Aspect-based sentiment analysis using smart government review data

Omar Alqaryouti and Nur Siyam
The British University in Dubai, Dubai, United Arab Emirates
Azza Abdel Monem
Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, and
Khaled Shaalan
The British University in Dubai, Dubai, United Arab Emirates and School of Informatics, University of Edinburgh, Edinburgh, UK

Abstract

Digital resources such as smart applications reviews and online feedback information are important sources to seek customers' feedback and input. This paper aims to help government entities gain insights on the needs and expectations of their customers. Towards this end, we propose an aspect-based sentiment analysis hybrid approach that integrates domain lexicons and rules to analyse the entities smart apps reviews. The proposed model aims to extract the important aspects from the reviews and classify the corresponding sentiments. This approach adopts language processing techniques, rules, and lexicons to address several sentiment analysis challenges, and produce summarized results. According to the reported results, the aspect extraction accuracy improves significantly when the implicit aspects are considered. Also, the integrated classification model outperforms the lexicon-based baseline and the other rules combinations by 5% in terms of Accuracy on average. Also, when using the same dataset, the proposed approach outperforms machine learning approaches that uses support vector machine (SVM). However, using these lexicons and rules as input features to the SVM model has achieved higher accuracy than other SVM models.

Keywords Sentiment analysis, Aspect extraction, Aspect-based sentiment analysis, Lexicon approach, Rule-based approach, Government smart apps

Paper type Original Article

Publishers note: The publisher wishes to inform readers that the article "Aspect-based sentiment analysis using smart government review data" was originally published by the previous publisher of *Applied Computing and Informatics* and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Alqaryouti, O., Siyam, N., Monem, A.A., Shaalan, K. (2019), "Aspect-based sentiment analysis using smart government review data", *Applied Computing and Informatics*. Vol. ahead-of-print No. ahead-of-print. https://10.1016/j.aci.2019.11.003. The original publication date for this paper was 23/11/2019.

1. Introduction

Governments around the world aim to deliver quality government services that are efficient, cost-effective, and socially just while keeping the needs of the public at the centre of the provided services [1,2]. Digital sources such as smart applications reviews and online feedback information are considered one of the important sources to obtain customers' feedbacks and inputs. With the advances in machine learning and natural language processing techniques it is now possible to analyse the obtained data to measure citizens satisfaction with the services provided [3].

Sentiment analysis, also known as opinion mining, is the field of study that analyses people's opinions and attitudes towards entities such as products, services, and topics and their attributes. Since 2000, this field has become an active research area due to its applicability to almost every possible domain, from consumer products, services, and healthcare to social events and elections, providing a strong motivation for research [4].

In this paper, our objective is to design a model that helps government agencies gain greater insights and a broader holistic view of their customers' needs and expectations. Accordingly, the proposed model examines the data for the government mobile apps reviews to improve the government smart applications and services performance. Towards this end, we propose a model to integrate several lexicons and rule-based techniques to extract users' sentiments from smart government apps reviews. The dataset and lexical resources produced by Alqaryouti et al. [5] were used in this study. To extract structured and useful information from the unstructured sources of textual data, aspect extraction is considered. Aspect extraction is based on the notion that an opinion aimed at no target (i.e. aspect) is of limited use. Thus, the goal of aspect-based sentiment analysis is to discover sentiments and the aspects that those sentiments are aimed at [4].

This paper is the first attempt to provide a way that helps the governments and enterprises identify the improvement areas of their services by understanding their customers' feedbacks through smart apps reviews. The proposed model can highlight the issues in the government mobile apps in terms of the following aspects categories: user interface, user experience, functionality, performance, support and updates according to the users' reviews [5]. This helps in framing the challenges that need to be addressed by governments to transform their services through finding the most innovative solutions to increase the citizens' satisfaction.

This paper is divided into six sections. Section 2 provides a background about the selected topic and discusses the related work in the literature. Section 3 illustrates the research methodology and related experiments to extract aspects and classify the sentiments of smart government apps domain. Section 4 discusses the experiments performance results. Section 5 discusses the research implications. The last section provides the conclusion of our work.

2. Background and related work

Sentiment analysis is the computational study that analyzes individuals' attitudes, excitements, emotions, expressions, viewpoints and opinions towards certain entity. The opinion sentiment can be represented as a quintuple of five components as follows: (e, t, s, h, dt), where e represents the entity, t represents the target aspect or feature for the sentiment, s represents the sentiment towards the target, h represents the opinion holder, and dt represents the time stamp of the opinion. Figure 1 illustrates an example of a sample review with quintuple sentiment components extracted on one sample review.

There are different levels where sentiments can be analyzed, namely document, sentence and aspect levels. At document level, the whole review is considered as a basic information unit and is then classified into positive, negative or neutral opinions. Similarly, sentences are regarded as short documents. The sentiment in the document-level is determined for the document as a whole, which clarifies the sentiment associated to the target. In this case, the

opinion holder might have a positive opinion regarding the entity but might not be satisfied with all the "aspects" of that target. To extract such information, aspect-based classification is used. Aspect-based analysis covers both entities and aspects. It does so by decomposing the entity into aspects (aspect extraction), then classify each aspect sentiment into positive, negative or neutral (aspect sentiment classification), and finally, summarize the results of the previous steps.

Another sentiment analysis level that has been considered is the concept-level [6]. Unlike word-based approaches, concept-level sentiment analysis focuses on the semantic analysis of text through the use of web ontologies and semantic networks. This allows the combination of conceptual and affective words associated with natural language. For example, if the concept "Cloud Computing" is split into two words, the word "cloud" would be wrongly associated with the weather. However, concept-level sentiment analysis is limited by the bounds of the knowledge base and by the fact that it fails to detect important discourse structure information that is essential for effectively detecting the polarity expressed by natural language opinions [7].

In this paper, our concern is aspect-based sentiment analysis, which is one of the levels of sentiment classification. It comprises aspect or feature extraction, sentiment polarity prediction and classification, and sentiment aggregation [8,9]. Aspects extraction in sentiment analysis is now becoming an active area of research as it is the most vital task in the aspect-based recognition [4,5]. Aspect-based sentiment analysis is the process in which sentiments in respect to different aspects are detected [10]. Aspects are attributes, characteristics, or features of a product or service. Aspect extraction phase involves the identification of these review characteristics through consumers' comments to identify aspects. Next, the polarity prediction and classification take place to decide if the aspect sentiment polarity denotes positive, negative or neutral orientation as well as its strength or tone level [11]. In the previous example, the word "*amazing*" denotes a positive sentiment polarity towards the "*design*" aspect. The last step is to summarize the results according to the extracted aspects and their corresponding classified polarity. This summary is essential to determine the strengths and weaknesses of each aspect within the app and compared to others. This summarization of results can be done, qualitatively through text-based aggregated opinions summary [12], or quantitatively through graphical and analytical representation [13].

