Improved Hierarchical Classifiers for Multi-Way Sentiment Analysis

Aya Nuseir[1], Mahmoud Al-Ayyoub[1], Mohammed Al-Kabi[2], Ghasan Kanaan[3], and Riyad Al-Shalabi[3]
[1]Jordan University of Science and Technology, Jordan
[2]Information Technology Department, Al-Buraimi University College, Oman
[3]Amman Arab University, Jordan

Abstract: *Sentiment Analysis (SA) is field in computational linguistics concerned with determining the sentiment conveyed in a piece of text towards certain entities (such as people, organizations, products, services, events, etc.) using NLP tools. The considered sentiments can be as simple as positive vs. negative. A more fine-grained approach known as Multi-Way Sentiment Analysis (MWSA) is based on ranking systems, such as the 5-star ranking system. In such systems, rankings close to each other can be confusing; thus, some researchers have suggested that using Hierarchical Classifiers (HCs) can yield better results compared with traditional Flat Classifier (FCs). Unlike FCs, which try to address the entire classification problem at once, HCs employ some kind of tree structures where the nodes are simple "core" classifiers customized to address a subset of the classification problem. This study aims to explore extensively the use of HCs to address MWSA by studying six different hierarchies. We compare these hierarchies using four well-known core classifiers (SVM, Decision Tree, Naive Bayes, and KNN) using many measures such as Precision, Recall, F1, Accuracy and Mean Square Error (MSE). The experiments are conducted on the Large Arabic Book Reviews (LABR) dataset, which consists of 63K book reviews in Arabic. The results show that using some of the proposed HCs yield significant improvements in accuracy. Specifically, while the best Accuracy and MSE for FC are 45.77% and 1.61, respective-ly, the best accuracy and MSE for an HC are 72.64% and 0.53, respectively. Also, the results show that, in general, KNN(k-nearest neighbors) benefitted the most from using hierarchical classification.*

Keywords: *Sentiment Analysis; Arabic Text Processing; Hierarchical Classifiers, Multi-Way Sentiment Analysis.*

Received March 1, 2017; accepted May 10, 2017

1. Introduction

In recent years, the scientific community paid a considerable attention to the analysis of different posts generated by users of social networks. The nodes of social networks are their users and embedded entities in the social context, while the links, collaborations, and interactions represent the edges of these networks. The analysis of these posts is essential to many parties, such as companies (who have interest in knowing customer opinions about their products and services), governments (who have interest in knowing the general public opinion about its performance), etc.

Users of social networks produce a huge number of multi-media posts daily, and this amount of posts cannot by analysed manually. Therefore, a new field of research emerges called Sentiment Analysis (SA) or Opinion Mining (OM). SA is interested in discovering the subjectivity and the sentiment polarities of these posts (reviews). The computer-based analysis conducted within SA to written text, emoticons, audio excerpts, video excerpts, and images aim to extract the attitude of its author, speaker or actor about a specific topic. The analysis of these multi-media posts is not straightforward and needs many advanced Natural Language Processing (NLP) techniques.

There are many variations of SA. It can be conducted on and aspect-level, a sentence-level, a paragraph-level,
or a document-level. The approaches used in SA at categorized into supervised (corpus-based) approaches, unsupervised (lexicon-based) approaches, and hybrid semi-/weakly-supervised approaches [1, 15].

The most common variation of SA considers two possible sentiment polarities: positive and negative. However, not all applications of SA benefit from such a crude view of the problem. One of the interesting variations of SA is known as Multi-Way Sentiment Analysis (MWSA), where the sentiments are represented by a range of values. A good example of MWSA is the 5-star ranking system, which is the focus of this work. In such systems, rankings close to each other can be confusing; thus, some researchers have suggested that using Hierarchical Classifiers (HCs) can yield better results compared with traditional Flat Classifier (FCs). Unlike FCs, which try to address the entire classification problem at once, HCs employ some kind of tree structures where the nodes are simple "core" classifiers customized to address a subset of the classification problem.

Many researchers depend on using FCs since they are easier and simpler to deal with; however, FCs have

many drawbacks such as having no consideration to the relationship between the predefined categories, which leads to poor performance with huge datasets. On the other hand, in HCs, the structure of the hierarchy can be utilized to cater for the relations between categories [12, 21, 24]. This study aims to improve the performance of the FC model for MWSA by exploring the effects of different hierarchies of classifiers on the performances of different classification models.

