Hierarchical Classifiers for Multi-Way Sentiment Analysis of Arabic Reviews

Mahmoud Al-Ayyoub and Aya Nuseir
Jordan University of Science and Technology
Irbid, Jordan

Ghassan Kanaan and Riyad Al-Shalabi
Amman Arab University
Amman, Jordan

***Abstract*—Sentiment Analysis (SA) is one of hottest fields in data mining (DM) and natural language processing (NLP). The goal of SA is to extract the sentiment conveyed in a certain text based on its content. While most current works focus on the simple problem of determining whether the sentiment is positive or negative, Multi-Way Sentiment Analysis (MWSA) focuses on sentiments conveyed through a rating or scoring system (e.g., a 5-star scoring system). In such scoring systems, the sentiments conveyed in two reviews of close scores (such as 4 stars and 5 stars) can be very similar creating an added challenge compared to traditional SA. One intuitive way of handling this challenge is via a divide-and-conquer approach where the MWSA problem is divided into a set of sub-problems allowing the use of customized classifiers to differentiate between reviews of close scores. A hierarchical classification structure can be used with this approach where each node represents a different classification sub-problem and the decision from it may lead to the invocation of another classifier. In this work, we show how the use of this divide-and-conquer hierarchical structure of classifiers can generate better results than the use of existing flat classifiers for the MWSA problem. We focus on the Arabic language for many reasons such as the importance of this language and the scarcity of prior works and available tools for it. To the best of our knowledge, very few papers have been published on MWSA of Arabic reviews. One notable work is that of Ali and Atiya, in which the authors collected a large scale Arabic Book Reviews (LABR) dataset and made it publicly available. Unfortunately, the baseline experiments on this dataset had very low accuracy. We present two different hierarchical structures and compare their accuracies with the flat structure using different core classifiers. The comparison is based on standard accuracy measures such as precision and recall in addition to using the mean squared error (MSE) as a more accurate measure given the fact that not all misclassifications are the same. The results show that, in general, hierarchical classifiers give significant improvements (of more than 50% in certain cases) over flat classifiers.**

***Keywords*—multi-way sentiment analysis, hierarchical classifiers, support vector machine, decision tree, naive bayes, k-nearest neighbor, mean squared error**

I. Introduction

In the last decade, the number of Internet users has increased significantly. This increase can be seen as a result of the technologies that facilitated the widespread of the Internet, along with the various services provided through the Internet. These services includes social networking (Facebook, Twitter, etc.), publications (news, books, etc.) and other day-to-day services. The exposure of people to these online services allowed them to express their feelings and emotions regarding the provided services or in reaction to some subject in their lives. Furthermore, organizations of various types utilized the Internet to allow them to collect people's opinions about almost all the subjects the concern them through easing the process of getting feedback or by collecting what people are feeling from the various public websites. After the collection of the raw unstructured data containing these expressions, some processing must be performed to analyze the people sentiments. As a result, the interdisciplinary Sentiment Analysis field has emerged.

Sentiment Analysis (SA), also known as Opinion Mining (OM), refers to the use of natural language processing, text analysis and computational linguistics to identify and extract the sentiment orientation of textual materials.[1] The extraction of a sentiment can be made either on a whole document (document-level SA), on each paragraph (paragraph-level SA), or on each sentence (sentence-level SA) [37]. The considered sentiment orientations are usually assumed to simply be positive and negative only; making SA a binary classification problem. Some researchers, however, add more classes for neutral or conflicted sentiments. Note that this is different from the more general problem of emotion analysis, where the authors are interested in identifying more complex emotions such as joy, fear, etc. [11], [20].

The reason behind the immense interest in SA is because obtaining truthful information about the opinions of the stakeholders is a crucial point in any decision making process [45]. The authors of [37], [21] list several examples such as a company's use of SA tools to obtain a true indicator of its customers' satisfaction with its products and services. It can plan ahead according to such feedback to guarantee wider acceptance and larger market share. Another example of SA application is as a quicker and more accurate alternative of public polls. Instead of relying on public polls with all of their problems and expenses, government can measure the public's opinion by simply crawling through what is written on social networks and evaluate it using SA tools. Finally, recommender systems can benefit greatly from efficient and effective SA tools.

Most of the available SA tools can be categorized into one of two approaches: the corpus-based (supervised) approach and the lexicon-based (unsupervised) approach [38]. The corpus-based approach simply views SA as being a special case of text classification in which the classes are simply the sentiment

[1] http://en.wikipedia.org/wiki/Sentiment_analysis

orientations. A large dataset of manually annotated examples is used to train the classifier and testing techniques such as cross validation are used to evaluate the performance of the classifier. On the other hand, the lexicon-based approach, as its name implies, utilizes a lexicon composed of terms along with their sentiment values. To determine the sentiment value of certain text, the lexicon-based approach searches through the lexicon for the sentiment values of the terms composing the text and combines them. Combining these two approaches resulted in a hybrid approach known as the weakly-supervised approach [25].

Each approach has its pros and cons. Compared to the effort required by the lexicon-based approach, the corpus-based approach is considered very expensive due to the need for a large annotated dataset. In [46], with the variation of the topics, domains and time-periods, the corpus-based approach has the advantage of higher accuracy in SA. The focus of this work is on the corpus-based approach.

Many of the current works on SA consider the simple binary (or ternary) setting. However, there are few works that consider sentiment orientations based on some scoring or rating systems, in what is known as the Multi-Way Sentiment Analysis (MWSA) problem. People might shy away from working on this problem due to the challenges associated with it despite its apparent importance and strong association with recommender systems.

