Accepted Manuscript

Sentiment analysis: An automatic contextual analysis and ensemble clustering approach and comparison

Murtadha Talib AL-Sharuee, Fei Liu, Mahardhika Pratama

PII: S0169-023X(17)30186-6

DOI: 10.1016/j.datak.2018.04.001

Reference: DATAK 1640

To appear in: *Data & Knowledge Engineering*

Received Date: 10 April 2017

Revised Date: 26 March 2018

Accepted Date: 3 April 2018

Please cite this article as: M.T. AL-Sharuee, F. Liu, M. Pratama, Sentiment analysis: An automatic contextual analysis and ensemble clustering approach and comparison, *Data & Knowledge Engineering* (2018), doi: 10.1016/j.datak.2018.04.001.

Sentiment Analysis: An Automatic Contextual Analysis and Ensemble Clustering Approach and Comparison

Murtadha Talib AL-Sharuee[a,*], Fei Liu[a], Mahardhika Pratama[b]

[a] *Department of Computer Science and Information Technology, La Trobe University, Bundoora, Victoria 3086, AUSTRALIA*

[b] *School of Computer Science and Engineering, Nanyang Technological University*

Abstract

Product reviews are one of the most important resources to determine public sentiment. The existing literature on review sentiment analysis mostly utilizes supervised models, which usually suffer from domain-dependency and require expensive manual labelling effort to provide training data. This article addresses these issues by describing a completely automatic and unsupervised approach to sentiment analysis. The method consists of two phases, which are contextual analysis and unsupervised ensemble learning. In the implementation of both phases, a sentiment lexicon, SentiWordNet, is deployed. Using effective contextual procedures and modifying the base learning component (the k-means algorithm) results in developing a successful approach to sentiment analysis which can overcome the domain-dependency and the labelling cost problems. The results show that the proposed nonrandom initialization of k-means yields a significant improvement compared to other algorithms. In terms of accuracy and performance, the proposed method is effective compared to supervised and unsupervised approaches. We also introduce new sentiment analysis problems relating to Australian airlines and home builders which could be potential benchmark problems in the sentiment analysis field. Our experiments on datasets from different domains show that contextual analysis and the ensemble phases improve the clustering performance in term of accuracy, stability and generalizability.

Keywords: Text mining, sentiment analysis, unsupervised learning, contextual analysis, ensemble learning, k-means algorithm

1. Introduction

Web development has changed human interaction and communication drastically and has led to an enormous and rapid growth in user-generated data. Thus, a very large number of product reviews is currently available which is rapidly and continuously increasing. Considerable attention has focused on analysing this data in terms of the sentiment it conveys, which has resulted in the emergence of the sentiment analysis (SA) research field. SA involves the computational analysis of user-generated materials, such as reviews, to determine its orientation (positive, negative or neutral). There are two main reasons to automate SA: first, the abundance of online materials is beyond human analysis; and second, public opinion is a significant consideration when governments, institutions, and individuals are making decisions. Many diverse domains and applications can benefit

*Corresponding author.

Email address: al-sharuee.m@students.latrobe.edu.au (Murtadha Talib AL-Sharuee)

from SA, including those in the political [63, 28] linguistic [21] and financial [49, 56, 54] domains.

SA has been considered on different analysis levels such as the document [48, 35], entity and aspect level [50, 76] and sentence level [74]. In this research work, the proposed method processes product reviews at the documents level. In the literature, a large variety of techniques has been suggested to address SA, with the most commonly used techniques being supervised learning techniques. In the earliest study by Pang et al. [48], naive Bayes, maximum entropy and support vector machine classifiers were trained on different feature sets. Their reported results show that support vector machines yield the best results with most of the utilized types of features. Other studies suggest graph-based semi-supervised learning methods [55, 20].

More complicated learning methods were introduced to enhance performance. For example, in [1] a feature selection technique is developed using a binary version of Particle Swarm Optimization (PSO), which considers the features that participate in the training of maximum entropy, conditional random fields and support vector machines. Ensemble learning methods were also suggested to address SA by combining the results of processing different classifiers and different text representations [19, 64, 22].

Most of the proposed methodologies were based on a supervised paradigm which usually produces a domain-dependent model that cannot effectively handle unseen data. In addition to this, it requires pre-training on labeled data which likely needs an expensive and time-consuming manual annotation effort. These issues seriously affected their usability and effectiveness because the targeted data are varied for different domains and rapidly accumulated, requiring constant training and human intervention. This motivates us to develop an effective and completely automatic unsupervised solution for SA which can overcome the drawbacks of existing methods.

This article introduces an unsupervised and completely automatic method for clustering reviews which consists of two phases, contextual analysis and ensemble clustering. The first phase enables automatic contextual analysis by effectively deploying a sentiment lexicon. SentiWordNet 3.0 is utilized to prepare the underlying text for further processing and to address common linguistic forms which are intensifiers, negation, and contrast. It is important that the proposed algorithm addresses sentiment modifiers because they are very common forms in natural language and they lead to significant sentiment modification. For example, the sentence "It is not a good movie" is considered a positive expression if the negation is not taken into consideration.

The second phase of the proposed method is binary ensemble clustering which is implemented by assembling the results of a modified k-means algorithm. Ensemble learning improves the clustering result because it handles the bias-and-variance problem better than a single model approach. The feature set consists of the adjectives and adverbs in all the documents which are extracted after dealing with the polarity shifters. This feature set is used to build several vector space models with different weight schemes to be combined using the voting mechanism. The k-means algorithm is modified by generating two polar initial centroids (seeds) using SentiWordNet to divide the feature set into groups based on the features' sentiment orientations in the lexicon. The positive group forms a positive initial centroid, and the negative group forms a negative initial centroid. These seeds will also be used later on in the process for group identification.

The main contribution of the research is that it defines an approach to address the domain-dependency and the annotation cost problems in SA as it is an unsupervised labeling-free method. It introduces a completely automatic method which requires no

training or human participation, and it is effective in processing high volume data. Our method does not produce a prediction model and it is suitable for a real-world SA system because usually, the actual need is to analyze a large quantity of reviews, not to predict a single or a few instances. Few studies have introduced unsupervised clustering to the field because of the high complexity of natural language which is difficult to handle using an unsupervised learning methodology. Enhancing unsupervised learning using linguistic rules and dealing with the drawbacks of the k-means algorithm, which are low accuracy, group interpretation and instability, has resulted in a promising clustering algorithm.
We also introduce two new problems which can be benchmark problems in the sentiment analysis field. The two datasets are Australian Airlines and HomeBuilders review datasets, which were scraped from productreview.com.au. In the Airlines dataset, there are 750 reviews for each class (positive and negative) and in the HomeBuilders dataset, there are 1100 reviews for each class. The following are the main contributions of this research:

- Introducing a reliable domain-independent algorithm by combining contextual analysis and unsupervised ensemble learning.
- Modifying k-means using SentiWordNet to generate two initial seeds, and discussing a reliable method for group interpretation.
- Two new reviews datasets, namely Australian Airlines and HomeBuilders, are collected and tested along with several other datasets.
- Uniquely handling intensifiers and negation using SentiWordNet, in addition to considering contrast.
- Providing a comparison between different clustering algorithms.

The proposed method utilizes the SentiWordNet lexicon to determine the terms' polarity, therefore for languages other than English, an alternative corresponding sentiment lexicon or method for clustering the features is required.
The organization of the remainder of the article is as follows: section 2 gives a review of the related work. In section 3, we describe the algorithm and give a background on the related methods. We also provide a comparison of several clustering algorithms in section 4. Section 5 presents the experiment data and analysis. In section 6, a conclusion is drawn.

2. Related work

Approaches to SA can be divided into two main types of methods, lexicon-based and machine learning methods. Lexicon-based methods are regarded as symbolic approaches because they simply rely on the appearance of documents' terms in a lexicon. Usually, these methods classify documents by aggregating sentiment polarity scores. In the earliest work by Hatzivassiloglou and McKeown [24], adjectives conjoined by "and" or "but" and their semantic orientation were used to form a graph, which is then processed by a clustering algorithm to produce groups of polar adjectives. The lexicons are either manually generated, such as MaxDiff [30], MPQA [67] and General Inquirer [57] or automatically generated, such as SentiWordNet [17]. The manual labelling or scoring of terms is subjected to the annotator's judgment which can be inconsistent from word to word.

Therefore, in our method, we use an automatically generated lexicon (refer to section 3) which also contains a comparably large number of terms.

However, the accuracy of lexicon-based methods is usually not at a satisfactory level as they neglect changes in the actual sentiment strength when a term appears in different contexts, which results in machine learning methods being more commonly used in SA.

Machine learning methods were first used in a study by Pang et al. [48], which is considered cornerstone work in this field, in which three classic supervised classifiers were trained on several feature types. In the literature, supervised classification is the most common learning paradigm for SA [5, 42, 5, 77, 60]. In [77], naive Bayes and support vector machine classifiers were applied using different feature representations, such as unigram, unigram_freq, bigram, bigram_freq, trigram, and trigram_freq extracted from restaurant reviews.

To handle natural language complexity, more complex supervised algorithms were suggested, for instance, ensemble learning [72, 70, 65], where certain mechanisms can be used to combine the results of several classifiers and vector space models, which can increase the accuracy rate. However, supervised learning suffers from the domain-dependency problem and usually cannot deal effectively with completely unseen data [78]. In addition to this, a high level of manual intervention is required in order to provide training data, which is expensive and time-consuming.

Cross-domain sentiment classification methods were suggested to solve the domain-dependency problem [8, 46, 9] by dealing with the training feature set because usually, the features are domain-specific. In [9], feature relatedness scores are computed to automatically build a sentiment-sensitive thesaurus using labeled and unlabeled data. Then, this thesaurus is utilized to expand the feature vector by adding relevant features that can enhance the training. Xia et al. [71] suggested a feature ensemble plus sample selection method in which four parts of speech feature groups are established to train four classifiers. They tackled the domain-dependency issue by increasing the weight of domain-free features and decreasing the weight of domain-specific features in the training process of the weighted ensemble. High computational complexity and labeled data availability are the main drawbacks of the suggested cross-domain SA methods.

To this end, unsupervised learning can be an ideal solution for domain-dependency and high manual intervention issues. Clustering-based approaches to SA have been investigated in a few research studies. Li and Liu [35] suggested a SA clustering approach leveraging the k-means clustering algorithm. They apply a voting mechanism to remedy k-means instability and decide the group membership of a document. Using the TFIDF weighting method with adjective and adverb features increases the accuracy rate by more than 15%. To enhance the performance, they applied Kamps et al. [27]'s method to obtain the term score using WordNet, which led to an increase in the accuracy rate. However, their approach relies on a random first centroids selection which affects its stability and performance. It also relies on experimentally chosen seeds for group identification, which means the seeds have to be selected each time new data is processed. In [40], experiments were conducted to test several clustering algorithms with different weight schemes. They reported that k-means is suitable for balanced datasets. Another comparative work by Ma et al. [40] concluded k-means results in higher accuracy on average.