SA has been studied for many languages including Arabic. However, the amount of work on Arabic SA is very limited compared with English SA. This is especially true for Arabic MWSA. This is due to many reasons such as limited resources for Arabic SA such as corpora, polarity dictionaries, parsers for the different Arabic dialects, etc. There are many variations of Arabic including Modern Standard Arabic (MSA) and Dialectal Arabic (DA). MSA is well-documented with known syntax and many NLP tools customized for it. This is not the case for colloquial Arabic, which makes analysing MSA posts to identify their polarities easier than analysing DA posts.

This study uses supervised approach (corpus-based) on the document-level, where a 5-star ranking system is used [6, 7, 17]. A large dataset of more than 63K Arabic book reviews called LABR [4] is used in this study. Most of the reviews in this dataset use MSA.

The steps followed in this work are as follows. Firstly, feature extraction is based on the Bag-Of-Words (BOW) approach. Secondly, feature reduction techniques such as stop words removal, feature selection using correlation analysis, etc., are used to reduce the large number of features already produced by BOW. Thirdly, the extracted features are analysed to build a classification model to determine sentiment value.

The remaining parts of this paper are organized as follows: section 2 presents a summary of related studies to hierarchical classification and sentiment analysis. Section 3 presents the framework of this study that includes an overview of the used dataset, data mining tools, pre-processing, flat classifiers, hierarchical structures and an evaluation. Section 4 discusses the statistical methods. Section 5 presents the conclusions and future plans to improve this study.

2. Related work

This work is concerned with leveraging the use of hierarchical classification for the MWSA of Arabic text. In this section, we discuss some of the existing works on hierarchical classification, especially when applied to text processing and SA. Then we discuss the works that paid special attention to Arabic.

The authors of [13] propose an approach that decompose each classification task into a number of simpler classification tasks, where each task is assigned to a node in the classification tree. They notice that the set of relevant features varies widely throughout the hierarchy. Therefore, each classifier can use small subset from the large set of the extracted features. They utilize the concept that each of these subtasks can be classified by using only a very small set of the extracted features. They show that hierarchical classification methods are superior to the flat classification methods. The same conclusion was reached by other works such as [9, 16].

The Web has a huge amount of heterogeneous collections that needs to be classified. The use of hierarchical structure to classify a sample from this heterogeneous collection is suggested in many papers such as in [9], where the authors use Support Vector Machine (SVM) to classify large hierarchical structures. Another similar study is conducted by [18], in which the authors conclude that the use of hierarchical structure helps to increase the classifier precision, where the hierarchical structure is based on the relationships among classes and the hierarchical topic structure.

In real life problems, we face instances that could belong to more than a single class in the underlying taxonomy. The authors of [20] show that the hierarchical structure enhances text classification performance relative to an equivalent flat model. The authors of [8] study such problems with multiclass instances. They present a new learning algorithm, called B-SVM, that differs from simple hierarchical version of SVM called H-SVM mainly in the evaluation phase, in order to be able to assign multi-labels to instances, and conclude that B-SVM is more effective than H-SVM.

The authors of [22] propose a top down level-based classification method that can classify documents to all categories within a tree. Furthermore, they propose a Category-Similarity Measures and Distance-Based Measures to determine the degree of misclassification in measuring the classification performance, and establish an evaluation framework of the performance of hierarchical classification. The authors of [28] propose an evaluation scheme for internal tree nodes to enhance the development process of hierarchical classifier systems. The same team of researchers enhance their study and published a new study that presents a high-performance hierarchical classification system suitable for large hierarchy of categories and a large number of text documents [27].

In his Master's thesis, Granitzer utilizes the hierarchical structure of classes to enhance the accuracy and reduce the computational complexity of the classification process [11]. He concludes that hierarchical classification is beneficial to classifiers with high precision values and lower recall values. Therefore, SVM and BoosTexter classifiers benefit more from hierarchical text categorization relative to other classifiers with low precision values and higher recall values.

Ensemble methods by binarization techniques are presented by [10], especially the most common decomposition strategies: One-Vs-One (OVO)/One-Against-One (1A1) and ONE-VS-ALL (OVA)/One-Against-All (1AA). The authors conclude that OVO methods are generally the best.