Having more classes is not the only additional challenge imposed by MWSA. The obvious difficulty of MWSA does not only come from considering more sentiment orientations, it is the relationship between these sentiment orientations that makes things difficult. To fully understand the effect of this issue, consider the settings of this work. This work focuses on the problem of automatically determining the rating of an Arabic review on a scale from 1 to 5. In a 5 star rating system, both 4 and 5 stars are considered positive sentiment orientations, while 1 and 2 stars are considered negative ones, and 3 stars as a neutral orientation. One of the challenges of this settings is the difficulty in distinguishing between the two positive ratings. It is much easier to tell whether a review is positive or not compared to telling whether it is a strong positive or a weak one. Distinguishing between the two negative ratings is equally hard. So, it is natural to decompose the MWSA problem into a set of sub-problems and employ a hierarchical classification structure in which an optimized classifier is devised to address each sub-problem separately. For example, one classifiers can be trained to distinguish between positive and negative reviews while another classifier is trained independently to distinguish between strong and weak positive reviews. Similarly a third classifier is trained independently to distinguish between strong and weak negative reviews. This gives the intuition that such an inherently hierarchical problem cannot be effectively addressed using a flat approach. The hierarchical classification approach has already been shown to be very useful for MWSA of English reviews [22]. Our goal is to investigate the effectiveness of this approach for Arabic reviews.

Most studies on SA in general and specifically on MWSA have been conducted on the English language with few considering other languages. This might be due to having large datasets publicly available for the English language. In this project, we consider the Arabic language for applying MWSA. Arabic is the language of 22 countries with more than 400 million inhabitants. Natural Language Processing for the Arabic language is considered challenging due to the special characteristics of the language such as: orthography, the existence of short vowels, the complex morphology compared to English, the widespread of synonyms and the lack of publicly and freely accessible corpora [29], [35], [10].

To the best of our knowledge, very few papers have been published on the MWSA problem. One notable work is that of Ali and Atiya [21], in which the authors collected a large scale Arabic Book Reviews (LABR) dataset and made it publicly available. Unfortunately, the baseline experiments on this dataset had very low accuracy. Motivated by the intuition that employing a flat classifier to handle an inherently hierarchical problem such as MWSA is one of the main reasons behind such poor accuracy, we propose to use hierarchical classification. We present two different hierarchical structures and compare their accuracy with the flat structure using different core classifiers. The comparison is based on standard accuracy measures such as precision and recall in addition to using the mean squared error (MSE) as a more accurate measure given the fact that not all misclassifications are the same.

The rest of this paper is structured as follows. Section II presents the related works. In Section III, we present our system model and evaluate it in Section IV. Finally, we conclude in Section V.

II. Background and Related Works

The problem at hand is the MWSA of Arabic reviews using hierarchical classification. The following coverage of the literature focuses on similar works on the English language before discussing the existing works on SA and MWSA of Arabic text.

The word hierarchical classification has appeared in several contexts such as: hierarchical labeling (in which the labels are structured into a hierarchy of classes and subclasses), hierarchical classifier (in which the classifiers themselves are structured in a hierarchy), ensemble methods, One-versus-All (OVA), One-versus-One (OVO), etc. Each one of these concepts is investigated by many researchers to study its applicability and effect on the classification processes. The focus of this work is on hierarchical classifiers. Using the divide and conquer mentality, a hierarchical classifier breaks the classification problem into several sub-problems and attacks each sub-problem according to a tree or a Directed Acyclic Graph (DAG) hierarchy [34].

Most of the available researches are based on flat classification. However, the exploding number of online data makes the use of flat classification methods more difficult giving rise to the concept of hierarchical classification. The hierarchical classification is based on divide-and-conquer principle, where the problem can be divided into sub problems and easily can be solved.

The most commonly used structure in hierarchical classification is the tree-like structure, however, many researchers have proposed using Directed Acyclic Graphs (DAG). One difference between the different structures proposed in the

literature is whether a node can have more than one parent or not. Another difference is whether the class is given in leaf nodes only or in leaf as well as internal nodes [55], [52].

In [32], the authors developed two types of hierarchies for emotion classification using SVM classifier. The dataset consisted of six emotional classes (happiness, sadness, fear, anger, disgust, and surprise) in addition to the no-emotion class which contain instances conveying no feelings or emotions. The first type of hierarchy is a two-level classification hierarchy where the first level tests whether the instance is emotional or non-emotional, then the second level takes instances that are classified as emotional and classify them into one of the six emotional classes. The second type is a three-level classification where the first level is the same as first level of the two-level type, the second level classifies the emotional instances into positive (happiness) or negative, and the last level classifies negative instances into one of the five classes (sadness, fear, anger, disgust, and surprise).

The authors of [22] proposed a hierarchical classifier tree (MCST) consisting of standard binary classifiers (linear SVM since it is considered as a very efficient text classifier) to perform the MWSA of English text. To construct MCST, Kruskal's algorithm is used along with a similarity measure between every pair of classes as follows. First, the algorithm determined the representative feature vector for each class (using two methods: centroid and sample selection). Then, it evaluates the distance between every pair of classes using Euclidean distance and Tanimoto Coefficient. According to [22], the major benefit of using MCST comparing with other hierarchical classifiers (OVA, OVO, and DAGSVM) is that it handles the overfitting problem in a better way.