We utilize ensemble clustering which is preceded by automatic contextual analysis because applying clustering solely will not yield a generic effective performance. Contextual analysis methods usually address common linguistic structures such as negation and contrast, which have also been referred to as sentiment shifters [38], sentiment modifiers [44],

polarity shifters [70, 36] and valence shifters [59, 51].
Contextual rules have been applied in many studies [59, 51, 44] to obtain higher accuracy and to tackle natural language ambiguity. For instance, in [70], a rule-based method is proposed to detect text containing negations and contrasts, which was used to train a component classifier of an ensemble method. Another component classifier was trained on processed reviews, where the negations have been removed and an antonym dictionary was used to replace the negated terms. The dictionary was built by deploying a weighted log-likelihood ratio algorithm.
In addition to contextual analysis, we also use nonstochastic initial starting points for clustering several data representations, and automatic group identification which is illustrated in section 3. These procedures ensure accuracy, instability, and reliability at competent and adequate levels by which the notion of using unsupervised clustering becomes suitable and effective for addressing SA.

3. An Automatic Contextual Analysis and Ensemble Clustering (ACAEC)

The main idea underpinning our method is to provide an efficient solution that can overcome the domain-dependency and labeling cost problems of the commonly-used supervised learning paradigm in SA. The solution is an unsupervised method which is domain-independent and completely automatic, meaning no human participation is required. ACAEC (Figure 1) is a two-phase hybrid method: the first phase is data preparation and contextual analysis where steps are taken to automatically prepare and clean the text followed by the processing of common language phenomena, such as intensifiers, negation, and contrast using specialized dictionaries. ACAEC's second phase is an unsupervised clustering algorithm ensemble where k-means is a base algorithm which operates on several data representations.
In contextual analysis, a simple construction of dictionaries based on SentiWordNet polarity scores results in obtaining effective and domain-nonspecific dictionaries. Intensifiers and negation processing is uniquely done by utilizing the dictionaries to add and replace sentiment-expressing terms instead of adjusting the word score which is common in the literature. This enhances the outcome of clustering ensemble by effectively capturing the conveyed sentiment.
In ACAEC's second phase, we propose the nonstochastic and polar initial starting points (seeds) for k-means by which the overall performance is significantly improved in terms of accuracy and stability (refer to Table 5). The designated initial starting centroids have a significant impact on k-means' overall performance [69, 11, 34, 26], therefore, a large number of methods [31, 10, 2, 58, 45] have been proposed to select proper initial starting points instead of the random initialization of standard k-means. Celebi et al. [11] reviewed the existing k-means initialization methods providing a comprehensive evaluation in terms of complexity. In comparison, our solution for this issue is an unsophisticated and computationally undemanding solution. In ACAEC, the initial starting points are formed from the feature space utilizing SentiWordNet. The clustering interpretation issue has also been solved using polar seeds in the ensemble clustering.
The method performance and reliability are also enhanced by combining different data representations using weight schemes experimentally tested in terms of effectiveness.
Experiments have been conducted on new real-world datasets alongside already publicly-available datasets to test and evaluate the method.
We also provide a comparison between different clustering algorithms in terms of accuracy.

In the following, a detailed illustration of ACAEC and a comparison of the algorithms is provided.

3.1. SentiWordNet

We utilize the specialized sentiment lexicon SentiWordNet in the implementation of both phases of ACAEC. SentiWordNet 1.0 [17] is an automatically generated lexicon in which three scores (positive, negative and objective) are assigned to each synset from WordNet. The scores measure the strength of each terms' polarity by assigning a value for each of the three classes, where the total of these values is equal to 1.0, and each class has a partial value based on the strength of the three invoked sentiments. A committee of eight ternary semi-supervised classifiers was utilized to build this lexicon. In this work, the enhanced version SentiWordNet 3.0 [4] is used, which is based on WordNet 3.0, where in addition to the committee classifier, random walk is used to enhance the scores. An improvement of over 19% was reported [4] when using the updated version.

Algorithm 1 Producing a lexicon of adjectives and adverbs associated with the average scores.

INPUT: A SentiWordNet Lexicon L contains all terms w_m
OUTPUT: Lexicon U of adjectives and adverbs u with average scores

for all terms w_m in L **do**
  for all synsets s such that $w_j \in s$, $s \in L$ **do**
    if w_j is an adjective or adverb **then**
      $vPos_j = \frac{1}{n}\sum_{(i=1)}^{n} pos_i$, where n is the synsets number
      $vNeg_j = \frac{1}{n}\sum_{(i=1)}^{n} neg_i$, where n is the synsets number
      Assign $vScore_j = vPos_j - vNag_j$ to u_j
      Add $(u_j, vScore_j)$ to U
    end if
  end for
end for

As SentiWordNet is generated based on WordNet's synsets, the same word in SentiWordNet can have different scores, because it may appear in several synsets. Using the polarity value of an individual synset requires a text ambiguity analysis approach, which is another research direction that will not be covered in this work. Therefore, the average of the synsets' scores for each term is used to build lexicon U (Algorithm 1). The obtained lexicon contains adjectives and adverbs with their average scores extracted from SentiWordNet. This is then used in the implementation of the contextual analysis and ensemble clustering phases. The score of a SentiWordNet's term is expressed in equation(1).

$$Score = pos - neg \tag{1}$$

where pos is the positive scores of term w_j, and neg is the negative scores of term w_j.

3.2. Automatic Contextual Analysis

Contextual analysis is the first phase of ACAEC, which comprises five automatic and consecutive processes for preparing reviews and tackling common language forms. SentiWordNet has been effectively utilized to generate specialized dictionaries for some of these processes. The five processes are (1) data preparation; (2) spelling correction; (3) intensifier handling; (4) negation handling; and (5) contrast handling.

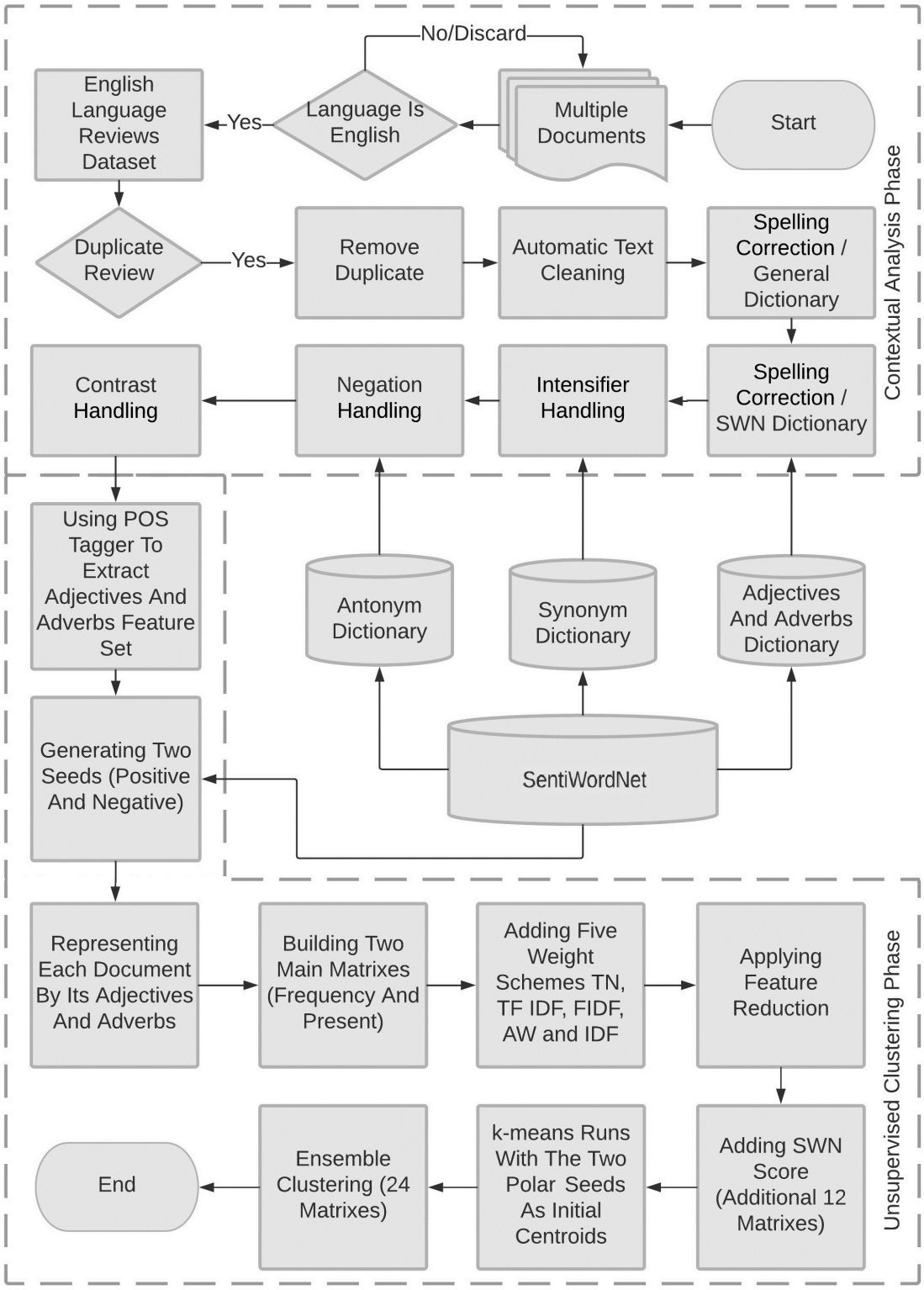

Figure 1: Flow chart of the Automatic Contextual Analysis and Ensemble Clustering method (ACAEC)

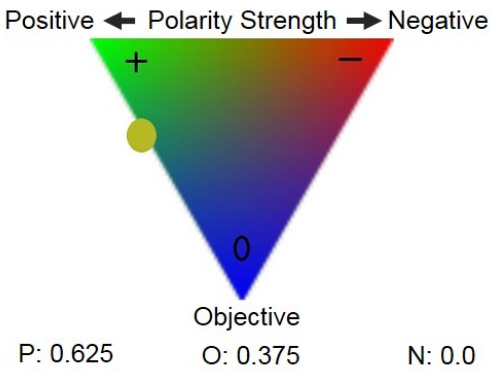

Figure 2: Example of SentiWordNet 3.0 online graphical representation of first sense of the word 'faithful' as an adjective.

3.2.1. Data Preparation

Language Detection. The first step of the algorithm is detecting the language using a language detection tool implemented by Cybozu Labs[1]. The library utilizes a naive Bayesian classifier and is reported to achieve over 99% accuracy for 53 languages. Language detection is considered because we are interested in processing reviews written in the English language only and it is likely that an underlying text from an online source will contain reviews that are written in languages other than English.

Data Cleaning. This step enables automatic data cleaning which is significant when processing raw web text. The cleaning method is role-based and involves removing duplicated reviews and XML tags. It also involves converting the text to lowercase letters and processing each review to separate non-separated tokens and sentences which results in more accurate sentence boundary detection and tokenization in the following processes.

3.2.2. Spelling Correction

Spelling correction plays an important role in ACAEC because misspelled terms cannot be processed in the following analysis which uses tools such as a POS tagger and dictionaries such as the antonym dictionary. Thus, correcting as many misspelled terms as possible can enhance performance, especially if there is a large number of misspelled adjectives and adverbs because these parts of speech will be the feature set of the clustering process. Misspelled words are corrected using two dictionaries, one being a general dictionary after which a specialized dictionary derived from SentiWordNet [2] is used to correct the adjectives and adverbs in the reviews.