To automate the summarization of reviews, aspect words are grouped into aspects categories. In an unsupervised machine learning approach, the model tries to make sense of the data and extract features on its own. However, when the aspect categories are known in advance, and there is enough training data available, a supervised machine learning approach to aspect category and polarity detection is feasible and may yield better results [14]. On the other hand, semi-supervised approaches use a small set of labeled data to label a larger set of unlabeled data [15].

Reviews aspects are domain dependent and differ from a context to another. Aspect-based sentiment analysis has been broadly used in numerous application domains such as products reviews, social media, hotel reviews and restaurant reviews.
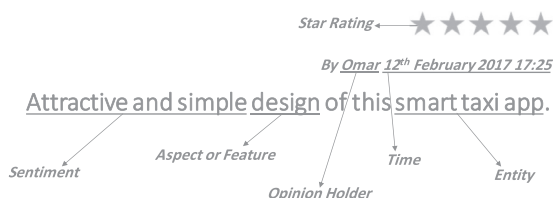

Figure 1.
Example for Sentiment Quintuple.

Researchers have reported various approaches in order to extract aspects from textual resources. For instance, Samha et al. [16] used frequent Part of Speech (POS) tags and rules in addition to opinion lexicon to identify aspects and opinion words from reviews as well as group them into categories and summarize the results. POS is the process of marking up a word in a text as corresponding to a particular part of speech (i.e. verb, adjective, preposition, etc.), according to the word's definition and context [15]. Devi et al. [17] proposed a feature-based approach for sentiment analysis using Support Vector Machine (SVM). The authors collected product reviews on laptops from e-commerce platforms such as Amazon and eBay. SentiWordNet, which is a lexical resource for sentiment classification and opinion mining applications, was used to identify objective sentences and later to identify the polarity for the opinion words. Furthermore, POS tagging was used to extract aspect terms from the dataset. The authors used Stanford parser to extract the opinion words and to find the grammatical dependencies to determine the connection between the opinion words and aspects extracted in the previous step. The dependencies also assist in determining the negations that were considered in calculating the polarity score. The SVM classifier was used to classify aspects and determine their sentiment polarity score. The result of the SVM classifier is a set of vectors that contain aspects and its opinion words for each review. The performance achieved significant results with overall accuracy of 88.16%.

Similarly, Manek et al. [18] proposed a feature selection method based on Gini index. The authors used SVM classifier to predict sentiment polarities for a movies' reviews dataset. In their approach, the reviews were pre-processed with tokenization, case transformation, filtering stop words and stemming. Further, the Term Frequency/Inverted Document Frequency (TF/IDF) has been adopted with the weighting mechanism using Gini Index as a feature selection approach. This helped in measuring the impurity of the attribute for categorization and creation of the feature vector for the top 50 attributes according to the Gini impurity index value. Then, the SVM classifier was applied in order to train and test the model. This approach achieved significant results with overall accuracy of 92.81%.

The introduction of the SemEval competition resulted in a rise to the number of proposed methods for aspects extractions. For instance, Mubarok et al. [19] used sentiment analysis and classification techniques to determine the sentiment polarity of restaurants reviews using the SemEval 2014 dataset. Feature extraction was performed using Chi Square, resulting in higher computational speed despite reducing the system performance. Naïve Bayes classification of sentiment polarity was used to classify both aspects and sentiments. The evaluation results indicated that the system performed well with a highest score of 78.12% for the F1-Measure. Al-Smadi et al. [20] proposed two supervised machine learning approaches, namely deep recurrent neural network (RNN) and SVM. The authors aimed to investigate three different tasks; aspect category identification, aspect opinion target expression (OTE), and aspect sentiment polarity identification. The Arabic Hotels' reviews dataset from the SemEval-2016 framework were used to evaluate the efficiency of the proposed approaches. When compared to baseline researches, the results indicated that SVM outperforms the deep RNN in all the investigated tasks. However, the deep RNN was found to be more appropriate and faster in terms of training and testing execution time. Moreover, Al-Smadi [21] used two applications of deep learning long short-term memory (LSTM) neural networks for aspect-based sentiment analysis of Arabic hotels' reviews. The first application was a character-level bidirectional LSTM with a conditional random field classifier used for aspect OTE extraction. The second application was an aspect-based LTSM for aspect sentiment orientation classification. The evaluation results indicated an improvement of 39% compared to baseline research for the aspect-OTE extraction application and 6% for the orientation classification application. Likewise, Al-Ayyoub et al. [22] proposed a supervised machine learning enhanced approach to extract aspects and classify sentiments for hotels' Arabic reviews in the SemEval-2016 dataset as well. The approach consisted of three tasks,

namely identifying aspect categories, extracting the opinion targets, and identifying the sentiment polarity. Evaluation results indicated that the study approach outperformed the benchmark approaches using the same dataset.

On the other hand, García-Pablos et al. [23] presented W2VLDA, an aspect sentiment classification system that requires minimal supervision and does not require language or domain specific resources. The system is able to distinguish opinion words from aspect terms in an unsupervised way. The only supervision required by the user is a single seed word per aspect and polarity. The system performance was also evaluated using the SemEval 2016 dataset. The analysis showed competitive results for different languages and domains. Similarly, Dragoni et al. [24] presented an aspect-based unsupervised system for opinion monitoring that supports data visualization. The authors adopted an open information extraction approach to extract the aspects. The system aimed at providing users with an effective analysis and visualization tool based on the user-generated content. The approach proved its effectiveness compared to baseline supervised approaches participated in SemEval campaign.

Rathan et al. [25] proposed an ontology-based feature level sentiment analysis model for the domain of "Smartphones" for tweets. The authors included various features such as spelling correction and emoji and emoticon detection. The lexicon-based approach was used for automated training data labelling. A "Smartphone" specific domain lexicon in addition to the general one was used to improve the accuracy of the classification using SVM. The system provided good accuracy and demonstrated the importance of using emoji and emoticons detection as well as the use of attribute specific lexicons.

Cambria et al. [26] proposed an ensemble technique of symbolic and sub-symbolic AI by employing a LSTM network to discover the verb-noun primitives by lexical substitution and link them to commonsense concepts in a new three-level knowledge representation for sentiment analysis, termed SenticNet 5.

On the other hand, Xianghua et al. [27] suggested an unsupervised approach to determine the aspects and sentiments in Chinese social reviews. The authors used Latent Dirichlet Allotiment (LDA) in social review to identify multi-aspect global topics. Then, they extracted the local topic and the related sentiment according to a sliding window through the review text. Chen et al. [28] improved the LDA model and presented the Automated Knowledge LDA which is a fully automatic approach that can use existing domain independent data to learn prior knowledge and identify new aspects. The Automated Knowledge LDA approach was able to produce aspects and resolve issues related to wrong knowledge by adopting and enhancing the Gibbs sampler method. Moreover, Poria et al. [29] asserted an unsupervised rule-based approach to obtain both explicit aspects as well as implicit aspects clues (IAC) from product and restaurant reviews. In the proposed approach, the authors used the IACs to identify the IACs in the review and plot them to the aspects they are signifying depending on common-sense knowledge and on the dependency structure of the sentence using WordNet and SenticNet.