In a followup work [23], the authors proposed a probabilistic approach that combines information from lexicons with a Naive Bays classifier. They compared their approach with a Naive Bayes classifier coupled with feature selection in addition to [22]'s approach. They used different datasets from different domains such as: movies, kitchen appliances, music, and post office. In addition to these works, the interested readers are referred to [38], [44], [31], [33], [51] for further information about English MWSA. In the following, we shift our attention to the works specifically geared towards the Arabic language.

Similar to the SA work for the English language, Arabic SA tools can be generally categorized into the following three approaches: the corpus-based approach (supervised), the lexicon-based approach (unsupervised) and a hybrid of the two called the weakly- or semi-supervised approach. Based on our study of the literature, the first paper on Arabic SA appeared in 2006, but it was not until 2010 that we started to see a surge of interest in Arabic SA manifested in an increasing number of published papers. Below, we discuss most of the influential papers in the field providing a comprehensive and up-to-date coverage. In 2006, Ahmad, Cheng and Almas [9] attempted automatic SA on financial news on Arabic and Chinese introducing the local grammar approach that was developed on an English archive and used it on Arabic and Chinese with almost the same results. In [18] Ahmad and Almas extended this work to be on English, Arabic and Urdu. They concluded that, considering the F-measure, Arabic text polarity identification was a bit better than for English text. An observation from the considered languages is that the number of positive sentiments were significantly more than the negative ones in Arabic, English and Urdu. Another work on business-related SA is the work of Elhawary and Elfeky [27] in which a MapReduce implementation was employed to improve the performance of the existing SA systems.

In 2010, Farra [30] followed the same general approach as Ahmad et al. [9], [18] by proposing another SA tool based on a grammatical approach. The approach also took advantage of a precompiled lexicon. The authors compared between the performance of their system on sentence-level and document-level. Other lexicon-based works include [12], [7].

Many papers [5], [36] appeared in the literature to compare the two most common approach for SA: the corpus-based approach and the lexicon-based approach. Moreover, El-Halees [25] studied the two approaches and proposed a hybrid approach that incorporates a third approach based on Maximum Entropy (ME). In another hybrid approach, Abdulla et al. [6] proposed to use an annotated dataset to automatically expand manually-created lexicons leading to significant improvements in the accuracy of the lexicon-based approach.

With the growing interest in Arabic SA, the need for standardized and publicly available datasets to serve as benchmarks became imminent. The works on the OCA and AWATIF datasets is considered pioneering in this aspect. In 2011, Rushdi-Saleh et al. [49], [48] published two papers explaining their Opinion corpus for Arabic (OCA), which consists of 500 movie reviews divided equally among the positive/negative classes. In addition to the process of dataset collection and annotation, the authors performed some experiments on their dataset including studying the effect of applying machine translation on it, which would generate an English version of OCA (EVOCA). On the other hand, the AWATIF dataset [2] was generated with many issues from linguistics point of view in mind such as whether the annotators are aware of certain linguistic features of subjectivity and sentiment analysis or not and how such knowledge would affect their decision. Other works on dataset collection was recently conducted [8], [14], where the authors provided more information about the collected comments than typical SA datasets which focus only on the sentiment orientation of the comment. The dataset of [8], [14] included information about the dialect used, the domain, the gender of the author, etc. Finally, the LABR dataset used in this project was collected by Aly and Atiya [21] in 2013. Since this is the dataset used in this project, more details about it will be provided in the following section. The same group presented another dataset named the Arabic Sentiment Tweets Dataset (ASTD) [42] consisting of more than 10,000 tweets (most of which are non-subjective tweets). Another dataset of large-scale nature was collected by ElSahar and El-Beltagy [28] consisting of more than 33,000 reviews in different domains such as hotels, movies, etc.

One interesting aspect of [3], [1], [37], [2], [4] is that it adds another dimension to the SA problem by bringing subjectivity analysis into the picture. Before thinking of any commercialization of any SA tool, one has to deal with determining whether a text is subjective or objective. Another useful aspect of this line of work is that it brings a lot of interesting and useful ideas from the traditional field of linguistics. Taking subjectivity analysis into account, the authors of [47] used a

simple two-level hierarchical classification system in which the first level distinguish polar vs. objective sentiments and the second level take the polar reviews and decide if they convey positive or negative sentiments.

Other works aimed at solving issues related to SA in general such as how to handle short text documents (e.g., tweets) [50], the different Arabic dialects [50], [13], [24], the imbalance in the dataset [41], the credibility of the comments [19] and how to identify the opinion holder [26], etc.

As mentioned before, the most relevant work to ours are those of Aly and Atiya [21] and Al Shboul el al. [15]. In [21], the authors constructed the Large scale Arabic Book Reviews (LABR) dataset which is, as the authors claim, one of the largest Arabic Sentiment Analysis corpora available recently. The dataset was collected from social network site www.goodreads.com and was subjected to data preparation process. After that, the authors conducted their experiments in order to investigate two main tasks. The first one is to decide whether a review is positive (with rate 4 or 5) or negative (with rate 1 or 2). The second task is to determine the rate of review on a scale from 1 to 5. By employing different features and classifiers (Multinomial Naive Bayes, Bernoulli Naive Bayes, and Support Vector Machines(SVM)) they found that the best accuracy for task one was 91% using SVM, and for task two the greatest accuracy was 50% using also SVM classifier. In another work on the same dataset, the authors of [15] experimented with other classifiers without being able to give significantly better results than the baseline provided by [21]. They also explored the imbalance issue of the LABR dataset arguing that it negatively affects the accuracy of the classifiers.