3.2.3. Intensifier Handling

An intensifier is normally an adverb in a sentence which quantifies the strength of an adjective. For instance, in the sentence "the performance was extremely successful", "extremely" is an intensifier, which shifts the sentiment from positive to extremely positive. Intensifiers are common and effective sentiment shifters, hence addressing this type of polarity shifter improves the algorithm's performance. We deal with intensifiers by utilizing a synonym dictionary which is generated from lexicon U (refer to Algorithm 1).

[1] http://labs.cybozu.co.jp/en/
[2] http://sentiwordnet.isti.cnr.it/

The dictionary contains all the adjectives and adverbs in SentiWordNet where each synonym pair is chosen to be of the same or close sentiment score regardless of their semantic meaning. We focus on adjectives and adverbs because they will form the feature set for the clustering phase. A predefined list of intensifiers is defined and used in the process of identifying the intensifiers. ACAEC handles intensifiers by replacing the intensifier with a synonym of the intensified term.
Let $I = \{I_1, I_2, \ldots, I_n\}$ $(n > 0)$ be a sequence of intensifiers and $A = \{A_1, A_2, ..., A_m\}$ $(m > 0)$ be the sequence of adjectives. In the sentence $S = \{w_1, w_2, \ldots, w_l\}$ $(l > 0)$ if there exists $k(k > 0)$ such that $w_k = I_i$ and $w_{k+1} = A_j(1 \leq i \leq n$ and $1 \leq j \leq m)$, then I_i is an intensifier of A_j, and I_i can be replaced by A'_j which is a synonym of A_j. For instance, given the expression "so popular", the word "so" is replaced with the word "palmy" which is the synonym of "popular" in the dictionary. In this way, the invoked sentiment, whether positive or negative, can be detected by adding synonyms which will be extracted as features for the algorithm's next learning phase.

3.2.4. Negation Handling

Another common explicit language form is negation, which changes a term's polarity to negative, for example, the word "didn't" changes the polarity of the statement "I didn't like the movie" to negative. To identify negative statements, we use a predefined list of negation terms such as "not" and "never". Then, to process negation, we build an antonym dictionary to replace adjectives and adverbs that follow negation terms with their opposite sentiment words. Therefore, in the above example, after removing the negation term, the word "like" is changed to "hate".
A similar approach was suggested in [70],however our method differs in that it processes positive/negative adjectives and adverbs only and also, we used SentiWordNet to build the dictionary. The dictionary is a list of pairs of polar terms which have been extracted from lexicon U (refer to Algorithm 1). The antonym words are antonyms in terms of sentiment strength, regardless of their actual meaning. When using a rule-based method to process negation, the scope of those words which are located close to the negation term, and are likely to be negated, needs to be specified. We tested different scopes and experimentally found that a five-word scope after a negation term is the most effective.

3.2.5. Contrast Handling

Contrast is another commonly used language structure in English. When a sentence contains a contrast term, it is followed by a clause that summarises the author's opinion. This will be on the focus in order to determine the sentiment of the sentence. For example, in the sentence "It is a classic feel movie but unfortunately being a cynic", the overall sentiment is expressed in the part that follows the contrast word "but" which is "unfortunately being a cynic". To identify contrast in a document, a list of predefined contrast words, such as "but" and "however", has been identified. Then, every sentence in each review which contains a contrast term is processed separately. Let $S = \{w_1, w_2, \ldots w_m\}$ be the sentence which includes contrast terms, and $C = \{c_1, c_2, \ldots c_n\}$ denotes the set of contrast terms. All words w_j of the revoked part will be removed and the words in the conclusion part will be kept.

3.3. Ensemble Clustering

Ensemble learning is an effective technique, especially when the targeted data is complex and can be represented in many forms. Although ensemble learning imposes a higher

complexity compared to a singleton learner, it is capable of generating a model of high diversity, which enhances accuracy and generalization power. Therefore, we use this technique with several vector space models, where each model represents the dataset in a unique weight scheme. The base component of ensemble clustering is a modified k-means algorithm.

3.3.1. k-means Algorithm

The component algorithm of the proposed ensemble method is the k-means algorithm. It is a statistical and conventional clustering mean with hard boundaries in which the produced clusters are of unshared instances. It is a simple, flat, hard and polythetic clustering algorithm, with a predefined number of clusters. Several researchers have contributed to the design of the algorithm for different disciplines.

The algorithm is suitable for our experiments because (1) k-means is an unsupervised clustering algorithm, therefore, it is suitable for a domain-independent method; (2) k-means will always converge with a low number of iterations [2], which we also observe experimentally (refer to Tables 3 and 4). The low number of iterations is also a result of a proper first centroid selection; (3) although predefining the number of clusters and hard clustering can be considered drawbacks of k-means, it is adequate for our method because ACAEC produces only two positive and negative clusters, and by using k-means we can pre-assign the number of clusters; (4) the instability of k-means is addressed via non-random initial centroid selection, which also enhances its accuracy.

The default k-means is initiated by selecting k random centroids (vectors) from a given dataset [69]. Firstly, the centroids are randomly selected after which each data point is assigned to its closest centroid via a similarity measurement, such as cosine distance or Euclidean distance or another appropriate measurement method. The next step is to set the average of the clustered points in each group as the new centroid for the corresponding cluster. Then, by iteratively recalculating the closest distances and the cluster means and setting the new centroids to the obtained groups, the convergence condition is obtained when no new centroids are found.

The performance of k-means is highly influenced by (1) the initial centroid selection; (2) data representation; and (3) distance measurement. We chose cosine distance because prior experiments have shown that this leads to more accurate results. In the following, we describe our attempt to enhance the performance of ACAEC using k-means as a base algorithm by focusing on first starting points using SentiWordNet and data representation.

3.3.2. Vector Space Models (VSM)

The vector space model is a commonly used representation in text processing, where terms are features and documents are observations. In ACAEC, the documents are represented by their adjectives and adverbs which are the sentiment-expressing part of speech [6]. A variety of matrix representations has been used (refer to Algorithm 2) to obtain the most accurate results. With the proposed system, it is possible to experiment with 24 vector space models which are different representations of approximately 2000 documents from each dataset in a comparably short time. This is mainly because it is an unsupervised system and using polar nonstochastic initial starting centroids with k-means, which is efficient compared to hierarchical clustering algorithms, results in a reduction in the computational complexity of the method. To build various VSMs, two matrixes are generated, namely the presence matrix and the frequency matrix.

- Presence matrix. This represents each document by a binary vector, where a value of 1.0 represents the presence of a particular feature in a document.

- Frequency matrix. This represents each document as a vector in the VSM, where each value is the logarithm of f the count of a feature's occurrences in a document (equation 2).

$$f = \log_{10}(f + 1) \tag{2}$$

For both matrixes, the following weights are used in the experiments and the results are shown in Tables 3 and 4. In addition to these weights, the VSM number is increased by adding scores from lexicon U (refer to Algorithm 1) to each matrix (Figure 3).

Term Normalization (TN) [13]. TN measures the importance of a term in a particular document, where the numerator is a word count and the denominator is the length of a document where the word occurs. It expresses the importance of the word, taking into consideration the differences in the documents' lengths. It seems to be a reasonable method in dealing with documents of unbalanced length, as in the set of movie reviews, where, for example, the shortest document contains only 17 words and the longest consists of over 2500 words. Equation (3) is the mathematical expression of term normalization.

$$tf_{i,j} = \frac{t_{i,j}}{l_j} \tag{3}$$

where t_i is the frequency of term i, l_i is the length of document j where term i is occurred.

Algorithm 2 Constructing the vector space models

INPUT: A corpus D

OUTPUT: A set of matrix files M_n

Create M_n empty matrix files, n is 24 matrixes
for all document $d_j \in D$ **do**
    Create a presence vector vp_j, and frequency vector vf_j
    Add vf_j to M_1
    Add vp_j to M_2
end for
for all vectors $vf_j \in M_1$ and $vp_j \in M_2$ **do**
    for all feature f_i **do**
        $f_i * weigth_i$, $weight_i$ denotes $TF, IDF, TFIDF, WFIDF$ and AW
    end for
    Add the new vector $v_{j=\{1,...,10\}}$ to $M_r (2 > r > 13)$
end for
Remove the neutral features
Add $vScore_j$ from lexicon U to the 12 matrixes to fill up another 12 matrixes of M_n

Inverse Document Frequency (IDF). IDF is usually used as a part of another weighting scheme, along with the measurement of term importance or frequency in a document. It measures the importance of a term in a given corpus, regardless of the term's importance in a particular document. As empirically observed, using IDF can be more effective in some circumstances than combining it with another term's importance measure. Equation

(4) (4) is the mathematical expression of IDF.

$$idf_i = \log\left(\frac{D}{df_i}\right) \tag{4}$$

Let D be the number of all the documents and df_i is the number of documents where term i occurs.

Term Frequency Inverse Document Frequency (TFIDF) [13]. TFIDF is a plausible and commonly used scoring scheme in text mining tasks. It measures the importance of a particular word not only in the document but also in the corpus via inverse document frequency. TFIDF is proportional to the term frequency value and offsets the inverse document frequency value. It is expressed mathematically by equation (5).

$$tfidf = tf_{i,j} * idf_i \tag{5}$$

where tf_{ij} is the term normalization of term i, and idf_i is the inverse document frequency of term i.

Weight Frequency Inverse Document Frequency (WFIDF) [41]. WFIDF is another common weight mechanism that has been proposed to improve the accuracy of text mining systems. It is a proposed solution to the drawback of using term frequency which is the assumption that the count of the number of appearances of a term in a document is equal to the count of the significance of a single occurrence. Equation (6) is the mathematical expression of WFIDF.

$$wfidf = \begin{cases} 1 + \log tf_{i,j}.idf_i, & \text{if } tf_{i,j} > 0 \\ 0, & \text{Otherwise} \end{cases} \tag{6}$$

where tf_{ij} is the term normalization of term i, and idf_i is the inverse document frequency of term i.

Average of Weights (AW). The average of the two weights, TFIDF and WFTDF, is calculated and the obtained results are more accurate than using a single weight scoring method (Table 3). Equation (7) is the mathematical expression of the average weight of term i in document j.

$$AW = \frac{(tfidf_{i,j}) + (wfidf_{i,j})}{2} \tag{7}$$

3.3.3. Neutral Term and Feature Reduction

Neutral terms can be considered as redundant features for our experiment because we are interested in two classes only, positive and negative, and it is assumed that no sentiment polarity is likely to be expressed by neutral features. Therefore, feature reduction can be conducted by eliminating the neutral terms. Careful consideration should be given before using other feature selection methods after removing the neutral features as it may lead to the inaccurate clustering of short documents in the high sparsity vector space.

3.3.4. Polar Seeds

In our approach, ACAEC, we propose using two polar seeds (refer to Algorithm 3) as nonstochastic initial starting points and cluster identifiers. Positive seed S_{pos} and

negative seed S_{neg} are automatically extracted from the feature set F, which are then inserted into the processed corpus D. This is implemented by matching each feature (adjectives and adverbs) in the feature set against the lexicon U (refer to Algorithm 1).