Blocking within hierarchical text classification refers to a wrong rejection of documents by the classifiers at higher-levels, and so it cannot be passed to the classifiers at lower levels. To handle this problem, the authors of [23] propose a classifier-centric performance measure called a blocking factor to measure the extent of the blocking. They treat the blocking problem by proposing three methods threshold reduction, restricted voting, and extended multiplicative. Sun *et al*. [23] conclude that their methods are beneficial to reduce the blocking problem, and restricted voting is the best method.

A detailed study of an evaluation of hierarchical classification systems is presented in [14]. The authors proposed two new evaluation measures as an alternative to the traditional measures that usually used to evaluate hierarchical classification systems. The empirical tests conducted by them on three large datasets prove that their proposed new evaluation measures are more accurate than traditional measures.

In [29], the authors introduce an approach for video genre categorization. This approach is based on solving the multi-classification problem using a hierarchical SVM binary tree. The trees are implemented using two types or forms of SVM binary tree. The first form is the local optimal SVM Binary tree, and the main task for this form is to find the best separation at each node using the cross-validation method. The second form is the global form; this form aims to find the best structure and order for the entire tree. Finally, in order to test their approach, they made experiments using the new approach plus to other three approaches as follow: C4.5 decision tree, typical 1-vs.-1 and voting multi-class SVM and Hierarchical SVM built by K-means. The results show their approach outperformed the other three approaches.

The authors of [12] proposed an approach, which tries to solve the weaknesses of the FC approach, which are the lack of structure to define the relations between the predefined categories, and handling each category separately. The approach is based on three phases. Phase one is data preprocessing, where the raw data are converted to normalized vector. This is done using many techniques such as stemmer, stop words filter, and feature selection. The second is the unsupervised learning phase, which is responsible for constructing the hierarchical structure using the Support Vector Clustering (SVC) algorithm. SVC is based on transforming the data from the original space to a high dimensional feature space using Gaussian kernel, and then discovering the smallest sphere which includes all the data in the feature space to create a cluster. However, if there is a large number of documents and features, performing SVC can be time consuming. To over this problem, the authors suggested to perform data compression and dimension reduction using Learning Vector Quantization (LVQ), and Latent Semantic Indexing (LSI). The next phase is the supervised learning phase. In this phase, the tree node classifier is trained according to the number of the branches. If it is equal to two, binary SVM is trained as the tree nod classifier. However, if the number of branches is bigger than two, multi-class SVM (one-against-one or one-against-rest) uses as the tree node classifier. Finally, the results show a good impact in using this approach.

In [21], the authors presented a new algorithm that guaranteed the searches of people on the web organized into meaningful way as hierarchy of topics. This is done according to the users' perspectives. The system is based on two stages. The first stage is the preprocessing and features extraction stage. During this stage, many techniques were performed on documents to convert them to feature vector. In the second stage, a hierarchical classifier is generated for classifying the unlabeled documents. The hierarchal classification algorithm is implemented using a top down approach. Multinomial Naïve Bayes (MNB) classifier is used at each internal nodes of the hierarchy.

The authors of [26] proposed deep classification algorithm to improve the performance of hierarchical classification. Their algorithm consists of two phases. The first phase finds the related categories for a document using cosine similarity measure. The aim of this phase is to reduce the number of categories. The second phase is the classification process using hierarchal structures. The naive bayes classifier was used for training since it does not take a long time for training.

The field of Arabic SA has been growing significantly over the past years [1], however, the use of HCs in Arabic SA (or in Arabic NLP, in general) is still limited. The authors of [19] used the HC in their work, where the first level distinguishes the subjective vs. objective instances and the second level takes the subjective ones and decide if they are positive or negative sentiments.

MWSA has not been getting enough attention. The most interesting work on Arabic MWSA is [4], in which the authors created LABR dataset consisting of 63K Arabic reviews. The LABR has been used in other works [2]; however, all these efforts employ FC. The only work attempting to address the MWSA problem using hierarchical classification is a prior work of ours [3], in which we proposed two HCs and showed that hierarchical classification can outperform flat one. In this paper, we extend that work and propose four more HCs

3. Methodology

In this section, we discuss the proposed hierarchical classification trees. However, we first briefly discuss flat classification. A Flat Classifier (FC) works in the traditional way where it is trained on the entire dataset (with all the classes it contains). For the testing part, it tries to classify each instance in a one-shot fashion.

3.1. Hierarchical Classifiers (HCs)

This section discusses the proposed hierarchical classification structures, where the top-down approach is used and each node in the tree is considered as a FC. The following paragraphs describe these structures in more detail. To be noted that the first two structures are the same as that were used in [3].