Other works benefited from the LABR dataset. One example of is the Human Annotated Arabic Dataset (HAAD) of Al-Smadi et al. [17], which consists of more than 1,500 reviews selected from the LABR dataset and annotated for Aspect-Based SA (ABSA) according to the SemEval2014 Task4 guidelines.[2]. In [17], the authors presented several baseline experiments, which they later improved in [43]. The same group followed the same approach of ABSA in another study on the effect of news on the users of social media [16].

III. Methodology and Experiment Setting

This section outlines the methodology and materials used in our work, including the description of the used dataset, the data mining tools employed, and the accuracy measurements used to evaluate the proposed approach.

A. Dataset

The dataset used in this study is the Large Scale Arabic Book Reviews (LABR dataset) which consists of 63,257 book reviews written in MSA as well as colloquial Arabic. The reviews were collected from goodreads.com during 2013. Each book review has a rating (1 to 5) along with the text of the review [21]. The distribution of reviews across the different ratings is discussed in Section III-D. The collected dataset underwent a filtering step to remove newline characters, HTML tags, hyperlinks, repeated dots, non-Arabic characters and some special unicode characters such as the heart symbol and special quotation symbols.

[2]http://alt.qcri.org/semeval2014/task4/

B. Preprocessing and Mining Tools

The tool that is used is Weka 3.7.10. It opens the source software and combines a large set of Machine Learning algorithms, tools for data preparation and preprocessing, and data visualization. Weka allows users to use its algorithms by invoking them using Java code or by applying them directly using its GUI. Our system, which is implemented using the Java programming language, imports Weka library to make benefit from its Machine Learning algorithms.

In our work we use four different classifiers: Support Vector Machine (SVM), Naive Bayes (NB), KNN and Decision Tree (DT). In this work we construct different hierarchical classifiers tree since weka doesn't offer the ability to use such hierarchical classifiers, then we compare the performance of this four classifiers when we use it during hierarchical classifiers tree with the performance of same four classifiers during flat classification problem.

In order to get the ability of using these documents in weka first it needs to be converted to arff file, and this is done by a program written in java code with weka TextDirectoryToArff converter. The next step is to extract features from these documents, the most common approach for features extraction is the Bag-Of-Word. Using StringToWordsVector weka filter we can represent each document as vector, this filter offers the ability to use different techniques help in features extraction and reduction, but before using these techniques to reduce large features vector, we need to tokenize the text to get meaningful words, the WordTokenizer used as tokenizer, which splits or extracts words from the text using standard delimiters. To reduce the number of features, we remove stop words such as pronouns, prepositions, and names of days the week, etc. As for the stemmer, in this our work we did not use any type of stemmers since some of the reviews were written in delicate form of language. The generated dataset divided into two dataset, the first one is the training set which involves about 66% of the original dataset, and the second one is the test dataset which has the rest percentage of the original dataset.

C. Core Classifiers

During this section we will introduce the classifiers that we used in the flat classification problem, as we mentioned before in chapter 2 about the classification methods as SVM, Naive Bayes, Decision Trees, and KNN classifiers. These four classifiers used in our implemented approaches trained under set of rules that will be expressed in the following:

Support Vector Machine (SVM). We used Sequential Minimal Optimization (SMO) algorithm for training SVM classifier, we made for this type of classifiers three experiments according to the kernel type, kernel ideas emerged when data cant be linearly separated, based on finding similarity between two points. In this research we have made our choice to utilize two widely used kernels, Polynomial Kernel (PK) of degree p and Radial Basis Function (RBF) Kernel with σ as the width of the radial basis function. We experimented with both kernel types and different values of p and σ as suggested by [54],

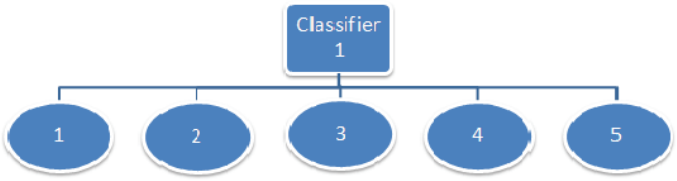

Fig. 1: The flat classifier.

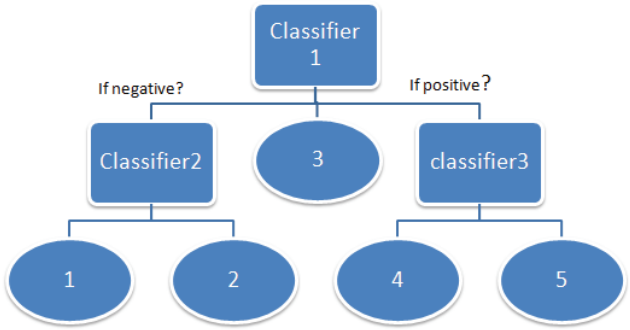

Fig. 2: The 2-level hierarchical classifier.

[40]; however, we only report the settings that produced the best results.

Decision Tree (DT). For this classifier, we used J48 algorithm, using this algorithm we made several experiments by using different values for confidence factor parameter, which controls the size of tree. The default value for this parameter is 0.25 in weka, however, better results are obtained by using other values such as 0.2.