Pham and Le [30] proposed a multi-layer architecture for denoting customers' hotels' reviews. Learning techniques such as word embeddings and compositional vectors models to obtain words, sentences, aspect-based graphs and higher aspect representation layers were used, resulting in rich knowledge of the input text. These representations were then integrated into a neural network to create a model for predicting the hotels' overall ratings, while being able to efficiently extract aspects ratings and weights.

Liu et al. [31] integrated both supervised and unsupervised domain independent automatic rule-based methods to improve double propagation. In their research, the double propagation method assumes that the opinion words will always have a target. Therefore, there is a syntactic relation between the opinion word and the target within the same sentence. This approach is able to select effective part of all rules with a better performance than the whole set. Lastly, Ma et al. [32] combined the LSTM network with a hierarchical

attention mechanism. The results indicated that the combination of the planned attention design and Sentic LSTM outperformed progressive strategies in aspect-based sentiment analysis tasks.

As can be noticed, most of the aforementioned works require labeled datasets for training their models for each of the domains. When a new domain is studied, it would be difficult to train the models without first extracting the domain specific aspects. When the domain independent parameters are used to describe the connections between aspects and their associated opinion expressions, the models will be able to capture the aspect specific sentiment with minimal data requirement [33].

3. Methodology

In this paper, we propose an integrated lexicon and rule-based aspect-based sentiment analysis approach to extract government mobile apps aspects and classify the corresponding implicit and explicit sentiments. This approach is selected due to the nature of the targeted dataset, which consists of short reviews and irregular sentences related to the various aspects of the government mobile apps. Shaalan [34] highlighted the significance of the rule-based over other approaches as it depends on manually created rules and because it is easier to integrate domain knowledge with natural language processing tasks. The propposed approach utilizes the manually generated lexicons by [5] with hybrid rules to handle some of the key challenges in aspect-based sentiment analysis in particular and sentiment analysis in general.

Figure 2 illustrates the proposed methodology. The following subsections illustrate the methodology in details to extract both explicit and implicit aspects in additions to its sentiment classification.

3.1 Dataset

In this paper, we use the lexical resources and dataset produced by [5]. This dataset consists of domain specific government smart apps lexical resources and dataset which were manually annotated and verified. According to the authors [5], the dataset consists of; short reviews in average, slangs, spelling mistakes, incorrect grammar, no spam reviews, and few occurrences of ineffective emoticons. This led to choosing an integrated lexicon and rule-based model.
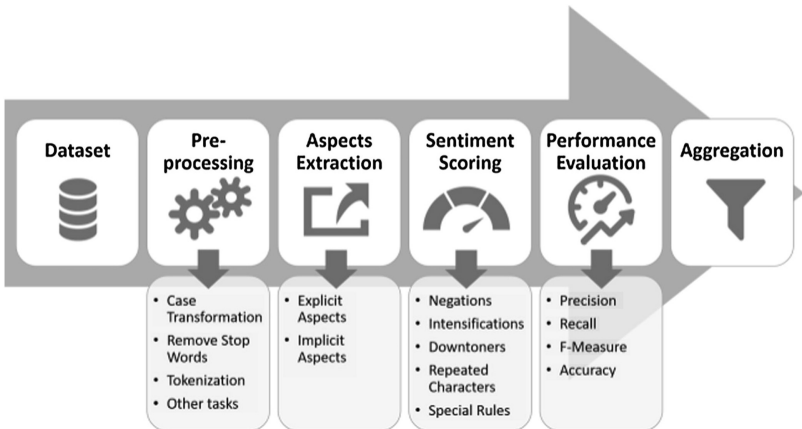

Figure 2.
Research Methodology.

The dataset was split into two subsets: the training and testing sets. The training set comprises 5141 (70%) of the reviews and the testing set comprises 2205 (30%) of the reviews. This provides a better indication of how well the proposed model will perform on unseen data compared to other approaches. Table 1 summarizes the dataset content statistics.

Aspect-based sentiment analysis

3.2 Pre-Processing

When adopting the rule-based approach with supportive lexicons, it is crucial to perform text pre-processing tasks. The first step in the proposed algorithm is to split the review into sentences based on punctuations that identify a sentence end, such as the full-stop, question mark and exclamation mark. This would have an important impact on linking the polarity score with the right aspect term without interfering with irrelevant sentences. In addition, the review subject is added as the first sentence of the review. Next, the sentences are tokenized where for each token, punctuations are removed and all letters are converted to lowercase. However, the pre-processing tasks do not perform normalization, such as removing repeated characters. The reason is that the aspect sentiment scoring phase, are responsible to treat it as intensification which affects the polarity score. For example, the proposed approach considers the word "*greaaat*" as "*very great*". Finally, stop words will be marked to be out of any of the following phases by using a customized list of stop words such as "*the*", "*an*" and "*of*". This list has been initiated by studying the domain and the reviews in the dataset.

3.3 Implicit and explicit aspect extraction

Aspects categories are vital for the aspect extraction task in sentiment analysis. To address this requirement in the government mobile apps domain, [5] have defined a set of aspects categories according to the written standards by Android, Apple and Smart Dubai Office [35–38]. The resulting aspects categories were User Interface, User Experience, Functionality and Performance, Security, and Support and Updates. These aspects categories are used in this study.

One of the main challenges in the aspect extraction task is that some aspects are not explicitly mentioned in a review. A sentence with explicit aspect basically contains a term that indicates the aspect category, while an implicit aspect would not be expressed by any specific word or term. For example, in a review with two sentences "*The app design is attractive. But it is buggy*", the first sentence contains an explicit term "design" with an opinion word "*attractive*" that indicates a positive sentiment toward the aspect category "*User Interface*". On the other hand, the second sentence does not contain a term explicitly

Number of reviews	7,345
Number of characters (No Spaces)	307,978
Number of characters (With Spaces)	366,797
Number of Words	67,777
Number of Explicit Aspects Terms	279
Number of Implicit Opinion-Aspects Terms	402
Number of Opinion Words	656
Average Words per Review	9.23
Number of Very Positive Reviews	2001
Number of Positive Reviews	2392
Number of Neutral Reviews	1435
Number of Negative Reviews	1015
Number of Very Negative Reviews	502

Table 1.
Dataset Content Statistics [5].

(Something like: "*The app functionalities are buggy*"). However, it contains an opinion word "*buggy*" which implicitly implies the aspect category "*Functionality and Performance*".

One of the approaches that are widely used in aspect identification is to consider opinion words as a good potential candidate for implicit aspect extraction [29]. Thus, the designed algorithm first looks for opinion words that directly denote aspect according to the lexicon. Otherwise, if the opinion word cannot determine the aspect category, the algorithm will search for the nearest aspect term in the same sentence with maximum window size of two with more priority to the right side, since the adjective usually occurs before the term. The pair of identified opinion word and aspect term will be looked up in the lexicon in order to determine the aspect category.

Function 1 illustrates the algorithm to extract the explicit and implicit aspects. This function returns two arrays where the first array (*aspectIndices*) represents the indices of the aspect terms in the review and the second array (*aspectCategories*) represents the as aspect categories to the corresponding aspect terms in the first array.