Nonstochastic and polar initial starting points. The initial selection of k-means starting points is an important factor in forming the final clusters, hence the process and outcome of clustering highly depends on the first iteration where the initial starting points are selected. The selection of the first centroids seems to be the main factor that affects the number of iterations and the convergence of the algorithm.
A random selection, in default k-means, can result in poor performance because in the binary clustering process, for example, these two randomly selected points can be of the same class, which will lead to inaccurate clustering based on similarity. Some suggestions were introduced to address this problem, such as k-means++ [2] which selects distanced initial starting centroids. However, the selection of initial noninformative points or outlier points which are uncorrelated and dissimilar from any of the other documents is another reason for the degradation of the performance of the clustering method, in spite of selecting dissimilar centroids.
Other studies suggest genetic algorithms to address this issue [32, 3]. Operating the algorithm several times is another suggestion to overcome the drawback of the random selection of the first centroids. In [35], several results of k-means runs were combined using a voting mechanism, however their method is still based on a stochastic initialization and it does not completely eliminate the instability problem.

Algorithm 3 Insert the two polar seeds

INPUT: A set of feature F
OUTPUT: Insert the positive S_{pos} and the negative S_{neg} seeds into the corpus D

1: **for all** $f_{i(1\leq i\leq size(F))} \in F$, where F is the feature set. **do**
2: Match f_i against $vScore_j$ of $u_j \in U$, U is the extracted set of terms by Algorithm 1
3: **if** $vScore_j > 0$ **then**
4: Add f_i to S_{pos} , S_{pos} is the positive seed
5: **else if** $vScore_j < 0$ **then**
6: Add f_i to S_{neg}, S_{neg} is the negative seed
7: **end if**
8: **end for**
9: Insert S_{pos} and S_{neg} into D, D is the corpus

In ACAEC, the polar seeds S_{pos} and S_{neg}, which are produced by Algorithm 3, are used as the nonstochastic initial starting centroids of k-means. These two polar centroids are guaranteed to be of different classes (i.e. distanced points) and are always informative and can correlate with most of the documents in the processed data. This initialization eliminates the instability problem because k-means will always produce the same clusters when operating on the same dataset. It is a computationally inexpensive and unsophisticated solution to effectively solve the instability of k-means and improve the clustering.

Clusters interpretation. The k-means algorithm requires an interpreting strategy when processing real-world data because no labels are provided to identify the acquired groups' polarity, that is, whether they are positive or negative. We use the polar seeds S_{pos} and

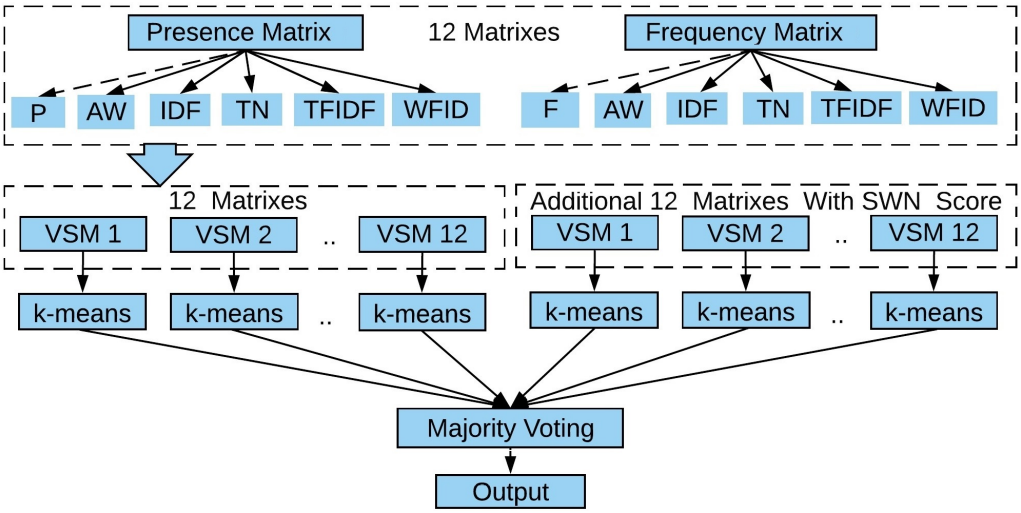

Figure 3: Ensemble method

S_{neg} (refer to Algorithm 3) to identify the sentiment orientations of the clusters. The assumption is that a positive cluster is where the positive seed S_{pos} appears and a negative cluster is where the negative seed S_{neg} appears. The seeds are highly oriented because all polar features are distributed between both of them. Therefore, assigning each seed to the right group by k-means is a straightforward process. Even weak ensemble components can easily assign these seeds correctly, as observed in the experiments.
We examine the case where an ensemble component assigns both seeds S_{pos} and S_{neg}, incorrectly to their opposite clusters by comparing our method to Li and Liu [35]'s method. Their method is based on using a confusion matrix (refer to Table 1), where a, b, c, and d are the number of documents, thus, $Cluster1$ is positive and $Cluster2$ is negative if $(b+c) \geq (a+d)$, otherwise vice versa. Throughout all the experiments, no contradictory interpretation is observed between using a confusion matrix and using the seeds.
However, when the two seeds appear together in one group, where one seed is misclustered, the ACAEC method gives no interpretation and neither group is determined as positive or negative. Therefore, the results of this ensemble component will not be considered in the ensemble. To this end, utilizing the seeds can be considered a reliable indication of the groups' identification because an ensemble component will always either correctly cluster the two seeds or miscluster one of them, which will be neglected in the ensemble.

3.3.5. Ensemble Learning

Ensemble learning (Figure 3) is a combination of several learners to achieve higher accuracy. It can combine learners of the same type, for example, bagging and boosting ensemble methods [66, 64]. It can also be an ensemble of different types of learners [14]. The ensemble algorithms that have been proposed for sentiment analysis are mostly supervised algorithms [65, 62, 37, 18]. They differ in the learning and the feature selection stage of the base classifiers and in its base classifier combination methods. The idea is that an ensemble can be more accurate compared to a single classifier if the component classifiers are diverse and accurate [23]. An accurate classifier, also referred to as a weak classifier by Schapire [53], is a classifier whose performance is better than random guessing, according to [15, 53].
An ensemble method often enhances performance because its outcome is due to the

base learners' results being collected and combined in a certain way, such as voting or weighting. As a result, complex problems can be solved, even by a combination of weak classifiers. It can also solve the overfitting problem, avoiding potential computational failure, such as a stack in local optima and solving complex problems which might be too difficult to solve using a single learner [15]. These advantages motivate us to examine the effect of an ensemble method by applying majority voting on the results of the modified k-means algorithm, with pre-specified initial starting centroids on different VSMs.
The diversity of the ensemble components is obtained by using different weight schemes, and also their accuracy is enhanced compared to random guessing by using polar seeds S_{pos} and S_{neg} as initial starting points. More importantly, in ACAEC, assembling is significant for the groups' identification. The chance of inaccurately misclustering these polar seeds in ensemble learning is extremely low because most of the ensemble components are able to allocate the seeds correctly, and this is considered a very strong indication of the groups' meaning. To enhance accuracy, and because a few of the weak learners, as previously mentioned, may miscluster one of the two seeds, S_{pos} and S_{neg} (refer to Algorithm 3), these components' results can be ignored when both seeds appear in the same group.

3.3.6. Ensemble Clustering Algorithm

The ensemble algorithm is as follows:

Algorithm 4 Pre-processing and the ensemble of the clustering algorithm

INPUT: A corpus D of m number of documents $\{d_1, d_2, \ldots, d_m\}$
OUTPUT: Assign a *positive* or *negative* label to each document $d_{i=\{1,2,\ldots,m\}} \in D$

Pre-processing:

for all document $d_j \in D$ **do**
  for all each word $w_i \in d_j$ **do**
    Tag w_i with part of speech tagging t_j
    if $t_j == a$ OR $t_j == r$, a and r denote adjective and adverb respectively **then**
      Keep w_i
      Add w_i to F, F is the features set
    else
      Remove w_i
    end if
  end for
end for

The idea of combining several VCMs not only leads to more reliable ensemble learning, it also has more flexibility because a future enhancement can be made by using additional weight schemes or another component algorithm that is suitable for large data analysis. However, extending the ensemble approach will increase the computational complexity; therefore, another component learner should be carefully selected.

3.3.7. Computational Complexity Analysis

If the complexity of k-means is $O(g(nkt))$ where n is the dataset instances, k is the number of clusters, and t is the number of iterations, then the computational complexity of ACAEC is $O(mg(nkt))$. The complexity of ensemble methods is mostly linear with respect to the number of components m, and it basically depends on the complexity of

Algorithm 4 Pre-processing and the ensemble of clustering algorithm

Clustering:

Set the number of clusters $K = 2$
for all matrix files M_i, $(i = 1, 2, \ldots, n)$, **do**
    Initialize positive seed S_{pos} and negative seed S_{neg} as first centroids
    Cluster M_i into two clusters G_1 and G_2 using k-means H_i and cosine similarity
    if $S_{pos} \in G_1$ and $S_{neg} \in G_2$ **then**
        H_i algorithm is accurate enough
        G_1 is the positive cluster, G_2 is the negative cluster
    else if $S_{pos} \in G_2$ and $S_{neg} \in G_1$ **then**
        H_i algorithm is accurate enough
        G_2 is the positive cluster, G_1 is the negative cluster
    else
        H_i algorithm is NOT accurate
    end if
end for

Voting:

for all $d_j \in D$, D is the corpus **do**
    for all result R_i of H_i **do**
        if H_i algorithm is accurate enough **then**
            if $\sum(d_j(R_i) = positive \geq \sum(d_j(R_i) = negative)$ **then**
                $d_j = positive$
            else
                $d_j = negative$
            end if
        end if
    end for
end for

the base learner. In addition, the computational cost of cosine distance, which is the similarity measurement of the base learner, depends on the vector length. Therefore, feature reduction can slightly improve the performance.

4. Comparison Of Clustering Algorithms

Several clustering algorithms are compared to show their effectiveness in grouping documents into k clusters in terms of its invoked sentiment. This comparison also shows the improvement resulting from using S_{pos} and S_{neg} as the initial starting points in clustering the data using k-means.
Six different data representations of the 24 matrixes produced by Algorithm (2) are selected for the experiments, as shown in Table 5. The compared algorithms are as follows.

- Different initialization methods for k-means.
    - The polar points: Using S_{pos} and S_{neg} produced by Algorithm (3), as the initial starting points of k-means which is also the base algorithm of the ensemble.
    - k-means [39]: Simple k-means algorithm where k initial starting points are randomly selected from a given VSM.

  - Subsample k-means: A random 10% subsample of a VSM is preliminarily clustered to select the initial starting centroids.
  - k-means Uniform: k initial starting points are drawn uniformly at random, where each centroid's value is selected from the interval between maximum and minimum components of that value within a VSM.
  - k-means++ [2]: Starts with the initialization phase to select dissimilar initial centroids where the probability of choosing each of these centroids is proportional to its overall potential contribution.