K-Nearest Neighbor (KNN). For this classifier, we used the IBk algorithm. We made several experiments using different numbers of K. The default value for K is 1. We tried other values besides the default one and report the best results. It is worth mentioning that we kept the default distance function, which is the Euclidean distance.

Naive Bayes (NB). For this one, we used the NaiveBayes algorithm. One experiment was made, using default settings that weka provides.

D. Classification Structures

In addition to the flat classification structure, in this work we construct two different hierarchical classification structures. In this section, we explain these structures and discuss the intuition behind them. For the hierarchical structures, the top down approach was followed in constructing the hierarchies and every node represents a binary core classifier (i.e., an instance of one of the four classifiers discussed in the previous subsection). The following provides more details about the structures considered in this work.

The first structure we discuss is the simplest one. It is the flat structure. It is a one-level structure with a single node containing a core classifier. This classifier is trained on the entire training set as is in order for it to distinguish the five classes under consideration in one shot. Figure 1 shows a graphical depiction of this structure.

For this structure, the LABR dataset is used as is. The distribution of the reviews across the five classes under consideration is as follows. Class 1 contains 2,939 reviews, class 2 contains 5,285 reviews, class 3 contains 12,201 reviews, class 4 contains 19,054 reviews and finally class 5 contains 23,778 reviews. This is a very unbalanced dataset.

The first non-flat hierarchical structure is created under two levels as shown in Figure 2. As the figure shows, the first level checks whether an instance is negative, neutral, or positive category. For the LABR dataset, the number of negative reviews (with ratings 1 or 2) is 8,224, while the number of positive reviews (with ratings 4 or 5) is 42,832. The remaining 12,201 reviews are neutral. This level is still suffering from the same imbalance of the original dataset. The classifiers in the next level are customized to determine whether a positive review is weak or strong or to determine whether a negative review is weak or strong. Each one of these classifiers deal with a more balanced datasets compared with the first level classifiers.

Inspired by a mixture of OVA hierarchical structures as well as the high imbalance in the dataset towards positive classes, we devise another hierarchical structures consisting of four levels as shown in Figure 3. As the figure shows, each level has a single core classifier responsible for making a single binary decision. The top level starts by determining whether a review belongs to the majority class (class label 5) or not (i.e., it belongs to one of the class labels 1 through 4). If not, then the decision is moved to the second level classifier which is responsible for determining whether a review belongs to class label 4 or not. This continues until we reach the bottom level classifier which is responsible for determining whether a review belongs to class label 1 or 2. One intuition behind building such a structure is to place the easiest OVA decisions (i.e., the one associated with the majority class) as near the top as possible.

E. Evaluation Measures

In order to evaluate the performance of the different classifiers produced by the different combination of structures with core classifiers, we report different metrics from each experiment. Five metrics are used: accuracy, micro-average precision, micro-average recall, F-measure and Mean Square Error (MSE). To explain these measures, it is generally assumed for a binary classification problem such as ours that there is a "positive" class and a "negative" one. *Precision* calculates the ratio of the true positives to the total number of positives predicted by the classifier. The higher the precision,

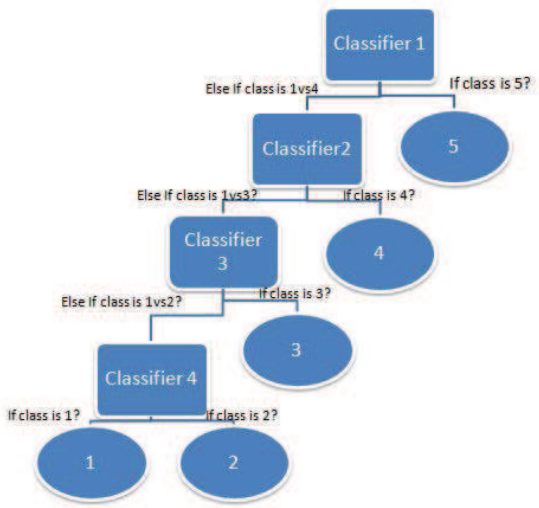

Fig. 3: The 4-level hierarchical classifier.

the more accurate the predication of the positive class. On the other hand, *recall* divides the true positives by the total actual positives belongs to that class. A high recall means high number of comments from the same class is labeled to its exact class. F-measure is weighted average of precision and recall. As for the *accuracy*, it simply reports the ratio of the correctly classified documents regardless of their class. The following are the formulas for these measures [56]:

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+FP+TN+FN} \\ \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ \text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ \text{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(y'_i - y_i)^2 \end{aligned}$$

where TP, FP, TN and FN are the numbers of true positives, false positives, true negatives and false negatives, respectively, and Ci is the number of instances belong to the class i. y' is predicted value y is the true value. True positives and negatives are the correctly classified comments whereas false positives are the number of comments that are incorrectly classified as positive and false negative are the number of comments that are incorrectly classified as negative.

Note that in previous paragraph, we mentioned MSE as an evaluation metric without discussing it. This is because the other four metrics are standard metrics that are assumed to be reported in any work like ours. However, these metrics alone fail to paint a correct picture of the accuracies of the considered classifiers. According to [53], these four measures are not efficient to use with hierarchical classification problems, since they fail to account for the the relationship between categories. Instead, they tackle each one in isolation from the other categories. Simply put, an error of classifying a strong positive review (with a class label 5) as a weak positive review (with a class label 4) is not as serve as classifying it as a strong negative review (with a class label 1). MSE compensate for such issues by relying on the absolute distance between the actual class and the predicted class making it more suitable for a problem like MWSA.