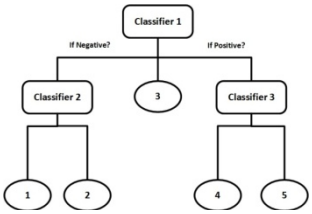

Figure 1. HC#1.

Figure 1 shows the first structure (HC#1), which consists of two levels. The first level finds out the polarity (negative, neutral, positive) of the inputted review and the second level determines whether a positive review is a "strong" positive or a "weak" one.

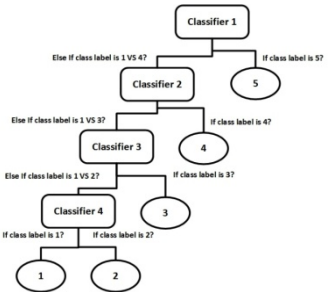

Figure 2. HC#2.

Figure 2 shows the second structure (HC#2) that includes four levels. Each classifier addresses a binary one-vs-all problem. E.g., the classifier in level one decides whether the instance belongs to the class 5 or to the classes 1-4. A classifier 2 task is to classify the instances as class 4 or 1-3 and so on.

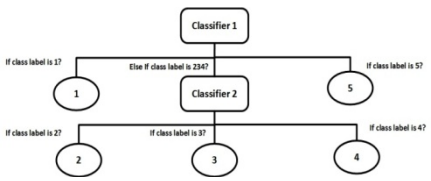

Figure 3.HC#3.

Figure 3 shows the third hierarchical classifier structure (HC#3) that consists of two levels. The first level classifier tries to determine whether a "strong" sentiment is conveyed or not. If not, then it looks into the problem of differentiating between weak positive, neutral and weak negative reviews.

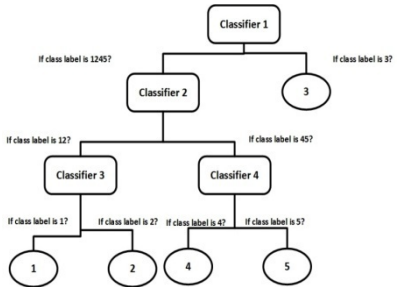

Figure 4. HC#4.

Figure 4 shows the fourth hierarchical classifier structure (HC#4). This structure starts with determining whether a review is neutral or not. For non-neutral reviews, it follows the basic idea of HC#1 and tries to differentiate positive reviews from negative ones. Then, it uses another level of classification to determine the strength of the sentiment.

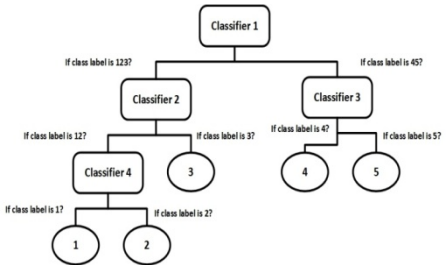

Figure 5. HC#5.

Figure 5 shows the fifth hierarchical classifier structure (HC#5) which is built with the dataset imbalance in mind. HC#5 starts by differentiating positive reviews (majority classes) from negative/neutral reviews. Another classifier is used to determine whether a positive review is a strong positive or a weak one. If the review is not positive, then HC#5 tries to determine whether it is neutral or negative. If it is the latter, it utilizes another classifier to determine whether it is a strong negative or a weak one.

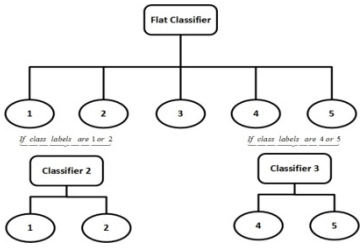

Figure 6. HC#6.

Figure 6 shows the sixth hierarchical structure (HC#6), which is built with a spirit similar to the one used in boosting classifiers. HC#6 is implemented

using two levels. The first level works as FC. If the review is determined to be positive, another classifier (which has been trained to differentiate strong positives from weak ones) is used. The same thing happens if the review is determined to be negative.

4. Experiments and Results

We now discuss the conducted experiments including the experiment set-up and the results.