- Partitioning Around Medoids (PAM) [29]: This solves the k-medoids problem which is closely related to k-means however it is based on finding medoids instead of centroids. In the experiments, k-means++ is used to find the first starting data points after which representative medoids, which are data points, are selected by iteratively calculating the distance and the total cost. It forms clusters by minimizing the distances around the medoids and assigning each point to its closest medoid.

- Clustering LARge Applications (Clara) [29]: This randomly chooses a data subset from given data and then repeatedly performs PAM. In every iteration, the full data is grouped around the medoids and the algorithm stops when the mediods do not change.

- Gaussian mixture model (GMM) [43]: The k-means ++ principle is used for the initialization of the expectation-maximization (EM) algorithm to fit full covariance matrixes to the data. It groups the data points by maximizing the component posterior probability by assigning observations to the multivariate normal components.

- Fuzzy c-means clustering (FCM) [16, 7]: This is a soft clustering algorithm where membership grades measure the degree of belonging of each data point to multiple clusters.

- Agglomerative Hierarchical Clustering (AHC): This is a complete linkage algorithm which starts by considering each data point as a cluster. Then, at each stage, two groups with the smallest complete linkage distance are merged until two clusters are formed.

5. Experiments and Analysis

In order to evaluate the method, we conduct experiments on different review datasets (refer to Table 2). For evaluation purposes, usually, when using machine learning algorithms, an experimental dataset is divided into training and testing portions. In ACAEC, the entire dataset is used for evaluation because it is an unsupervised method. The positive/negative actual labels that are attached to each document are used to construct a confusion matrix. The allocation of the polar seeds S_{pos} and S_{neg} enable us to identify the orientation of the produced clusters (Table 1).
As we are interested in both negative and positive classes, the evaluation is done by calculating the accuracy [41, 25]. Equation (8) is a mathematical expression for calculating accuracy based on the confusion matrix and the seeds' positions. In addition to accuracy, we also calculate precision, recall, and F-measure [41].

Table 1: Confusion matrix.

	Actual negative	Actual positive
*Cluster*1/Positive	a	b
*Cluster*2/Negative	c	d

$$accuracy = \frac{a+b}{a+b+c+d} \tag{8}$$

ACAEC is implemented with Java 8 and NetBeans IDE 8.0.2. The experiments were conducted on a Dell machine with a 3.40 GHz Intel Core I7 CPU and 16GB RAM, running Windows7 Enterprise. For comparison with other machine learning algorithms, we used MATLAB 9.1 and Weka 3.8.1.

5.1. Datasets

The across domain performance of ACAEC is evaluated by experimenting on two datasets, the Australian Airlines and HomeBuilders review datasets. We also conduct experiments on the movie dataset and multi-domain datasets [8] (refer to Table 2).

5.1.1. Airlines and HomeBuilders Datasets

Publicly available online reviews are collected from www.productreview.com.au which is an Australia consumer opinion website. Each review is associated with one of five rating categories (excellent, good, ok, bad and terrible). This enables us to select reviews with excellent and good ratings as positive instances and reviews with bad and terrible ratings as negative instances.

Airlines dataset. To construct this dataset, 1500 reviews on four Australian airlines are randomly collected. These reviews were written between September 2006 and January 2017.

HomeBuilders dataset. The reviews collected in this dataset are on 14 home builders' companies in Australia. These reviews were written between January 2009 and January 2017.

5.1.2. Movie and Multi-domain Datasets

The movie review dataset in [47], which is the enhanced version of Pang et al. [48]'s dataset, is a well-known dataset in the field of sentiment analysis and has been used in many research studies. It is widely believed that movie reviews are difficult documents to analyze compared to other product reviews [12, 68]. This is because many aspects are likely to be discussed and different polarities can be invoked. The wide variety of movies can complicate this task even further because of the number of subjects being discussed in the reviews, such as the plot of the movie, the actors, and the movie's location. It is also likely to contain unbalanced samples of different lengths, which can also cause difficulties in analyzing short documents.

The multi-domain dataset [8] is a benchmark dataset which was constructed by Blitzer et al. [8] using reviews on different products taken from Amazon.com. Reviews on four domains are used in the experiments. The datasets have the same balanced composition, that is, 1000 positive documents and 1000 negative documents, except for baby product reviews, where there are 900 reviews for both classes. Each review in both datasets was automatically labeled using the rating information associated with each document, which is provided by the authors.

Table 2: Datasets

Datasets	Number of samples		Sources
	Positive	Negative	
Airlines	750	750	http://www.productreview.com.au
HomeBuilders	1100	1100	
Movie [47]	1000	1000	http://www.imdb.com
Kitchen [8]	1000	1000	https://www.amazon.com
Apparel [8]	1000	1000	
Toys&Games [8]	1000	1000	
Baby [8]	900	900	

5.2. First Phase of ACAEC

In the following, we detail the results of each procedure of the first phase. Figure (4) shows the effect of the contextual analysis phase on accuracy where the accuracy rate increases by an average of about 3.0 percent when applying contextual analysis procedures.

Data Preparation. Figure 5 shows a slight enhancement in accuracy when preparing the data compared to applying the ensemble method to raw text. This step is significant for the following procedures and also for the second phase because it enhances the process of tokenization and sentence boundary detection.

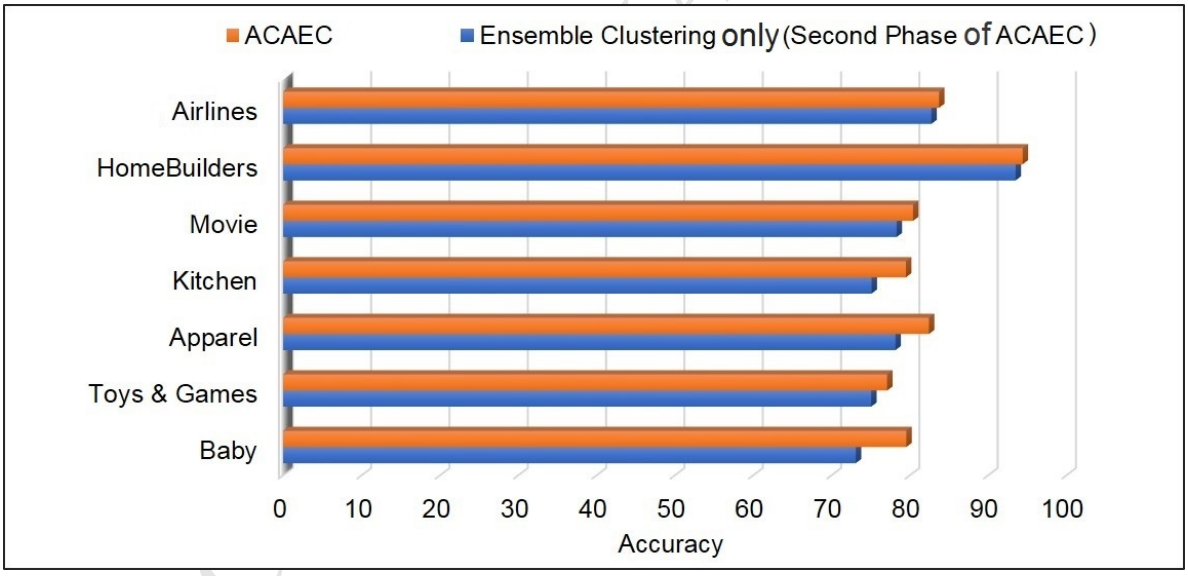

Figure 4: The effect of the first phase on accuracy

Spelling Correction. Correct spelling positively affects the result because it assists processing and extracting as many adjectives and adverbs as possible in the following steps. When processing raw web text, there is a need for data preparation and spelling correction because it is very likely the text will contain misspelled terms.

Intensifier Handling. An improvement is noticed when processing the intensifiers, which is due to the strong sentiment intensifying caused by these terms and also to the common use of intensifiers.

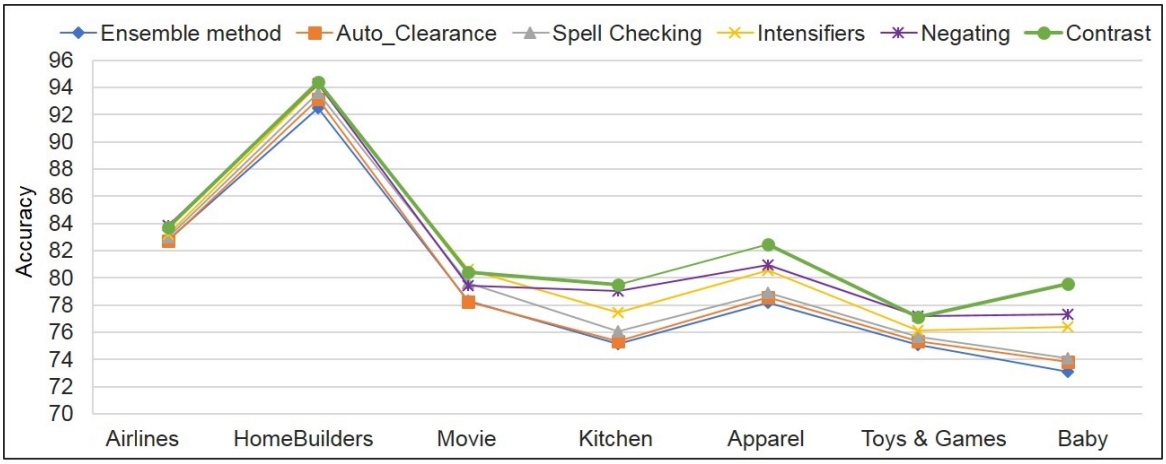

Figure 5: The effect of each procedure of the first phase on accuracy

Negation Handling. This is a common form of language structure which results in strong polarity shifts. As shown in Figure 5, processing negation increases the accuracy for four datasets.

Contrast Handling. This is the last procedure of the first phase and addresses the contrasts, resulting in a considerable enhancement in processing two datasets (Apparel and Baby), and a slight enhancement in the other datasets. As a preceding stage, the contextual analysis procedures improve the outcome of ACAEC (Figures 4 and 5).

5.3. Second Phase of ACAEC

The experiments were conducted by obtaining high-dimensional matrixes of all adjectives and adverbs as features. To extract the adjectives and adverbs, we use the Stanford part-of-speech tagger [61]. A matrix represents all the documents of each dataset in a VSM, where each document is a vector in the vector space. This model was proposed for the information retrieval system [52]. In this representation of the corpus, the order of terms in a document is ignored and the sparsity of the obtained matrix is very high.

Vector Space Models. The experiment results of the ensemble components of ACAEC on the Airlines and HomeBuilders datasets using five weighting schemes are shown in Tables 3 and 4. The first two matrixes that were tested are the presence matrix and the frequency matrix. The frequency matrix is generally inferior to the presence matrix in terms of accuracy but the difference between these matrixes' results decreases significantly when the weight schemes are used. One of these weights is TN which always leads to lower accuracy, probably because it measures the term importance regardless of its importance to the entire corpus. The effect of the terms' weights in the entire corpus becomes clearer when we used the IDF, where the term weight in a particular document is neglected. IDF enhances the performance, as shown in Tables 3 and 4. Using the standard weights TFIDF and WFIDF with the presence and frequency matrixes significantly enhances accuracy from at least 5% to over 20% for the Airlines and HomeBuilders datasets.