IV. Results and Analysis

In this section, the results of our experiments are described in details. The goal is to study and comparing the performance of the different classification structures using different core classifiers. For the classification process, we use four core classification algorithms.

- The first one is the Sequential Minimal Optimization (SMO) algorithm for training SVM classifier using Radial Basis Function (RBF) kernel with $\sigma = 0.10$. We perform extensive experiments for the different available choices for each kernel and its parameters based on the suggestions and observations of previous similar works in the literature [54], [40]. However, we only report here the settings that gave the best results.
- The second one is the Naive Bayes (NB) classifier used with default settings as provided by the Weka tool.
- The third one is the decision tree algorithm. Specifically, we use the Java implementation of DT provided by the Weka tool, which is known as J48. We note here the we experimented with many values for the decision factor parameter and the best results are obtained when it is set to 0.2.
- The last one is a variant of the KNN algorithm known as the IBK algorithm. After several experiments, we reached a conclusion that using $K = 1$ returned the best results, which is in accordance with previous observations in the literature that argue that increasing the value of K will cause a degradation in accuracy as it decreases the distinct boundaries between categories [39].

As for the testing option, we use the holdout method since the dataset is large enough to allow such a choice. The dataset is split into a training set consisting of two thirds of the original dataset and a testing set consisting of the remaining third of the dataset. Specifically, the testing dataset consists of 21,508 instances distributed among the different classes in a way that preserves the percentages in the original dataset as follows. Class 1 has 981 instances, class 2 has 1,814 instances, class 3 has 4,181 instances, class 4 has 6,451 instances, and finally class 5 has 8,081 instances.

Table I reports the accuracy measures for the flat classification structure with different core classifiers. This table serves two important purposes. It allows us to related our results with existing approaches in the literature on the same dataset (which are all flat approaches), which helps us in arguing about our choices for the evaluation metrics. The second purpose is to allow us to compare the hierarchical structures with the flat structure to determine under which setting resorting to hierarchical classification pays off.

TABLE I: Accuracy of the flat classifier.

	Accuracy	Precision	Recall	F1	MSE
SVM	45.7%	45%	45%	45%	1.6
DT	40.2%	38.7%	40%	39%	1.74
NB	38.2%	36%	38%	37%	1.87
KNN	38.6%	36%	38%	37%	2.05

TABLE II: Accuracy of the 2-level hierarchical classifier.

	Accuracy	Precision	Recall	F1	MSE
SVM	45.2%	54%	45%	49%	0.86
DT	43.9%	54%	43%	48%	0.84
NB	39.9%	42.5%	44.7%	43.5%	2.04
KNN	46.2%	54%	46%	50%	0.9

As was reported in previous works on this dataset ([21], [15]), SVM gives the most accurate results. Note that the table shows non-negligible difference between SVM and DT in terms of the standard accuracy measures. However, the difference in terms of MSE is not as high which means that the mistakes DT is making are probably marginal mistakes. The same thing is observed when comparing NB and KNN. The standard accuracy measures suggests that KNN is slightly better than NB (which is unexpected for a text classification problem); however, MSE suggests otherwise, which is in accordance to what is known in the literature. These observations strengthen our argument that MSE is a more accurate measure of performance than the other four standard measures.

Table II reports the accuracy measures for the 2-level hierarchical classification structure with different core classifiers. The table shows seemingly conflicting results in terms of standard accuracy measures versus MSE. For the standard measures, SVM is negatively affected by the imposition of the first hierarchical strictures, whereas the other classifiers (especially, KNN) are positively affected. As argued before, these observations might be misleading. SVM might be making more mistakes with the use of the first hierarchical structure. But is this necessarily a bad thing? To answer this questions, we need to inspect SVM's mistakes. We do this by computing MSE. The tables shows that imposing the first hierarchical structures cut MSE in half. This is a significant improvement enjoyed by other classifiers. This means that even if the classifiers are not improving significantly in terms of their fine-grained decisions, they are improving in the sense that their mistakes are less severe. The only exception is the NB classifier, which seems like a single-shot classifier that is actually hurt by the decomposition of the original large problem into a set of smaller sub-problems in hierarchical classification.

Table III reports the accuracy measures for the 4-level hierarchical classification structure with different core classifiers. Compared with the results of the two previously discussed classification structures, the results of this structure also seem to have conflicting observations in terms of standard accuracy measures versus MSE. For the standard measures, it looks like this structure is the best one with the accuracy of certain classifiers surpassing the best known results for this dataset. However, inspecting MSE shows that the results of this structure lies in between the results of the two previously discussed structures. Another observation of this table and the ones before it is that NB enjoys both improvement in terms of accuracy and degradation in terms of MSE with the increase in the number of levels in the hierarchical structures. So, it looks like increasing the number of levels allows NB to make less mistakes; however, the mistakes it makes are becoming more severe. On the other hand, the classifier that benefited the most from increasing the number of levels is KNN whose accuracy witnesses significant jumps with every increase in the number of levels.

TABLE III: Accuracy of the 4-level hierarchical classifier.

	Accuracy	Precision	Recall	F1	MSE
SVM	47.4%	65%	47%	55%	1.16
DT	47.6%	66%	47%	55%	1.54
NB	48.9%	70%	48%	57%	2.71
KNN	57.8%	70%	57%	63%	0.96

TABLE IV: Improvements in accuracy over the flat classifier.