4.1. Experiment Set-up

The LABR dataset is used in this work. It was constructed by [4] and it consists of more than 63K Arabic book reviews. Each review has a score that ranges from 1 to 5. The numbers of reviews for the scores 1-5 are 2,939, 5,285, 12,201, 19,054 and 23,778, respectively. Obviously, this dataset is unbalanced, most of the reviews located in scores 5 and 4, and while other scores 1 and 2 have the smallest number of the reviews. To prepare the dataset, the textual contents are tokenized and unwanted parts filtered out such as stop words. Then, the filtered dataset is divided into two parts: training part (66%), and testing part (34%).

In order to evaluate the effectiveness of HCs compared with FCs, five metrics are used: Precision (Pr), Recall (Rc), F1, Accuracy (Ac), and Mean Square Error (MSE). Since Pr, Rc and F1 are only suitable for binary classification problems, we report their micro-averages. The description and motivation of these measures can be found in [3].

In this work, the core classifiers (used as FC or as the nodes within HCs) we consider are SVM, Naive Bayes (NB), k-nearest neighbours (KNN) and Decision Tree (DT). We use the Java implementations of these algorithms as provided by the Weka 3.7.10 library. These algorithms have parameters used to tune their performance. We experimented with various combinations of these parameters. However, we only report the best results we obtain. For SVM, Weka provides an implementation of the Sequential Minimal Optimization (SMO). Several experiments were conducted using different kernel parameters. When the data cannot be separated linearly, there is a need to utilize two widely used kernels, Polynomial Kernel (PK) and Radial Basis Function (RBF) [5, 25]. Both of these types are used in our experiments.

Weka provides IBk and J48 as the implementations of the *KNN* and DT classifiers, respectively. For IBk, experiments were conducted using three different values of *K* (1, 5, and 10). We only report the best results obtained with *K*=1. As for J48, experiments were conducted with four different values for confidence factor (cf) parameter, which controls the size of tree. The best results are obtained when cf=0.2.

4.2. Flat Classification Results

The results for FC are presented in Table 1. The table shows that SVM (RBF) yields the best performance according to all traditional accuracy measures whereas SVM (PK) yields the lowest MSE. As for the worst overall performance, it is obtained by KNN.

Table 1. FC results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	45.66%	45.77%	45.72%	45.77%	1.61
SVM (PK)	42.80%	45.36%	44.04%	45.35%	1.48
DT	38.72%	40.27%	39.48%	40.27%	1.75
NB	36.93%	38.11%	37.51%	38.20%	1.87
KNN	36.69%	38.55%	37.60%	38.64%	2.05

4.3. Hierarchical Classification Results

We now discuss the results of each of the six HCs adopted in this study

- *HC#1*: The results for HC#1 are presented in Table 2. The table shows that DT and KNN yield the best performance (KNN has the best accuracy and F1 while DT has the best MSE). As for the worst performance, it is obtained by NB according to all accuracy measures.

Table 2. HC#1 results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	54.80%	45.20%	49.54%	45.20%	0.87
SVM (PK)	55.85%	42.92%	48.54%	42.92%	0.99
DT	54.97%	44.00%	48.88%	43.99%	0.84
NB	42.83%	39.99%	41.36%	39.99%	2.04
KNN	54.99%	46.27%	50.25%	46.27%	0.91

- *HC#2*: The results for HC#2 are presented in Table 3. The table shows that KNN yields the best performance according to all accuracy measures. As for the worst performance (in terms of accuracy and F1), it is obtained by SVM (RBF). What is interesting in this hierarchy is the comparison between NB on one side and SVM (RBF) and DT on the other side. NB's MSE is relatively bad despite having relatively good accuracy and F1. This means that while NB did not make a lot of mistakes, the ones it did make were very severe.

Table 3. HC#2 results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	65.67%	47.48%	55 .11%	47.48%	1.16
SVM (PK)	70.04%	56.35%	62.46%	56.36%	1.22
DT	66.39%	47.62%	55.46%	47.62%	1.55
NB	70.93%	48.97%	57.94%	48.97%	2.71
KNN	70.08%	57.87%	63.39%	57.87%	0.96

- *HC#3*: The results for HC#3 are presented in Table 4. The table shows that NB yields the best performance in terms of accuracy and F1, while SVM (RBF) yields the best performance in terms of MSE. As noted in the previous paragraph, it looks like SVM (RBF) makes more mistakes than NB, however, NB's mistakes were more severe. As for the worst performance, it is obtained by DT according to all accuracy measures, except for MSE, for which SVM (PK) is the worst.