In the first sentence of the previous example "*The app design is attractive*", the algorithm will identify "*attractive*" as an opinion word that cannot determine the aspect as per the lexicon, so it will look for the nearest aspect term in the same sentence which is "*design*" and determine the "*User Interface*" as aspect category through the lexicon. Conversely, in the second sentence "*But it is buggy*", as per the lexicon, "*buggy*" implies an opinion toward the aspect category "*Functionality and Performance*" even though the sentence does not contain any explicit aspect term. Similar approach was also adopted by [39].

Function 1: Explicit and Implicit Aspect Extraction

```
Function 1 extractAspects (review) public
            returns (int[]aspectIndices, string[] aspectCategories)
    For each token t in review
        #Extract Explicit Aspects
        If (isToken_AspectTerm(t))
            aspectCategory =getAspectCategoryFromLexicon(t)
            aspectCategories.append(aspectCategory)
            aspectIndices.append(token_index)
        #Extract Implicit Aspects
        If (isToken_SentimentWord(t) AND
                getNearestAspect(token_index, aspectIndices[])==null)
            aspectCategory =getAspectCategoryFromLexicon(t)
            aspectCategories.append(aspectCategory)
            aspectIndices.append(token_index)
    End For
End Function
```

3.4 Aspect sentiment scoring

The approach that has been followed employs the populated lexicons produced by [5]. Basically, the algorithm as Function 2 represents, navigates through the sentences and once an opinion word is identified, its polarity score is retrieved through the lexicon and linked with the extracted aspect. In the experiment, we applied several settings to the algorithm in order to identify opinion words in a sentence in addition to the use of the lexicons. For instance, various rules are adopted to handle negations, intensification, downtoners, repeated characters, and the special case of negation-opinion rules. Finally, as

illustrated in Function 2, the function returns two arrays that store the polarity score in the first array (*polarities*) and the corresponding aspect category in the second array (*orderedAspectCategories*).

Function 2: Aspect Sentiment Scoring Algorithm

```
Function 2 scoreSentiment(review, int[]aspectIndices, string[] aspectCategories)
        public returns (float[] polarities, string[] orderedAspectCategories)
    For each token t in review
        If (isToken_SentimentWord(t))
            aspectCategory=getNearestAspect(token_index,
                        aspectIndices[], aspectCategories[])
            orderedAspectCategories.append(aspectCategory)
            polarity =getPolarityFromLexicon(t)
            polarity*=getNearestNegationFactor(token_index, review)
            polarity*=getNearestModifierFactor(token_index, review)
            polarities.append(polarity)
    End For
End Function
```

As a baseline, the basic lexicon approach is followed where a straightforward lookup returns the polarity score. However, the polarity score retrieved from the sentiment lexicon is not final. There are several rules that can affect the accuracy of the polarity score such as negation, intensification and downtoners [40]. For each of these modifiers' parameters, a lexicon that lists different terms that implies a modification has been constructed. For instance, the term "*not*" implies a negation and in case it occurs near an opinion word, as in the sentence "*The app design is not attractive*", the polarity score is inverted through multiplying it by (-1). This case is called "switch negation" as it reverses the polarity of the opinion word. However, there is another negation case in which the polarity is not inverted. For example, in the following review: "*The app design is not innovative*", the negation does not invert the polarity in this case since the strength of the opinion word "*innovative*" is highly positive. The algorithm will perform polarity shifting through multiplying it by certain threshold value (a fraction e.g. Threshold $= 0.5$) instead of polarity reversal. In shifting negation, the proposed approach identifies the highly positive opinion word score that are greater than (0.75) and highly negative opinion word score that are less than (-0.75). In the previous example, "innovative" polarity score as per the lexicon is (1.0) which is greater than (0.75), so this value will be shifted to (0.375). Table 2 shows examples for some of the negation words from the negation lexicon.

The proposed algorithm treats the intensifiers and downtoners which can increase or decrease the polarity score of an opinion word when they appear in the sentence. For example,

Negation Term	Example
Not	The app design is *not* attractive
Without	Works smooth *without* bugs. Great app.
Doesn't	App *doesn't* work. It always shows error in login.
Never	*Never* able to login. Once you put in login details… That's it…
Can't	You *can't* open it again… Crashes. …… Waste of time
None	Old design, *none* responsive and *non*-interactive
Non	

Table 2.
Samples that Show
Some of the
Negation Words.

the sentence "*The app design is <u>very</u> attractive*" implies more positive sentiment than the sentence "*The app design is attractive*", so "*very*" considered as intensifier. On the other hand, the sentence "*The app design is <u>quite</u> attractive*" implies less positive sentiment, so the word "*quite*" is considered as downtoner. Hence, the algorithm modifies the polarity score when it locates any of the intensifiers or downtoners that occurs before the opinion word according to the lexicon. However, not all intensifiers or downtoners have the same impact or power. For instance, the word "*most*" denotes more intensification than the word "*very*". Similar to [41], the proposed approach assigned a multiplication factor to each intensifier and downtoner. For example, the polarity score of the opinion word that follows the intensifier "*very*" is multiplied by the factor (1.25). On the other hand, the polarity score of the opinion word that follows the downtoner "*quite*" is multiplied by the factor (0.75). Table 3 shows several intensifiers and downtoners along with their factors and examples.

Modifications can appear in different forms and can be embedded in the opinion word itself. For example, the suffix "*est*" as in "the *greatest*" is also considered as intensifier and it is equal to the word "most". On the other hand, people like to exaggerate their feelings by repeating letters as in "*greaaaat*" or "*baaaaad*" which is somehow equal to "*very great*" or "*very bad*". [42] treated the repeated characters as intensification problem and suggested a solution that whenever two or more consequent characters are found to improve the word sentiment scoring for both positive and negative orientation. For example, in the following review: "*Baaaaaaad app*", assume that the score of "*bad*" is equal to "-2" then the sentiment score due to the repeated characters could be "-4". Hence, the algorithm identifies such cases by looking for repeated characters or specific suffixes and treats them as intensifiers. Table 4 provides the details of the complete set of intensifiers and downtoners with their factors.

3.5 Aspect sentiment aggregation
The algorithm targets to determine the star rating for different aspects extracted in the review. The five-star rating scale (1–5) is chosen in the experiment, where: a one-star expresses a very negative sentiment toward this aspect, two-star expresses a negative sentiment, three-star expresses a neutral sentiment, four-star expresses a positive sentiment, and five-star expresses a very positive sentiment. This can play a crucial role in understanding users' feedback toward specific aspects rather than a general feedback where the smart government apps owners can be aware of the areas of pains and gains of their customers.

Term	Type	Factor	Example
Extremely	Intensifier	2.0	Made me settle very quickly. *Extremely* reliable application *Extremely* slow loading and shows database connection error.
Absolutely	Intensifier	1.75	*Absolutely* useless app. The app is useless, never works properly. An *absolutely* useful and friendly application!
Quite	Downtoner	0.75	*Quite* useful app for Dubai residents. It's *quite* frustrating as I am unable to use the app Great interface. The app is *quite* slow but has good functionalities.
Pretty	Downtoner	0.50	*Pretty* good but some bugs Poorly designed app. *Pretty* much useless and probably just a media stunt.
Always	Intensifier	1.5	App doesn't work. It *always* shows error in login. New version *Always* crashes.