	2-level	4-level
SVM	-01.2%	+3.7%
DT	+9.2%	+18.2%
NB	+4.6%	+28.1%
KNN	+19.7%	+49.7%

To appreciate the effect of using a hierarchical classification approach, we report the improvements of each hierarchical structure over the flat structure in terms of both accuracy and MSE in Tables IV and V, respectively. These improvements are computed by taking the difference between the new values and the old values and dividing it by the old values.

The tables show that improvements of 50% or more are obtained in both measures for different settings. The tables show that the best improvement in terms of accuracy is about 50% and it is enjoyed by KNN when imposing the second hierarchical structure. As for the MSE, several settings shows improvements of more than 50% with the best improvements enjoyed by KNN. These results make hierarchical classification a very appealing approach to address a problem like MWSA of Arabic reviews.

TABLE V: Improvements in MSE over the flat classifier.

	2-level	4-level
SVM	+45.9%	+27.4%
DT	+51.8%	+11.5%
NB	-8.8%	-45%
KNN	+55.7%	+53%

V. Conclusion

In this work, we addressed the Multi-Way Sentiment Analysis (MWSA) problem for Arabic reviews. This important problem is yet to find sufficient interest within the research community of Arabic text processing and mining. Among the very limited existing works are a couple of papers following simple flat baseline approaches on a publicly available dataset (LABR). Motivated by the intuition that employing a flat classifier to handle an inherently hierarchical problem such as MWSA is one of the main reasons behind such poor accuracy, we proposed to use hierarchical classification. We presented two different hierarchical structures and compared their accuracy with the flat structure using different core classifiers. The comparison is based on standard accuracy measures such as precision and recall in addition to using the mean squared error (MSE) as a more accurate measure given the fact that not all misclassifications are the same. The results showed that, in general, hierarchical classifiers gave significant improvements (of more than 50% in certain cases) over flat classifiers.

References

[1] Muhammad Abdul-Mageed and Mona T Diab. Subjectivity and sentiment annotation of modern standard arabic newswire. In *Proceedings of the 5th Linguistic Annotation Workshop*, 2011.

[2] Muhammad Abdul-Mageed and Mona T Diab. Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *LREC*, pages 3907–3914, 2012.

[3] Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem. Subjectivity and sentiment analysis of modern standard arabic. In *ACL*, 2011.

[4] Muhammad Abdul-Mageed et al. SAMAR: A system for subjectivity and sentiment analysis of arabic social media. In *WASSA*, 2012.

[5] Nawaf Abdulla et al. Arabic sentiment analysis: Corpus-based and lexicon-based. In *AEECT*. IEEE, 2013.

[6] Nawaf Abdulla et al. Automatic lexicon construction for arabic sentiment analysis. In *FiCloud*, 2014.

[7] Nawaf Abdulla et al. Towards improving the lexicon-based approach for arabic sentiment analysis. *International Journal of Information Technology and Web Engineering (IJITWE)*, 9(3):55–70, 2014.

[8] Nawaf A Abdulla, Mahmoud Al-Ayyoub, and Mohammed Naji Al-Kabi. An extended analytical study of arabic sentiments. *International Journal of Big Data Intelligence*, 1(1):103–113, 2014.

[9] Khurshid Ahmad, David Cheng, and Yousif Almas. Multi-lingual sentiment analysis of financial news streams. In *Grid in Finance*, 2006.

[10] Nizar Ahmed et al. Scalable multi-label arabic text classification. In *ICICS*. IEEE, 2015.

[11] Mohammad Al-A'abed and Mahmoud Al-Ayyoub. A lexicon-based approach for emotion analysis of arabic social media content. In *The International Computer Sciences and Informatics Conference (ICSIC)*, 2016.

[12] Mahmoud Al-Ayyoub, Safa Bani Essa, and Izzat Alsmadi. Lexicon-based sentiment analysis of arabic tweets. *International Journal of Social Network Mining (IJSNM)*, 2(2):101–114, 2015.

[13] Mohammed Al-Kabi et al. An opinion analysis tool for colloquial and standard arabic. In *ICICS*, 2013.

[14] Mohammed Al-Kabi et al. A prototype for a standard arabic sentiment analysis corpus. *The International Arab Journal of Information Technology*, 13(1A):163–170, 2016.

[15] Bashar Al Shboul, Mahmoud Al-Ayyoub, and Yaser Jararweh. Multi-way sentiment classification of arabic reviews. In *Information and Communication Systems (ICICS), 2015 6th International Conference on*, pages 206–211. IEEE, 2015.

[16] Mohammad Al-Smadi et al. Using aspect-based sentiment analysis to evaluate arabic news affect on readers. In *The 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2015.

[17] Mohammad Al-Smadi, Omar Qawasmeh, Bashar Talafha, and Muhannad Quwaider. Human annotated arabic dataset of book reviews for aspect based sentiment analysis. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 726–730. IEEE, 2015.

[18] Yousif Almas and Khurshid Ahmad. A note on extracting sentiments in financial news in english, arabic & urdu. In *CAASL*, 2007.

[19] Izzat Alsmadi, Mohammed Naji Al-Kabi, Heider Wahsheh, and Bassima Bassam. Video spam and public opinion in current middle eastern conflicts. *International Journal of Social Network Mining*, 1(3):318–333, 2013.

[20] Kholoud Alsmearat et al. Emotion analysis of arabic articles and its impact on identifying the author's gender. In *ACS/IEEE AICCSA*, 2015.