Table 4. HC#3 results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	48.00%	31.41%	37.97%	31.41%	1.39
SVM (PK)	50.83%	24.90%	33.42%	24.89%	3.06
DT	49.76%	27.32%	35.27%	27.32%	2.59
NB	51.65%	43.33%	47.13%	43.32%	2.51
KNN	50.41%	38.97%	43.96%	38.96%	1.84

- *HC#4*: The results for HC#4 are presented in Table 5. The table shows that KNN yields the best performance in terms of accuracy and F1 and the worst performance in terms of MSE. The table also shows that SVM (RBF) yields the best performance in terms of MSE. As for the worst performance in terms of accuracy and F1, it is obtained by SVM (PK), and in terms of MSE, the worst performance is by KNN.

Table 5. HC#4 results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	00.00%	42.72%	00.00%	42.72%	0.88
SVM (PK)	00.00%	41.46%	00.00%	41.46%	1.48
DT	53.20%	41.91%	46.89%	41.91%	0.89
NB	53.11%	43.32%	47.72%	43.32%	1.29
KNN	54.45%	45.31%	49.46%	45.31%	1.51

- *HC#5*: The results for HC#5 are presented in Table 6. The table shows that SVM (RBF) yields the best performance according to all accuracy measures. As for the worst overall performance, it is obtained by KNN.

Table 6. HC#5 results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	60.47%	62.63%	61.53%	62.63%	0.38
SVM (PK)	60.52%	54.83%	57.53%	54.83%	0.62
DT	58.45%	50.73%	54.31%	50.73%	0.65
KNN	57.17%	50.46%	53.61%	50.46%	0.89

- *HC#6*: The results for HC#6 are presented in Table 7. The table shows that KNN yields the best performance according to all accuracy measures. As for the worst overall performance, it is obtained by NB.

Table 7. HC#6 results.

	Pr	Rc	F1	Ac	MSE
SVM (RBF)	74.83%	69.26%	71.94%	69.25%	0.56
SVM (PK)	64.98%	59.08%	61.89%	59.08%	0.64
DT	71.58%	66.56%	68.98%	66.56%	0.55
NB	58.38%	52.80%	55.45%	52.79%	0.97
KNN	77.12%	72.65%	74.82%	72.64%	0.53

- *Flat vs. Hierarchical*: We now compare the results of hierarchical classification with the flat classification. Tables 8 and 9 show the percentages of improvements on the accuracy and MSE for each HC compared with FC.

Table 8. Accuracy improvements.

	HC#1	HC#2	HC#3	HC#4	HC#5	HC#6
SVM (RBF)	-1.25%	3.74%	-31.37%	-6.66%	36.84%	51.30%
SVM (PK)	-5.36%	24.28%	-45.12%	-8.58%	20.90%	30.28%
DT	9.24%	18.25%	-32.16%	4.07%	25.97%	65.28%
NB	4.69%	28.19%	13.40%	13.40%	-	38.19%
KNN	19.75%	49.77%	0.83%	17.26%	32.59%	87.99%

For accuracy, Table 8 shows that the improvements vary greatly across the different HC and core classifier combinations. The biggest overall improvement is obtained by using HC#6 with KNN. KNN benefitted the most from HCs (HC#1, HC#2, HC#4 and HC#6). As for HC#3 and HC#5, the biggest improvements are obtained by using NB and SVM (RBF), respectively. Another interesting observation is that all core classifiers witnessed the best improvements with HC#6.

Table 9. MSE improvements.

	HC#1	HC#2	HC#3	HC#4	HC#5	HC#6
SVM (RBF)	45.96%	27.95%	13.66%	45.34%	76.40%	65.22%
SVM (PK)	33.11%	17.57%	-106.7%	0.00%	58.11%	56.76%
DT	52.00%	11.43%	-48.00%	49.14%	62.86%	68.57%
NB	-9.09%	-44.92%	-34.22%	31.02%	-	48.13%
KNN	55.61%	53.17%	10.24%	26.34%	56.59%	74.15%

For MSE, Table 9 shows that the improvements vary greatly across the different hierarchical structures and core classifier combinations. Similar to accuracy, the biggest overall improvement in MSE is obtained by using HC#6 with KNN. KNN and SVM (RBF) benefitted the most from HCs: HC#1, HC#2 and HC#6 for KNN and HC#3 and HC#5 for SVM (RBF). As for HC#4, the biggest improvement is obtained by using DT. Another interesting observation is that all core classifiers witnessed the best improvements with HC#6. The only exceptions are SVM (RBF and PK), which saw the best improvement with HC#5.