Table 3.
Sample Intensifiers and Downtoners from the Dataset.

Since the opinion words are extracted along with their final polarity scores and aspects, it would be a straightforward procedure to calculate the average of polarity scores for opinion words that are categorized under each aspect. Consider the following illustrative scenario of the user review: "*The app design is very attractive. It is organized and neat. But it is quite buggy and some features are missing*". The algorithm identifies the five sentiments shown in Table 5.

Function 3: Aspect Sentiment Aggregation Algorithm

```
Function 3 aggregateSentimentScores(float[] polarities,
string[] orderedAspectCategories) public returns (int starRating)
    #Assign initial values (0) to each aspect category in SentimentScoring array
    #Assign initial values (0) to each counter in SentimentCount array – This is
    needed to calculate the average
    float SentimentScoring['UserInterface', 'UserExperience', 'Functionality',
            'Security', 'Support'] = [0,0,0,0,0]
    float SentimentCount['UserInterface', 'UserExperience', 'Functionality',
            'Security', 'Support'] = [0,0,0,0,0]
    #Sum polarity scores together for each aspect category
    For each category c, polarity p in orderedAspectCategories, polarities
        Switch c:
            Case 'UserInterface':     SentimentScoring['UserInterface'] +=p
                                      SentimentCount['UserInterface'] +=1
            Case 'UserExperience':    SentimentScoring['UserExperience']+=p
                                      SentimentCount['UserExperience'] +=1
            Case 'Functionality':     SentimentScoring['Functionality']+=p
                                      SentimentCount['Functionality'] +=1
            Case 'Security':          SentimentScoring['Security']+=p
                                      SentimentCount['Security'] +=1
            Case 'Support':           SentimentScoring['Support']+=p
                                      SentimentCount['Support'] +=1
        End Switch
    End For
    #Calculate the average score for each aspect category. Then based on the
    result range, determine the star rating.
    For each categoryScore cs, categoryCount cc
            in SentimentScoring, SentimentCount
        cs=cs / cc
        Switch cs:
            Case cs >= 0.75: starRating=5
            Case cs >= 0.25: starRating=4
            Case cs > -0.25: starRating=3
            Case cs <= -0.25: starRating=2
            Case cs <= -0.75: starRating=1
        End Switch
    End For
End Function
```

Function 3 illustrates the algorithm that handles the aggregation task as explained in the previous example and returns the final star rating of a review. The final stage of the aspect-based sentiment analysis is to use data visualization to summarize the full set of reviews in a graphical representation that shows the equivalent star rating of each aspect based on five stars' scale that are returned from Function 3.

4. Discussion of results

As explained in the previous sections, several attempts were conducted to evaluate the algorithm improvements on both aspects' extraction and opinion classification. The experiments evaluation for aspect extraction and sentiment were carried out according to the several settings and parameters. Standard confusion matrix is employed to measure the performance on the unseen testing set, which comprises 2205 reviews (30%) of the overall reviews (7345). This can help in calculating more advanced evaluation metrics such as Precision, Recall, F-measure and Accuracy metrics. The confusion matrix elements for both aspect extraction and sentiment classification are defined as illustrated in Tables 6 and 7. In addition, the pearson product-moment correlation coefficients was adopted to examine the correlation between the predicted values the actual values of the sentiment classification task. This metric returns a value between -1 and $+1$. Whereas, higher positive value denotes a positive relationship between the predicted and the actual values. On the other hand, lower

Term	Type	Factor	Term	Type	Factor
Very	I	1.25	too	I	1.25
Highly	I	1.75	absolutely	I	1.75
Really	D	0.75	completely	I	1.5
More	D	0.50	most	I	1.75
Always	I	1.5	mostly	D	0.75
Many	I	1.25	pretty	D	0.75
Fully	I	1.5	somewhat	D	0.5
So	I	1.25	slightly	D	0.75
Lot	I	1.5	extraordinarily	I	2.0
Full	I	1.5	ever	I	1.75
extremely	I	2.0	still	D	0.75
totally	I	1.25	just	D	0.75
little	D	0.25	barely	D	0.25
slightly	D	0.5	hardly	D	0.25

Table 4.
The set of Intensifiers (I) and Downtoners (D) with their Factors.

#	Opinion	Aspect	Score	Modified By	Final Polarity	Star Rating
1	Attractive	User Interface	0.69	1.25	0.86	4 Stars
2	Organized	User Interface	0.47	–	0.47	
3	Neat	User Interface	0.36	–	0.36	
4	Buggy	Functionality and Performance	−0.5	0.75	−0.375	2 Stars
5	Missing	Functionality and Performance	−0.56	–	−0.56	

The aggregation task groups the polarities by the aspect categories and calculates the average. Here there are two aspect categories:

1. User interface: (0.86 + 0.47 + 0.36)/3 = 0.56. This is considered as a 4-star rating in terms of "User Interface".
2. Functionality and Performance: (−0.375 + −0.56)/2 = −0.47. This is considered as a 2-star rating in terms of "Functionality and Performance".

Table 5.
The Extracted Aspects and Corresponding Sentiment Scores.

negative value denotes that the relationship between the predicted and actual values are negatively correlated and that means if one value increases, the other value decreases. Correlation value close to zero refers to the situation where there is no relationship between the predicted and actual values.

The corresponding performance evaluation measures are calculated according to the following formulas:

$$Precision = \frac{TP}{(TP + FP)} \qquad (1)$$

$$Recall = \frac{TP}{(TP + FN)} \qquad (2)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (3)$$

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FP)} \qquad (4)$$

The aspect extraction results include a comparison of experiments that were conducted in this study. As Likewise, several parameter settings have been employed to measure the progression of the aspect's sentiment classification. As stated in Section 3.5, the basic lexicon (L) based approach is considered as a baseline for the performance evaluation. Moreover, other rules settings are added to measure the improvements on the performance measures. These rules settings are adopted to handle numerous challenges in sentiment analysis such as handling negation (N), intensification (I), downtoners (D), repeated characters (R) and special cases of negation-opinion rules (S). As shown in Table 10, the integrated lexicon and the various rules $(L + N + I + D + R + S)$ have achieved the highest performance scores with respect to aspect extraction and opinion classification.

		Predicted (Proposed Aspect Extraction Approach)	
		Retrieved	Not Retrieved
Actual (Annotated Dataset)	Relevant	TP (true positive) Number of aspects that are correctly extracted	FP (false positive) Number of aspects that are annotated, but the not extracted by the algorithm
	Irrelevant	FN (false negative) Number of aspects that are not annotated, but extracted by the algorithm	TN (true negative) Number of aspects that are not annotated and not extracted by the algorithm

Table 6. Aspect Extraction Confusion Matrix.