[21] Mohamed A Aly and Amir F Atiya. Labr: A large scale arabic book reviews dataset. In *ACL (2)*, pages 494–498, 2013.

[22] Adrian Bickerstaffe and Ingrid Zukerman. A hierarchical classifier applied to multi-way sentiment detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 62–70. Association for Computational Linguistics, 2010.

[23] Minh Duc Cao and Ingrid Zukerman. Experimental evaluation of a lexicon- and corpus-based ensemble for multi-way sentiment analysis. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, pages 52–60, Dunedin, New Zealand, December 2012.

[24] Samhaa R El-Beltagy and Ahmed Ali. Open issues in the sentiment analysis of arabic social media: A case study. In *IIT*, 2013.

[25] Alaa El-Halees. Arabic opinion mining using combined classification approach. In *ACIT*, 2011.

[26] Mohamed Elarnaoty, Samir AbdelRahman, and Aly Fahmy. A machine learning approach for opinion holder extraction in arabic language. *arXiv*, 2012.

[27] Mohamed Elhawary and Mohamed Elfeky. Mining arabic business reviews. In *ICDMW*. IEEE, 2010.

[28] Hady ElSahar and Samhaa R El-Beltagy. Building large arabic multi-domain resources for sentiment analysis. In *Computational Linguistics and Intelligent Text Processing*, pages 23–34. Springer, 2015.

[29] Mosab Faqeeh et al. Cross-lingual short-text document classification for facebook comments. In *FiCloud*, 2014.

[30] Noura Farra, Elie Challita, Rawad Abou Assi, and Hazem Hajj. Sentence-level and document-level sentiment mining for arabic texts. In *ICDMW*, 2010.

[31] Michael Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841. Association for Computational Linguistics, 2004.

[32] Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. Hierarchical approach to emotion recognition and classification in texts. In *Advances in Artificial Intelligence*, pages 40–50. Springer, 2010.

[33] Andrew B Goldberg and Xiaojin Zhu. 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*, pages 45–52. Association for Computational Linguistics, 2006.

[34] Joshua Congfu He, Wee Kheng Leow, and Tet Sen Howe. Hierarchical classifiers for detection of fractures in x-ray images. In *Computer Analysis of Images and Patterns*, pages 962–969. Springer, 2007.

[35] Ismail Hmeidi et al. Automatic arabic text categorization: A comprehensive comparative study. *Journal of Information Science*, 41(1):114–124, 2015.

[36] Maher M. Itani et al. Classifying sentiment in arabic social networks: Naive search versus naive bayes. In *ACTEA*, 2012.

[37] Mohammed Korayem, David Crandall, and Muhammad Abdul-Mageed. Subjectivity and sentiment analysis of arabic: A survey. In AboulElla Hassanien, Abdel-BadeehM. Salem, Rabie Ramadan, and Tai-hoon Kim, editors, *Advanced Machine Learning Technologies and Applications*, volume 322 of *Communications in Computer and Information Science*, pages 128–139. Springer Berlin Heidelberg, 2012.

[38] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[39] Ashis Kumar Mandal and Rikta Sen. Supervised learning methods for bangla web document categorization. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 5(5), Sept 2014.

[40] D Ben Ayed Mezghani, S Zribi Boujelbene, and N Ellouze. Evaluation of svm kernels and conventional machine learning algorithms for speaker identification. *International Journal of Hybrid Information Technology*, 3(3):23–34, 2010.

[41] Asmaa Mountassir, Houda Benbrahim, and Ilham Berrada. An empirical study to address the problem of unbalanced data sets in sentiment classification. In *SMC*. IEEE, 2012.

[42] Mahmoud Nabil, Mohamed Aly, and Amir F Atiya. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, 2015.

[43] Islam Obaidat et al. Enhancing the determination of aspect categories and their polarities in arabic reviews using lexicon-based approaches. In *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2015.

[44] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[45] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2), 2008.

[46] Jonathon Read and John Carroll. Weakly supervised techniques for domain-independent sentiment classification. In *TSA*. ACM, 2009.

[47] Eshrag Refaee and Verena Rieser. Subjectivity and sentiment analysis of arabic twitter feeds with limited resources. In *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme*, page 16, 2014.

[48] Mohammed Rushdi-Saleh, M Teresa Martín-Valdivia, L Alfonso Ureña-López, and José M Perea-Ortega. Oca: Opinion corpus for arabic. *JASIST*, 62(10):2045–2054, 2011.

[49] Mohammed Rushdi-Saleh, Maria Teresa Martín-Valdivia, L Alfonso Ureña-López, and José M Perea-Ortega. Bilingual experiments with an arabic-english corpus for opinion mining. In *RANLP*, 2011.

[50] Amira Shoukry and Ahmed Rafea. Sentence-level arabic sentiment analysis. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 546–550. IEEE, 2012.

[51] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *HLT-NAACL*, pages 300–307, 2007.

[52] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 521–528. IEEE, 2001.

[53] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. Performance measurement framework for hierarchical text classification. *Journal of the American Society for Information Science and Technology*, 54(11):1014–1028, 2003.

[54] Shrawan Kumar Trivedi, Shubhamoy Dey, and Prabandh Shikhar. Effect of various kernels and feature selection methods on svm performance for detecting email spams. *International Journal of Computer Applications*, 66(21), 2013.

[55] Mohammed Abdul Wajeed and T Adilakshmi. Text classification using machine learning. *Journal of Theoretical and Applied Information Technology*, 7(2):119–123, 2009.

[56] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.