. Domeniconi, G. Moro, R. Pasolini, and C. Sartori, "A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of TF. IDF," in *Proc. Int. Conf. Data Manage. Technol. Appl.*, 2015, pp. 39–58.

[82] I. Alsmadi and G. K. Hoon, "Term weighting scheme for short-text classification: Twitter corpuses," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3819–3831, Aug. 2019.

[83] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 507–517.

[84] K. Labille, S. Alfarhood, and S. Gauch, "Estimating sentiment via probability and information theory," in *Proc. 8th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, 2016, pp. 121–129.

[85] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. Lrec*, 2010, pp. 2200–2204.

SUMAIA MOHAMMED AL-GHURIBI was born in Taiz, Yemen, in 1986. She received the B.Sc. degree (Hons.) in computer science from Taiz University, Yemen, in 2008, and the M.S. degree in computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 2014. She is currently pursuing the Ph.D. degree with Universiti Kebangsaan Malaysia (UKM), Malaysia. Her research interests include artificial intelligence, web mining, sentiment analysis, and recommender systems.

SHAHRUL AZMAN MOHD NOAH (Member, IEEE) received the B.Sc. degree (Hons.) in mathematics from Universiti Kebangsaan Malaysia (UKM), Malaysia, in 1992, and the M.Sc. and Ph.D. degrees in information studies from the University of Sheffield, U.K., in 1994 and 1998, respectively. He is currently a Professor with the Center for Artificial Intelligence Technology, UKM, where he is also the Head of the Knowledge Technology Research Group. His research interests include information retrieval and ontology with a special emphasis on semantic search and recommender systems. He has published more than 200 research articles in these areas. He is a member of the International Association for Ontology and its Applications (IAOA) and the IEEE Computer Society. He is also the Chair of the Persatuan Capaian Maklumat dan Pengurusan Pengetahuan (PECAMP).

SABRINA TIUN received the B.Sc. degree (Hons.) from Bradley University, USA, and the master's and Ph.D. degrees from University Sains Malaysia. She is currently a Senior Lecturer with the Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM), Malaysia. She is also the Head of the Asian Language Processing (ASLAN) Laboratory. Her research interests include natural language processing, computational linguistic, speech processing, and social science computing. She teaches several courses in the field of natural language processing and data science at the undergraduate and postgraduate levels. She also actively supervises postgraduate students for master and Ph.D. level studies, especially in natural language processing and speech processing.