		Predicted (Proposed Sentiment Classification Approach)	
		Retrieved	Not Retrieved
Actual (Annotated Dataset)	Relevant	TP Number of sentiment polarity scores that are calculated correctly by the algorithm	FP Number of sentiment polarity scores that are incorrectly calculated by the algorithm
	Irrelevant	FN Number of aspects that does not have sentiment assigned, but calculated by the algorithm	TN Number of aspects that does not have sentiment assigned and not calculated by the algorithm

Table 7. Sentiment Classification Confusion Matrix.

Table 9 shows, there is a significant increment in the evaluation results when considering implicit aspects as well as explicit ones. While analyzing the algorithm progress in identifying the sentiments toward aspects without taking the implicit aspects into consideration, the algorithm was still able to identify opinion words and attempt to search for surrounding terms to recognize the aspect. Thus, it succeeds in aspect extraction in some cases, especially when there is another opinion word with explicit aspect in the same sentence. However, in many cases, a misleading aspect term appeared nearby the identified opinion word, where in this case the sentiment has been assigned to a false aspect and the correct aspect has not been extracted, as the sample reviews show in Table 8. This may increase both FP and FN values and as a result it will affect both Precision and Recall.

Likewise, several parameter settings have been employed to measure the progression of the aspect's sentiment classification. As stated in Section 3.5, the basic lexicon (L) based approach is considered as a baseline for the performance evaluation. Moreover, other rules settings are added to measure the improvements on the performance measures. These rules settings are adopted to handle numerous challenges in sentiment analysis such as handling negation (N), intensification (I), downtoners (D), repeated characters (R) and special cases of negation-opinion rules (S). As shown in Table 10, the integrated lexicon and the various rules (L + N + I + D + R + S) have achieved the highest performance scores with respect to aspect extraction and opinion classification.

#	Review	Aspect Term	Opinion Word	Aspect Category Aspect Extracted As	The Correct Aspect
1	After *update*, it seems to me that it is *good*	Update	Good	User Experience	Support and Update
2	Before reaching to the *payment* it is fine. After that it is not *working*	Payment	Working	User Experience	Security
3	I *installed* this app several times still not *working*	Install	Working	User Experience	Support and Update

Table 8.
Sample Misclassified Reviews.

	Precision	Recall	F-Measure	Accuracy
Explicit Aspects	75.49%	70.64%	72.98%	86.83%
Explicit & Implicit Aspects	92.63%	84.03%	88.12%	93.01%

Table 9.
Aspect Extraction Performance Evaluation Results.

		Precision	Recall	F-Measure	Accuracy	Correlation
Parameters	L (Baseline)	73.66%	60.98%	66.72%	83.17%	0.65
	L + N	80.43%	65.84%	72.41%	85.89%	0.68
	L + N + I	83.42%	69.43%	75.79%	87.37%	0.74
	L + N + I + D	85.21%	72.15%	78.14%	88.57%	0.74
	L + N + I + D + R	86.17%	72.95%	79.01%	88.97%	0.79
	L + N + I + D + R + S	92.25%	79.59%	85.46%	91.64%	0.81
	Gini Index with SVM [18]	80.60%	75.44%	77.93%	86.64%	0.73
	Feature-based using SVM [17]	66.29%	61.93%	64.04%	82.45%	0.66
	Lexicon-based SVM	84.81%	72.19%	77.99%	88.41%	0.77

Table 10.
Aspect-based Sentiment Classification Performance Evaluation.

Furthermore, to compare the performance of our proposed approach with other techniques, the feature-based approach for aspect-based sentiment analysis using SVM proposed by [17] and the Gini Impurity Index with SVM by [18] was adopted. In this vein, the supervised machine learning approach by [17] has been replicated to extract aspects and classify the corresponding sentiments from the mobile government apps reviews. First, the SentiWordNet was used to identify and remove the objective sentences in each review. Second, the aspect terms were extracted using NNS, NN, and NNP POS tags to identify the most and least frequently used aspects by the users. The most frequently used aspects were defined based on a pre-defined frequency threshold ($n = 10$). These aspects were considered in the experiment. Conversely, the least frequently used aspects were discarded. Third, Stanford parser was used to extract the opinion words and to find the grammatical dependencies to determine the connection between the opinion words and the extracted aspects. Finally, the feature sets that comprises the aspects, opinion words and polarity scores were inducted into the SVM classifier model. RapidMiner was used to train the SVM model on the training set in order to allow the classifier to gain the domain knowledge. Further, the model was evaluated based on the same unseen testing set used in our proposed integrated rule-based model. The result of the SVM classifier is a set of vectors that contain aspects and its opinion words for each review. The vector can be represented as follows:

$$V_k(R_n, F_i, O_{ij}, P_{ij}, N_{ij})$$

where, V_k is the kth vector, R_n is the nth review, F_i is the ith feature in R_n, O_{ij} is the jth opinion word for i^{th} feature, P_{ij} is the positive score for j^{th} opinion word, and N_{ij} is the negative score for j^{th} opinion word.

Additionally, we used RapidMiner to replicate the proposed model by [18] with minor modifications for further evaluation. The TF/IDF was used as a weighting technique. TF/IDF is among the most regular and widely used term weighting techniques. Apart from TF/IDF fame, it has been considered as an empirical approach which is a frequency-based approach that considers the words and document frequencies. The TF/IDF is a well-known model with its proven performance and efficiency [43]. But, this method considers the words of low frequency as important and the words with high frequency as unimportant which impacts the precision of the classifier negatively [18]. To overcome this issue, the weightage has been assigned to the attributed using the Gini Impurity Index. This index calculates the weight of attributes with respect to the label attribute by computing the Gini Index of the class distribution. This is done by adding the "Weight by Gini Index" operator after "Process Documents" operator that uses the TF/IDF vector creation method parameter. To evaluation the performance, the "Cross Validation" operator has been used with 10 folds along using the SVM classifier. Cross-validation is an approach to evaluate SVM models by dividing the dataset into a training set and a testing set. In 10-fold cross-validation, the original dataset is randomly divided into 10 equal subsets. Out of these 10 subsets, a single subset is retained as testing the model, and the remaining 9 subsets are used as training data. The cross-validation process is then repeated 10 times, with each of the 10 subsets used exactly once as the testing sets. The average results of the 10 folds are calculated.

To measure the impact of domain lexicons and rules on the performance of SVM and provide a complete analysis and discussion, these lexicons and rules are incorporated as the input features to represent the reviews. The input for the SVM is represented as set of rows that denote the interpretation for each review, where each row is represented according to the following vector:

$$V_i(N_{i1}, I_{i1}, D_{i1}, R_{i1}, O_{i1}, \cdots, N_{im}, I_{im}, D_{im}, R_{im}, O_{im})$$

From the above representation, each feature (N, I, D, R, O) is indexed using two variables. The first variable represents the review identifier and the second variable represents the word index in the review. Whereas, m denotes the maximum words count in a review in the given dataset. As illustrated in Table 10, the results show that the achieved accuracy is higher than other SVM models. This underlines that fact that the input dataset has distinct lexicon features which can help in the classification process.