Feature Reduction Effect. Undertaking careful feature selection usually improves the learning process in terms of efficiency and effectiveness. Irrelevant features can negatively affect the learning process [33, 75]. Therefore, to enhance the algorithm's performance,

Table 3: Results of operating the ensemble components on the Airlines dataset using five weighting schemes

Matrixes	Accuracy	Precision	Recall	F-measure	Iterations	Time in seconds
Frequency	63.16	61.22	71.9	66.14	13	1
Frequency-AW	83.34	87.33	78.03	82.42	19	1
Frequency-IDF	83.94	87.61	79.09	83.14	12	1
Frequency-TN	61.29	58.69	76.43	66.4	13	1
Frequency-TFIDF	79.95	84.19	73.77	78.64	13	0
Frequency-WFIDF	83.48	87.04	78.7	82.66	12	1
Presence	75.95	79.55	69.91	74.42	17	0
Presence-AW	83.88	84.63	82.82	83.71	11	0
Presence-IDF	84.21	85.99	81.76	83.82	9	1
Presence-TN	62.09	59.87	73.5	65.99	11	1
Presence-TFIDF	82.54	85.49	78.43	81.81	16	1
Presence-WFIDF	83.54	84.33	82.42	83.37	12	0

Table 4: Results of operating the ensemble component on the HomeBuilders dataset using five weighting schemes

Matrixes	Accuracy	Precision	Recall	F-measure	Iterations	Time in seconds
Frequency	87.55	94.03	80.18	86.56	9	1
Frequency-AW	94.5	95.62	93.27	94.43	11	1
Frequency-IDF	94.5	95.79	93.09	94.42	8	1
Frequency-TN	74.5	71.65	81.09	76.08	7	3
Frequency-TFIDF	92.27	95.06	89.18	92.03	15	1
Frequency-WFIDF	94.45	95.62	93.18	94.38	8	1
Presence	88.59	95.4	81.09	87.67	8	2
Presence-AW	93.95	95.66	92.09	93.84	13	1
Presence-IDF	93.73	95.64	91.64	93.59	9	1
Presence-TN	82.95	86.29	78.36	82.13	7	2
Presence-TFIDF	93.86	95.73	91.82	93.74	11	1
Presence-WFIDF	93.95	95.48	92.27	93.85	7	1

we conduct feature reduction by matching all adjectives and adverbs against lexicon U (refer to Algorithm 1).Since we are interested in positive and negative classes, only polar features are considered and the reduction is done by removing neutral terms because they do not carry the clustering characteristic of reviews. When applying feature reduction on the Airlines and HomeBuilders datasets, there are slight changes, which are shown in Figure 6.

Sentiment Scores. In Figure 7, the sentiment scores from SentiWordNet are added to all the matrixes. The polarity score has a negative impact on accuracy which was anticipated because the sentiment score is the average score of the synsets to which a term belongs, and the context in which a term occurs is not considered. However, the average score is likely to correctly indicate the term polarity, that is, whether it is positive, negative or neutral. This step doubles the number of VCMs which promotes the cluster interpretation process because more ensemble components will participate in deciding the polar seeds' memberships.

Experiments on Multi-domain Datasets. After conducting the contextual analysis and constructing the matrixes, the last step is to feed the matrixes into the ensemble method,

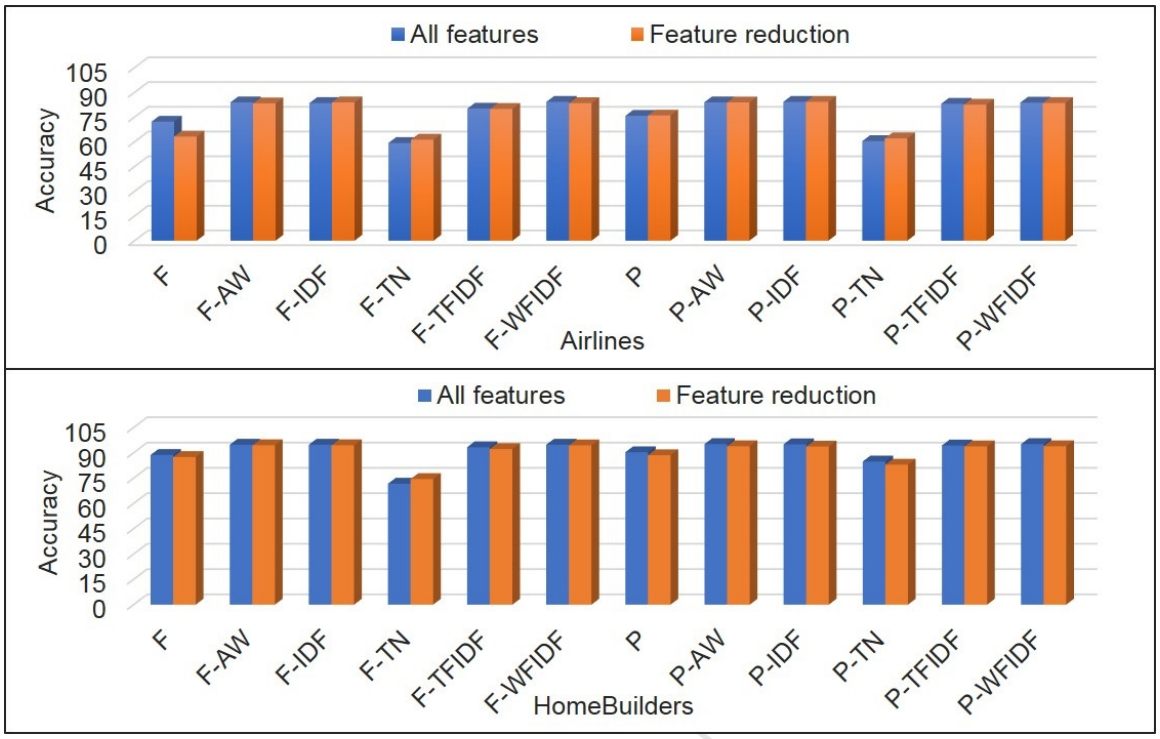

Figure 6: Feature reduction effect on accuracy

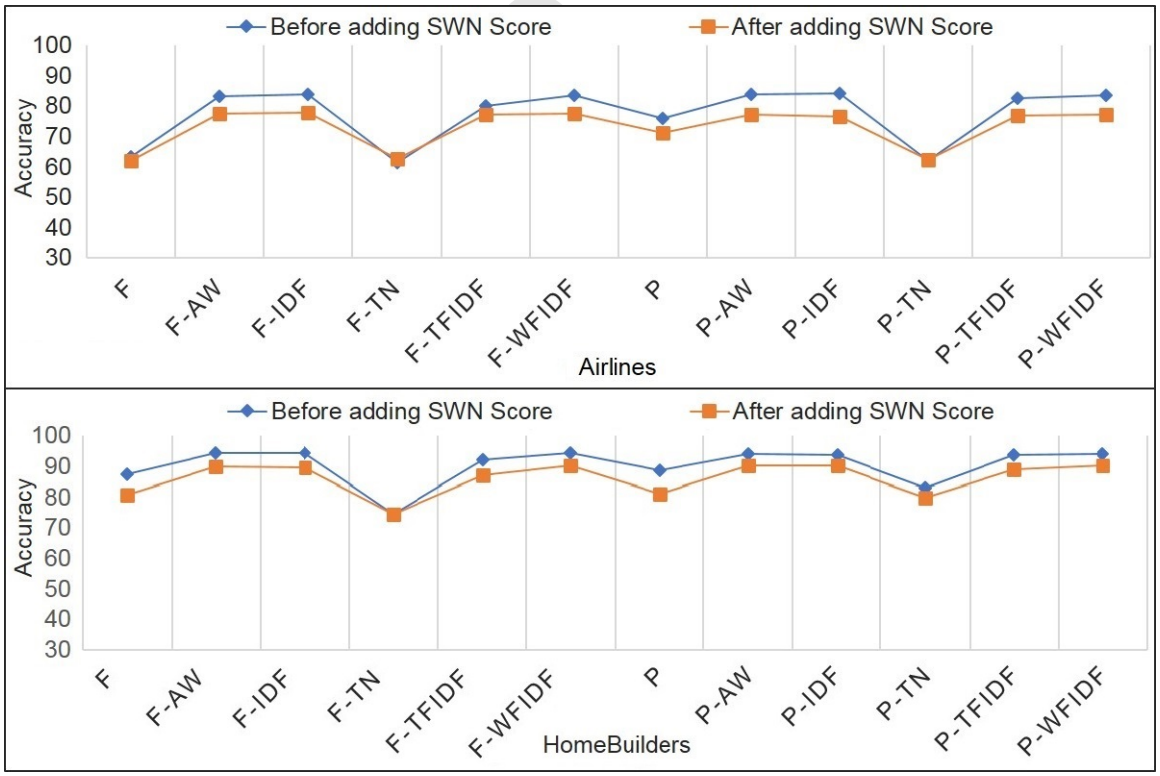

Figure 7: Adding SentiWordNet score to Airlines and HomeBuilders datasets.

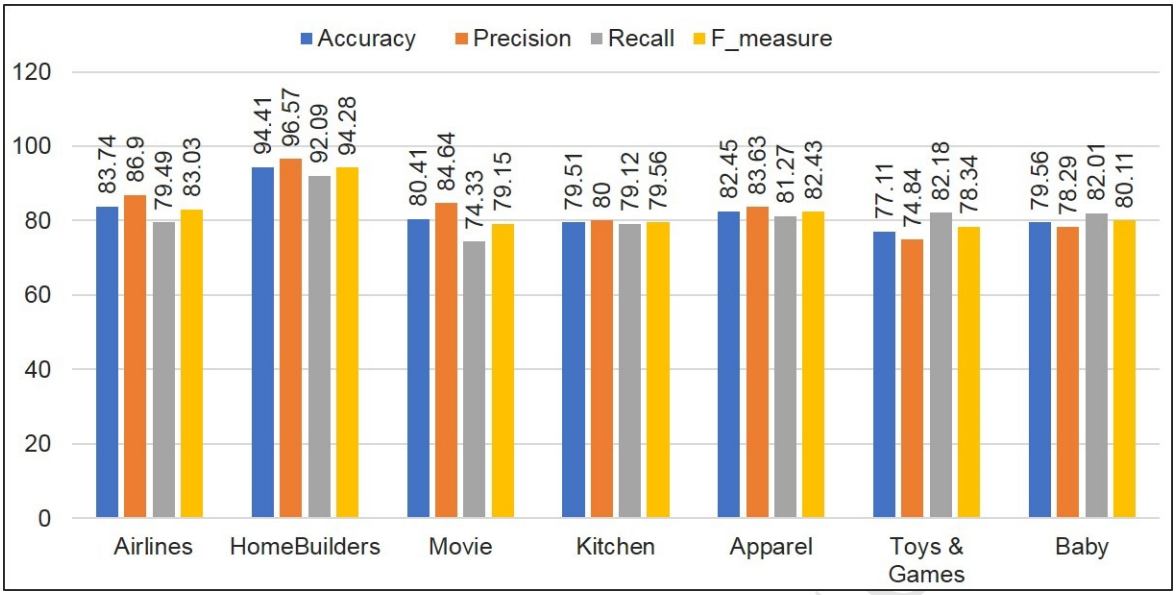

Figure 8: The performance of ACAEC on different datasets

where a document will be assigned to a positive/negative group if the majority of the ensemble components agree on the clustering decision. The ensemble method combines 24 matrixes which are the VSMs shown in Tables 3 and 4 in addition to those produced by adding the SentiWordNet score. This variation makes the group interpretation more reliable because the two seeds are assigned by operating on various data representations. The results of the experiments on the multi-domain datasets are shown in Figure 8. In addition to the Airlines and HomeBuilders datasets, five sets of product reviews (movie, kitchen, apparel, toys and games and baby) were compared. The accuracy rate is between 94.41% and 79.56% for six of the datasets, however the toys and games dataset has an accuracy of 77.11%. In general, the results show that ACAEC is a domain-independent algorithm with competitive accuracy.