According to the results that are illustrated in Table 10, the proposed integrated lexicon and rule-based approach with all combined rules has significantly outperformed both the feature-based aspect-based sentiment analysis using SVM and the aspect terms extraction using Gini Index and SVM classifier. One of the main reasons for these results is that the feature-based aspects-based sentiment analysis using SVM approach does not consider the implicit aspects. Furthermore, this approach is dependent on structured sentences for accurate POS tagging and parsing. On the other hand, the aspect term extraction using Gini Index with SVM achieved higher performance than the feature-based aspects-based sentiment analysis using SVM approach since it does not look into the structure of the reviews and only relies on the weightage of the terms. However, by adopting the lexical features as inputs for the SVM, the model was able to achieve higher performance than other SVM models and slightly lower performance than the integrated lexicon and rule-based model. With respect to the pearson product-moment correlation coefficients, as illustrated in Table 10, the usage of various lexical rules significantly impacts the correlation value. The highest correlation value was achieved when using the combined rules (L + N + I + D + R + S) with a score of 0.81.

5. Implications
Public's opinion is very important to increase the usage of the smart government mobile apps. Government agencies are looking for customer centric approaches to better understand their consumers' opinions toward their services. The reviews represent a main source for apps owners to understand the user's perspectives and hence, provide services that are up to their customers' expectations. Moreover, people are keen to look at others' opinions before downloading and using any smart app.

Research has made significant progress in regards of natural language processing albeit the difficulties faced in such tasks [44]. There is a huge need in the industry for sentiment analysis services as every business wants to know the opinions of their customers. With such needs, research in the field of sentiment analysis and opinion mining will stay active for years to come [4].

One may argue that current methods and algorithms are sufficient to provide unsupervised solutions for different domains datasets. However, a fully automated and accurate solution that can be applied to all domains does not yet exist. However, it is possible to develop effective semi-supervised solutions were domain specific aspects are manually annotated while other tasks are automated. Moreover, cross-domain sentiment analysis can be more effectively utilized to reduce the manual efforts in labelling data. In this case, machine learns from a particular domain and applies the knowledge to analyse the sentiment of texts in another domain [45].

Therefore, despite our study being domain dependent, the methods applied in this research can extend their applicability to other domains concerned about public and private services. Moreover, the methods proposed in this research can be applied in the social media where the textual posts are similar in nature and structure.

6. Conclusion
Aspect-based sentiment analysis is considered as one of the challenging tasks in sentiment analysis area of research. It is important that all feedbacks are understood and categorized so

that smart governments can rely on this channel to listen to their customers. Therefore, this can be considered as a factor for future smart services improvements and optimizations that exceed the people's expectations. In this regard, an integrated lexicon and rule-based approach was employed to extract explicit and implicit aspect as well as sentiment classification for these aspects. In this study, an integrated lexicon and rule-based model has been chosen. This model utilized the manually generated lexicons in this study with hybrid rules to handle some of the key challenges in aspect-based sentiment analysis in particular and sentiment analysis in general. This approach reported high performance results through an integrated lexicon and rule-based model. The approach confirmed that integrating sentiment and aspects lexicons with various rules settings that handle various challenges in sentiment analysis, such as handling negation, intensification, downtoners, repeated characters, and special cases of negation-opinion rules, outperformed the lexicon baseline and other rules combinations.

The dataset contains a substantial number of comparable sentences. This can be considered as additional rules that can extend the proposed model. [46] pointed out that users may express their opinions by comparing several entities together. These comparative rules are considered as a significant source for determining the strengths and weaknesses of the government mobile apps and highlighting the future enhancements and improvement areas. Another area that can be applied on such dataset, is to identify the improvement areas and innovative ideas through analyzing the customer's reviews which may include suggestions, complaints, problems, bugs, comparisons among others.

Government mobile apps owners should take a step forward and think how to gather their customer's reviews and feedbacks. One quick and easy way is to prompt the users for providing a review within the mobile app. Also, a rewarding program could be beneficial to increase the customers' reviews towards this particular application.

References

[1] S. Coleman, J. Gøtze, Bowling together: Online public engagement in policy deliberation, in: Bowling Together: Online Public Engagement in Policy Deliberation, Hansard Society, London, 2001, pp. 39–50.

[2] UAE mGovernment, Guidelines for eParticipation and Social Media for Government Entities, (2016). https://www.government.ae/en/resources/guidelines (accessed 19.08.2019).

[3] M. Anshari, S.A. Lim, E-government with big data enabled through smartphone for public services: possibilities and challenges, Int. J. Public Admin. 40 (2017) 1143–1158, http://dx.doi.org/10.1080/01900692.2016.1242619.

[4] B. Liu, Sentiment analysis and opinion mining, Synthesis Lectures Human Lang. Technol. 5 (2012) 1–167, http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016.

[5] O. Alqaryouti, N. Siyam, K. Shaalan, A Sentiment analysis lexical resource and dataset for government smart apps domain, in: A. Hassanien, M. Tolba, K. Shaalan (Eds.), Proceedings of the International Conference on Advanced Intelligent Systems and Informatics on Advanced Intelligent Systems and Informatics 2018 (AISI2018), Springer, Cham, 2019.

[6] S. Poria, E. Cambria, G. Winterstein, G.-B. Huang, Sentic patterns: Dependency-based rules for concept-level sentiment analysis, Knowl.-Based Syst. 69 (2014) 45–63, http://dx.doi.org/10.1016/j.knosys.2014.05.005.

[7] S. Poria, E. Cambria, A. Gelbukh, F. Bisio, A. Hussain, Sentiment data flow analysis by means of dynamic linguistic patterns, IEEE Comput. Intell. Mag. 10 (2015) 26–36, http://dx.doi.org/10.1109/MCI.2015.2471215.

[8] M.Z. Asghar, A. Khan, S. Ahmad, F.M. Kundi, A review of feature extraction in sentiment analysis, J. Basic Appl. Scientific Res. 4 (2014) 181–186.

[9] S.Y. Ganeshbhai, Feature based opinion mining : a survey, in: Advance Computing Conference (IACC), 2015 IEEE International, IEEE, 2015, pp. 919–923.

[10] M.S. Akhtar, D. Gupta, A. Ekbal, P. Bhattacharyya, Feature selection and ensemble construction: a two-step method for aspect based sentiment analysis, Knowl.-Based Syst. 125 (2017) 116–135, http://dx.doi.org/10.1016/j.knosys.2017.03.020.

[11] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2016) 102–107, http://dx.doi.org/10.1109/MIS.2016.31.

[12] G. Carenini, J.C.K. Cheung, A. Pauls, Multi-document summarization of evaluative text, Comput. Intell. 29 (2013) 305–312.

[13] B. Liu, M. Hu, J. Cheng, Opinion observer: analyzing and comparing opinions on the web, in: Proceedings of the 14th International Conference on World Wide Web, 2005, pp. 342–351.

[14] K. Schouten, O. van der Weijde, F. Frasincar, R. Dekker, Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data, IEEE Trans. Cybern. 48 (2018) 1263–1275, http://dx.doi.org/10.1109/TCYB.2017.2688801.

[15] B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in: Mining Text Data, Springer US, 2012, pp. 415–463.

[16] A.K. Samha, Y. Li, J. Zhang, Aspect-based opinion extraction from customer reviews, in: Computer Science and Information Technology (CS and IT), Volume 4, Number 4: Proceedings of the Second International Conference of Database and Data Mining (DBDM 2014), AIRCC Publishing Corporation, 2014, pp. 149–160.

[17] D.V.N. Devi, C.K. Kumar, S. Prasad, A feature based approach for sentiment analysis by using support vector machine, in: 2016 IEEE 6th International Conference on Advanced Computing (IACC), IEEE, 2016, pp. 3–8, http://dx.doi.org/10.1109/IACC.2016.11.

[18] A.S. Manek, P.D. Shenoy, M.C. Mohan, K.R. Venugopal, Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier, World Wide Web. 20 (2017) 135–154, http://dx.doi.org/10.1007/s11280-015-0381-x.

[19] M.S. Mubarok, M. Adiwijiaya, Aldhi Dwi, Aspect-based sentiment analysis to review products using Naïve bayes, in: AIP Conference Proceedings, AIP Publishing, 2017, pp. 1–8, http://dx.doi.org/10.1063/1.4994463.

[20] M. Al-smadi, O. Qawasmeh, M. Al-ayyoub, Y. Jararweh, B. Gupta, Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews, J. Comput. Sci. (2017) 386–393, http://dx.doi.org/10.1016/j.jocs.2017.11.006.

[21] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, Y. Jararweh, Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews, Int. J. Mach. Learn. Cybern. (2018) 1–13, http://dx.doi.org/10.1007/s13042-018-0799-4.

[22] M. Al-ayyoub, M. Al-smadi, M. Al-ayyoub, Y. Jararweh, O. Qawasmeh, Enhancing aspect-based sentiment analysis of Arabic hotel' reviews using morphological, syntactic and semantic features, Inf. Process. Manage. (2018) 308–319, http://dx.doi.org/10.1016/j.ipm.2018.01.006.

[23] A. García-Pablos, M. Cuadros, G. Rigau, W2VLDA: almost unsupervised system for aspect based sentiment analysis, Expert Syst. Appl. 91 (2018) 127–137.

[24] M. Dragoni, M. Federici, A. Rexha, An unsupervised aspect extraction strategy for monitoring real-time reviews stream, Inf. Process. Manage. (2018) 1103–1118.

[25] M. Rathan, V.R. Hulipalled, K.R. Venugopal, L.M. Patnaik, Consumer insight mining: aspect based Twitter opinion mining of mobile phone reviews, Appl. Soft Comput. J. 68 (2018) 765–773, http://dx.doi.org/10.1016/j.asoc.2017.07.056.

[26] E. Cambria, S. Poria, D. Hazarika, K. Kwok, SenticNet 5: Discovering Conceptual Primitives for Sentiment Analysis by Means of Context Embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 1795–1802. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16839 (accessed 26.10.2019).

[27] F. Xianghua, L. Guo, G. Yanyan, W. Zhiqiang, Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon, Knowl.-Based Syst. 37 (2013) 186–195, http://dx.doi.org/10.1016/j.knosys.2012.08.003.

[28] Z. Chen, A. Mukherjee, B. Liu, Aspect Extraction with Automated Prior Knowledge Learning, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014: pp. 347–358.

[29] S. Poria, E. Cambria, L.-W. Ku, C. Gui, A. Gelbukh, A rule-based approach to aspect extraction from product reviews, in: Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP), 2014, pp. 28–37.

[30] D. Pham, A. Le, Data & knowledge engineering learning multiple layers of knowledge representation for aspect based sentiment analysis, Data Knowl. Eng. 114 (2018) 26–39, http://dx.doi.org/10.1016/j.datak.2017.06.001.

[31] Q. Liu, Z. Gao, B. Liu, Y. Zhang, Automated rule selection for aspect extraction in opinion mining, in: IJCAI, 2015: pp. 1291–1297.

[32] Y. Ma, H. Peng, E. Cambria, Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018: pp. 5876–5883. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16541 (accessed 14.09.2019).

[33] S. Mukherjee, P. Bhattacharyya, Feature specific sentiment analysis for product reviews, in: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, Springer, Berlin Heidelberg, 2012, pp. 475–487.

[34] K. Shaalan, Rule-based approach in arabic natural language processing, Int. J. Inf. Commun. Technol. 3 (2010) 11–19.

[35] Apple, App Store Review Guidelines – Apple Developer, Developer.Apple.Com. (2017). https://developer.apple.com/app-store/review/guidelines/ (accessed 03.01.2017).

[36] Andriod, Core app quality | Android Developers, Developer.Android.Com. (2017). https://developer.android.com/develop/quality-guidelines/core-app-quality.html (accessed 15.01.2017).

[37] Smart Dubai Government, Smart Websites Excellence Model Version 4.0, Smart Dubai Government. (2016). http://dsg.gov.ae/en/OurPublications/Pages/StandardsGuidelines.aspx (accessed 03.01.2017).

[38] Apple, Themes – Overview – iOS Human Interface Guidelines, Developer.Apple.Com. (2017). https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/ (accessed 05.01.2017).

[39] J.R. Gil-garcia, N. Helbig, A. Ojo, Being smart: Emerging technologies and innovation in the public sector, Government Inf. Q. 31 (2014) I1–I8, http://dx.doi.org/10.1016/j.giq.2014.09.001.

[40] D. Vilares, C. Gómez-Rodríguez, M.A. Alonso, Universal, unsupervised (rule-based), uncovered sentiment analysis, Knowl.-Based Syst. 118 (2017) 45–55.

[41] M. Taboada, B. Julian, T. Milan, V. Kimberly, S. Manfred, Lexicon-based methods for sentiment analysis, Comput. Linguistics 37 (2011) 267–307.

[42] F.M. Kundi, A. Khan, S. Ahmad, M.Z. Asghar, Lexicon-based sentiment analysis in the social web, J. Basic Appl. Scientific Res. 4 (2014) 238–248.

[43] O. Alqaryouti, H. Khwileh, T. Farouk, A. Nabhan, K. Shaalan, Graph-based keyword extraction, 740 (2018) 159–172. doi:10.1007/978-3-319-67056-0_9.

[44] W. Zhao, H. Peng, S. Eger, E. Cambria, M. Yang, Towards Scalable and Reliable Capsule Networks for Challenging NLP Applications, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019: pp. 1549–1559. doi:10.18653/v1/P19-1150.

[45] R. Varghese, J. M, A survey on sentiment analysis and opinion mining, 2013.

ACI

[46] K. Xu, S. Shaoyi, J. Li, Y. Song, Mining comparative opinions from customer reviews for Competitive Intelligence, Decis. Support Syst. 50 (2011) 743–754, http://dx.doi.org/10.1016/j.dss.2010.08.021.

Corresponding author
Omar Alqaryouti can be contacted at: omar.alqaryouti@gmail.com