Comparison of Clustering Algorithm. In Table 5, ten clustering algorithms are compared using cosine distance, except with the GMM and FCM methods where Euclidean distance is used and for AHM where Spearman measurement is used. The experiments are conducted on six VSMs which are constructed after being contextually analyzed using the first phase of ACAEC. For a more reliable comparison, the methods that have an initial randomization are run 20 times and the mean of the accuracy and the standard deviation are shown.

5.4. Discussion

Table 6 compares ACAEC and the seven different classifiers. Five of them are supervised classifiers, namely support vector machine (SVM), random forest (RF), decision tree (J48), naive Bayes (NB) and multinominal naive Bayes (MNB). To conduct experiments using supervised classifiers, all part-of-speech tags are used as a set of features and TFIDF weight is also utilized. We also report the results of a clustering-based method by Li and Liu [35] on a sample of the movie review dataset. In addition, a simple classifier based on SentiWordNet (SWN-based) is constructed to classify a document by aggregating SentiWordNet's average scores, obtained by Algorithm 1, of its adjectives and adverbs to determine the polarity.

Table 5: Comparison of clustering algorithms by showing the mean of the accuracy rates (Mean), and the standard deviation (SD).

Algorithms		Presence	Presence -TFIDF	Presence -WFIDF	Frequency	Frequency -TFIDF	Frequency -WFIDF
Polar Points, Algorithm (3)	Accuracy	**75.95**	**82.54**	**83.54**	**63.16**	**79.95**	**83.48**
k-means	Mean	69.64	78.28	80.92	62.37	77.35	81.94
	SD	6.62	7.68	6.62	1.92	5.64	0.97
Subsample k-means	Mean	66.33	80.69	81.99	62.69	77.22	81.35
	SD	6.91	1.34	0.87	0.15	4.96	1.09
k-means Uniform	Mean	68.17	80.41	82.06	62.75	76.54	81.67
	SD	4.54	1.42	0.66	0.19	6.41	1.26
k-means++	Mean	67.95	79.65	82.26	62.33	78.04	81.63
	SD	6.09	6.12	1.05	1.89	4.98	1.13
k-medoids	Mean	60.93	56.62	52.85	66.11	60.49	53.59
	SD	0.88	9.11	3.99	0	2.74	6.74
Clara	Mean	59.28	58.41	56.68	60.28	56.62	57.08
	SD	5.2	5.7	4.05	5.38	9.11	4.13
FCM	Mean	61.25	61.47	53.64	60.67	60.57	57.03
	SD	1.58	1.81	1.34	1.61	1.65	1.35
GMM	Mean	51.37	50.05	52.96	50.18	50.03	52.14
	SD	0.81	0.03	1.64	0.1	1.42	0.79
AHC	Accuracy	60.37	54.77	56.17	51.77	59.17	55.97

The results show that the average rate of the SWN-based classifier's accuracy is comparably low which is probably because the average score extracted from SentiWordNet does not accurately reflect the sentiment strength of a term which is mainly because the language context in which this term appears is neglected. As shown in Table 6, the performance of ACAEC is competitive compared to both supervised and unsupervised methods. The average accuracy of ACAEC is very close to the average accuracy of SVM which is the best average rate of the compared methods. ACAEC also yields the best performance on three datasets and has comparable performance on the other four datasets. The accuracy of ACAECC is enhanced by at least 2% compared to the unsupervised method by Li and Liu [35] which is due to the contextual analysis phase, using polar seeds and utilizing diverse weight schemes.

The results in Table 5 show a significant enhancement when using the polar seeds S_{pos} and S_{neg} which outperforms the other algorithms. It also solves the problem of k-means instability in an efficient way. Unlike other methods, such as Li and Liu's method proposed in [35], this study suggests a more robust and reliable solution because for every run on the same data, the algorithm guarantees the same performance and output which is due to nonstochastic centroid initialization. In addition, ACAEC provides a more reliable group interpretation strategy using ensemble learning to assign the polar seeds (refer to Algorithm 3).

The advantages of the method are: (1) it is a competitive method in terms of accuracy; (2) it is stable and domain independent; and (3) it requires no human participation (i.e. unlike the supervised learning methodologies, it requires no training).

Table 6: Evaluation

Datasets	ACAEC	SVM	RF	J48	NB	MNB	Clustering [35]	SWN-based
Airlines	83.74	**86.4**	85.07	70.93	73.33	85.33	—	73.13
HomeBuilders	94.41	93.45	**94.73**	85.45	80.91	94.36	—	86.41
Movie	80.41	**83.6**	75.6	68.6	66.4	75	77.17 - 78.33	66.1
Kitchen	**79.51**	77.6	76.4	69.8	72.2	77.6	—	70.65
Apparel	**82.45**	79.8	80.6	67	71.2	77.8	—	73.6
Toys&Games	77.11	**79.6**	79	70.2	75.8	76	—	70.0
Baby	**79.56**	77.11	77.11	64.9	70.22	75.11	—	67.11
Average	82.45	**82.5**	81.21	70.98	72.86	80.17	—	72.42

Research implications. This study has shown that SA can be effectively addressed by unsupervised clustering learning which results in a domain-independent algorithm. Our findings from the experiments on multi-domain datasets show the merit in adopting a cluster analysis method for SA. ACAEC involves two phases that improve the outcome: contextual analysis and an ensemble of clustering algorithms. Utilizing contextual analysis has a significant impact on the results because the language forms which are tackled are very common and can be strong sentiment shifters, such as negation and intensifiers. In ensemble learning, we use the traditional representation of a corpus where the documents are the observations and the words are the features (adjectives and adverbs). This study supports what has been suggested in the previous research [6] that adjectives and adverbs are the most informative parts of speech in terms of sentiment analysis. However, for binary problem analysis, only polar adjectives and adverbs are significant for learning, which can be seen (Figure 6) when eliminating neutral terms which have no significant impact on the results. The experiment results using diverse term weighting schemes indicate that term weighting in the entire corpus is more important compared to its weight in a particular document.

The ensemble method has positive implications for ACAEC. Increasing the number of diverse and accurate ensemble members slightly enhances the algorithm's accuracy, which supports the work in [23]. More importantly, group judgment is more reliable in ACAEC as a result of ensemble clustering. The group identification issue was addressed in [35], where the authors observed 100 clustering results after which they defined 22 documents as solid polarity documents because they were always correctly grouped which is a result of their strong orientations. The possibility of incorrectly clustering these documents, which is 10^{-x} where x is the number of positive/negative documents, is very low. However, this low possibility is based on the experimented dataset and will probably be altered if there is a modification to the dataset size or if another dataset is used. Our solution for interpreting clusters is automatic and general and it can be applied to process any given corpus.

One of the research observations is that the generalization performance of ACAEC is enhanced which is a result of applying contextual analysis and using various data representations. Table 6, shows that ACAEC's performance is relatively stable when operating on different datasets compared to the other algorithms. For example, ACAEC yields higher accuracy when operating on the Kitchen dataset whereas the accuracy rates of the other algorithms are comparably low when processing this dataset. This is because of the two processing phases of ACAEC where the processed text is contextually analyzed in the first phase, then in the second phase, different data representations are combined.

This study shows that SA can be addressed by employing an unsupervised clustering

algorithm. K-means, as a base clustering algorithm, is suitable for our method because, in ACAEC, the cluster number is pre-defined, and k-means can process a large quantity of data in a short time because it is a non-hierarchical algorithm.
The comparison in Table 5 shows the impact of k-means initialization and also shows that k-means with cosine distance is more suitable for sentiment clustering analysis. The compared methods (refer to Table 5) perform inconsistently based on the utilized weights, whereas using polar seeds yields the best performance on all the weight schemes. Using the WFIDF weight scheme with k-means mostly leads to higher accuracy and less standard deviation. We enhanced the k-means algorithm by using nonrandom polar initial starting points which significantly increases k-means accuracy and efficiency. Selecting the initial points for k-means' first iteration is crucial and this has been a research topic for many studies [32, 3, 73].

6. Conclusions

In this article, we discussed a completely automatic unsupervised machine learning method for sentiment analysis. The method combines automatic contextual analysis and unsupervised ensemble clustering. Unsupervised learning and reliability are the features that distinguish the proposed method from the other work in the literature. The reliability of ACAEC is derived from the combination of the contextual analysis phase and the ensemble learning methodology. It is an unsupervised solution with competitive accuracy, and subsequently, it is a domain-independent analysis algorithm. ACAEC solves the problem of data annotation, which is an expensive process.
As future work, we will consider a multi-class problem based on the sentiment strength. An enhancement can also be achieved by considering deeper contextual analysis and utilizing other weighting schemes or even other machine learning approaches.

Acknowledgment
The authors would like to acknowledge the financial support from the Iraqi Ministry of Higher Education and Scientific Research (MoHESR).

References

[1] Akhtar, M. S., Gupta, D., Ekbal, A., Bhattacharyya, P., 2017. Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. Knowledge-Based Systems 125, 116–135.

[2] Arthur, D., Vassilvitskii, S., 2007. k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, pp. 1027–1035.

[3] Babu, G. P., Murty, M. N., 1993. A near-optimal initial seed value selection in k-means means algorithm using a genetic algorithm. Pattern Recognition Letters 14 (10), 763–769.

[4] Baccianella, S., Esuli, A., Sebastiani, F., 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Language Resources and Evaluation Conference (LREC). Vol. 10. pp. 2200–2204.

[5] Bai, X., 2011. Predicting consumer sentiments from online text. Decision Support Systems 50 (4), 732–742.

[6] Benamara, F., Cesarano, C., Picariello, A., Recupero, D. R., Subrahmanian, V. S., 2007. Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone. In: The International AAAI Conference on Web and Social Media (ICWSM). Citeseer.

[7] Bezdek, J. C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press.

[8] Blitzer, J., Dredze, M., Pereira, F., 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th annual meeting of the association of computational linguistics. pp. 440–447.

[9] Bollegala, D., Mu, T., Goulermas, J. Y., 2016. Cross-Domain Sentiment Classification Using Sentiment Sensitive Embeddings. IEEE Transactions on knowledge and data engineering 28 (2), 398–410.

[10] Cao, F., Liang, J., Jiang, G., 2009. An initialization method for the K-Means algorithm using neighborhood model. Computers & Mathematics with Applications 58 (3), 474 – 483.

[11] Celebi, M. E., Kingravi, H. A., Vela, P. A., 2013. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert systems with applications 40 (1), 200–210.

[12] Chaovalit, P., Zhou, L., 2005. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 112c–112c.

[13] Croft, W. B., Metzler, D., Strohman, T., 2010. Search Engines: Information Retrieval in Practice. Vol. 283. Addison-Wesley Reading.

[14] da Silva, N. F., Hruschka, E. R., Hruschka, E. R., 2014. Tweet sentiment analysis with classifier ensembles. Decision Support Systems 66, 170–179.

[15] Dietterich, T. G., 2000. Ensemble Methods in Machine Learning. In: Multiple classifier systems. Springer Berlin Heidelberg, pp. 1–15.

[16] Dunn, J. C., 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3 (3), 32–57.

[17] Esuli, A., Sebastiani, F., 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In: Proceedings of Language Resources and Evaluation (LREC). Vol. 6. Citeseer, pp. 417–422.

[18] Fersini, E., Messina, E., Pozzi, F., 2016. Expressive signals in social media languages to improve polarity detection. Information Processing & Management 52 (1), 20–35.

[19] Fersini, E., Messina, E., Pozzi, F. A., 2014. Sentiment analysis: Bayesian ensemble learning. Decision support systems 68, 26–38.

[20] Goldberg, A. B., Zhu, X., 2006. Seeing Stars when There Aren'T Many Stars: Graph-based Semi-supervised Learning for Sentiment Categorization. In: Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing. Association for Computational Linguistics, pp. 45–52.

[21] Guerini, M., Strapparava, C., 2016. Why do urban legends go viral? Information Processing & Management 52 (1), 163–172.

[22] Hagen, M., Potthast, M., Büchner, M., Stein, B., 2015. Webis: An Ensemble for Twitter Sentiment Detection. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). pp. 582–589.

[23] Hansen, L. K., Salamon, P., 1990. Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence 12, 993–1001.

[24] Hatzivassiloglou, V., McKeown, K. R., 1997. Predicting the Semantic Orientation of Adjectives. In: Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the European chapter of the association for computational linguistics. Association for Computational Linguistics, pp. 174–181.

[25] Hubert, L., Arabie, P., 1985. Comparing Partitions. Journal of classification 2 (1), 193–218.

[26] Ismkhan, H., 2018. I-k-means-+: An iterative clustering algorithm based on an enhanced version of the k-means. Pattern Recognition.

[27] Kamps, J., Marx, M., Mokken, R. J., De Rijke, M., et al., 2004. Using WordNet to Measure Semantic Orientations of Adjectives. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC). Vol. 4. Citeseer, pp. 1115–1118.

[28] Kato, Y., Kurohashi, S., Inui, K., Malouf, R., Mullen, T., 2008. Taking sides: User classification for informal online political discourse. Internet Research 18 (2), 177–190.

[29] Kaufman, L., Rousseeuw, P. J., 2009. Finding Groups in Data: an Introduction to Cluster Analysis. Vol. 344. John Wiley & Sons.

[30] Kiritchenko, S., Zhu, X., Mohammad, S. M., 2014. Sentiment Analysis of Short Informal Texts. Journal of Artificial Intelligence Research, 723–762.

[31] Kumar, K. M., Reddy, A. R. M., 2017. An efficient k-means clustering filtering algorithm using density based initial cluster centers. Information Sciences 418, 286–301.

[32] Laszlo, M., Mukherjee, S., 2007. A genetic algorithm that exchanges neighboring centers for k-means clustering. Pattern Recognition Letters 28 (16), 2359–2366.

[33] Law, M. H., Figueiredo, M. A., Jain, A. K., 2004. Simultaneous Feature Selection and Clustering Using Mixture Models. IEEE transactions on pattern analysis and machine intelligence 26 (9), 1154–1166.

[34] Li, C. S., 2011. Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters. Procedia Engineering 24, 324–328.

[35] Li, G., Liu, F., 2012. Application of a clustering method on sentiment analysis. Journal of Information Science 38 (2), 127–139.

[36] Li, S., Lee, S. Y. M., Chen, Y., Huang, C.-R., Zhou, G., 2010. Sentiment Classification and Polarity Shifting. In: Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, pp. 635–643.

[37] Li, W., Wang, W., Chen, Y., ???? Heterogeneous [e.

[38] Liu, B., 2012. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies 5 (1), 1–167.

[39] Lloyd, S., 1982. Least squares quantization in PCM. IEEE transactions on information theory 28 (2), 129–137.

[40] Ma, B., Yuan, H., Wei, Q., 2013. A Comparison Study of Clustering Models for Online Review Sentiment Analysis. In: Web-Age Information Management (WAIM). Springer, pp. 332–337.

[41] Manning, C. D., Raghavan, P., Schütze, H., et al., 2008. Introduction to Information Retrieval. Vol. 1. Cambridge University press Cambridge.

[42] McDonald, R., Hannan, K., Neylon, T., Wells, M., Reynar, J., 2007. Structured Models for Fine-to-Coarse Sentiment Analysis. In: Proceedings of the 45th annual meeting of the association of computational linguistics. pp. 432–439.

[43] McLachlan, G., Peel, D., 2004. Finite mixture models. John Wiley & Sons.

[44] Muhammad, A., Wiratunga, N., Lothian, R., 2016. Contextual sentiment analysis for social media genres. Knowledge-Based Systems 108, 92–101.

[45] Onoda, T., Sakai, M., Yamada, S., 2012. Careful Seeding Method based on Independent Components Analysis for k-means Clustering. Journal of Emerging Technologies in Web Intelligence 4 (1), 51–59.

[46] Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., Chen, Z., ???? Cross-domain [s.

[47] Pang, B., Lee, L., ???? A Sentimental Education: [s.

[48] Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, pp. 79–86.

[49] Peetz, M.-H., de Rijke, M., Kaptein, R., 2015. Estimating reputation polarity on microblog posts. Information Processing & Management.

[50] Pham, D.-H., Le, A.-C., 2017. Learning multiple layers of knowledge representation for aspect based sentiment analysis. Data & Knowledge Engineering.

[51] Polanyi, L., Zaenen, A., 2006. Contextual Valence Shifters. In: Computing attitude and affect in text: Theory and applications. Springer, pp. 1–10.

[52] Salton, G., 1971. The Smart Retrieval System-Experiments in automatic Document Processing. Prentice-Hall, Inc.

[53] Schapire, R. E., 1990. The strength of weak learnability. Machine learning 5 (2), 197–227.

[54] Sehgal, V., Song, C., 2007. SOPS: Stock Prediction using Web Sentiment. In: Proceedings of the International Conference on Data Mining Workshops (ICDMW). IEEE, pp. 21–26.

[55] Sindhwani, V., Melville, P., 2008. Document-word co-regularization for semi-supervised sentiment analysis. In: Eighth IEEE International Conference on Data Mining (ICDM'08). IEEE, pp. 1025–1030.

[56] Smailović, J., Grčar, M., Lavrač, N., Žnidaršič, M., 2014. Stream-based active learning for sentiment analysis in the financial domain. Information Sciences 285, 181–203.

[57] Stone, P., Dunphy, D. C., Smith, M. S., Ogilvie, D. M., 1968. The general inquirer: A computer approach to content analysis. Journal of Regional Science 8 (1), 113–116.

[58] Su, T., Dy, J. G., 2007. In search of deterministic methods for initializing K-means and Gaussian mixture clustering. Intelligent Data Analysis 11 (4), 319–338.

[59] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M., 2011. Lexicon-based methods for sentiment analysis. Computational linguistics 37 (2), 267–307.

[60] Tan, S.-S., Na, J.-C., 2017. Mining Semantic Patterns for Sentiment Analysis of Product Reviews. In: International Conference on Theory and Practice of Digital Libraries. Springer, pp. 382–393.

[61] Toutanova, K., Manning, C. D., 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora (EMNLP/VLC): held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 63–70.

[62] Tsutsumi, K., Shimada, K., Endo, T., 2007. Movie Review Classification Based on a Multiple Classifier. In: Proceedings of the annual meetings of the Pacific Asia conference on language, information and computation (PACLIC). pp. 481–488.

[63] Tumasjan, A., Sprenger, T. O., Sandner, P. G., Welpe, I. M., 2010. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In: In Fourth International AAAI Conference on Weblogs and Social Media (ICWSM). Vol. 10. pp. 178–185.

[64] Wang, G., Sun, J., Ma, J., Xu, K., Gu, J., 2014. Sentiment classification: The contribution of ensemble learning. Decision support systems 57, 77–93.

[65] Wang, G., Zhang, Z., Sun, J., Yang, S., Larson, C. A., 2015. Pos-rs: A Random Subspace method for sentiment classification based on part-of-speech analysis. Information Processing & Management 51 (4), 458–479.

[66] Whitehead, M., Yaeger, L., 2010. Sentiment Mining Using Ensemble Classification Models. In: Innovations and advances in computer sciences and engineering. Springer, pp. 509–514.

[67] Wilson, T., Wiebe, J., Hoffmann, P., 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, pp. 347–354.

[68] Wollmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K., Morency, L.-P., 2013. YouTube Movie Reviews: Sentiment Analysis in an Audio-Visual Context. Intelligent Systems, IEEE 28 (3), 46–53.

[69] Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al., 2008. Top 10 algorithms in data mining. Knowledge and information systems 14 (1), 1–37.

[70] Xia, R., Xu, F., Yu, J., Qi, Y., Cambria, E., 2016. Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis. Information Processing & Management 52 (1), 36–45.

[71] Xia, R., Zong, C., Hu, X., Cambria, E., ???? Feature Ensemble Plus Sample Selection: Domain [a.

[72] Xia, R., Zong, C., Li, S., 2011. Ensemble of feature sets and classification algorithms for sentiment classification. Information Sciences 181 (6), 1138–1152.

[73] Yedla, M., Pathakota, S. R., Srinivasa, T., 2010. Enhancing K-means Clustering Algorithm with Improved Initial Center. International Journal of computer science and information technologies 1 (2), 121–125.

[74] Yu, H., Hatzivassiloglou, V., ???? Towards Answering Opinion Q]uestions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences, booktitle = Proceedings of the 2003 conference on Empirical methods in natural language processing, year = 2003, pages = 129–136, organization = Association for Computational Linguistics,.

[75] Zeng, H., Cheung, Y.-M., 2009. A new feature selection method for Gaussian mixture clustering. Pattern Recognition 42 (2), 243–250.

[76] Zhang, W., Xu, H., Wan, W., 2012. Weakness finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis. Expert Systems with Applications 39 (11), 10283–10291.

[77] Zhang, Z., Ye, Q., Zhang, Z., Li, Y., 2011. Sentiment classification of internet restaurant reviews written in cantonese. Expert Systems with Applications 38 (6), 7674–7682.

[78] Zhou, G., Zhou, Y., Guo, X., Tu, X., He, T., 2015. Cross-domain sentiment classification via topical correspondence transfer. Neurocomputing 159, 298–305.