5. Conclusions and Future Work

Many studies showed that hierarchical classification systems yield better performances compared with flat classification systems. This study explores six hierarchical structures with four well-known core classifiers (SVM, DT, NB and KNN), where these structures are varying in their depths. The LABR dataset is used to test the effectiveness of these different hierarchical structures. In this study, several experiments were conducted on each of HC and core classifier combinations. The results showed that using some HCs improved the different accuracy measures compared with FC, whereas other HCs decreased the accuracy. The best accuracy and MSE for FC are 45.77% and 1.61, respectively. As for HCs, the best accuracy and MSE are 72.64% (obtained by HC#6 with KNN) and 0.38 (obtained by HC#5 with SVM), respectively. Overall, KNN benefitted the most from using hierarchical classification. In the future, we intend to use more hierarchical classification trees with possibly more complex structures, and use a mix of classifiers within these trees. Furthermore, future studies will include other datasets.

References

[1] Abdulla N., Al-Ayyoub M., and Al-Kabi M., “An Extended Analytical Study of Arabic

Sentiments," *International Journal of Big Data Intelligence*, vol. 1, no. 1-2, pp. 103-113, 2014.

[2] Al Shboul B., Al-Ayyoub M., and Jararweh Y., "Multi-Way Sentiment Classification of Arabic Reviews," *in Proceeding of 6th IEEE International Conference on Information and Communication Systems*, Amman, pp. 206-211, 2015.

[3] Al-Ayyoub M., Nuseir A., Kanaan G., and Al-Shalabi R., "Hierarchical Classifiers for Multi-Way Sentiment Analysis of Arabic Reviews," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, pp. 531-539, 2016.

[4] Aly M. and Atiya A., "LABR: A Large Scale Arabic Book Reviews Dataset," *in Proceeding of the Association for Computational Linguistics*, Sofia, pp. 494-498, 2013.

[5] Begg R. and Kamruzzaman J., "A Machine Learning Approach for Automated Recognition of Movement Patterns Using Basic, Kinetic and Kinematic Gait Data," *Journal of Biomechanics*, vol. 38, no. 3, pp. 401-408, 2005.

[6] Bickerstaffe A. and Zukerman I., "A Hierarchical Classifier Applied to Multi-way Sentiment Detection," *in Proceeding of the 23rd International Conference on Computational Linguistics*, Beijing, pp. 62-70, 2010.

[7] Cao M. and Zukerman I., "Experimental Evaluation of a Lexicon- and Corpus-based Ensemble for Multi-Way Sentiment Analysis," *in Proceeding of Australasian Language Technology Association Workshop*, New Zealand, pp. 52-60, 2012.

[8] Cesa-Bianchi N., Gentile C., and Zaniboni L., "Hierarchical Classification: Combining Bayes with SVM," i*n Proceeding of the 23rd International Conference on Machine Learning*, Pennsylvania, pp. 177-184, 2006.

[9] Dumais S. and Chen H., "Hierarchical Classification of Web Content," *in Proceeding of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, Athens, pp. 256-263, 2000.

[10] Galar M., Ferna´ndez A., Barrenechea E., Bustince H., and Herrera F., "An Overview of Ensemble Methods for Binary Classifiers in Multi-Class Problems: Experimental Study on One-vs-One and One-vs-All Schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761-1776, 2011.

[11] Granitzer M., "Hierarchical Text Classification Using Methods from Machine Learning," M.S. Thesis, Graz University of Technology, 2003.

[12] Hao P., Chiang J., and Tu Y., "Hierarchically SVM Classification Based on Support Vector Clustering Method and its Application to Document Categorization," *Expert Systems with Applications*, vol. 33, no. 3, pp. 627-635, 2007.

[13] Koller D. and Sahami M., "Hierarchically Classifying Documents Using Very Few Words," *in Proceeding of the 14th International Conference on Machine Learning*, San Francisco, pp. 170-178, 1997.

[14] Kosmopoulos A., Partalas I., Gaussier E., Paliouras G., and Androutsopoulos I., "Evaluation Measures for Hierarchical Classification: A Unified View and Novel Approaches," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 820-865, 2015.

[15] Liu S. and Chen J., "A Multi-Label Classification Based Approach for Sentiment Classification," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1083-1093, 2015.

[16] McCallum A., Rosenfeld R., Mitchell T., and Ng A., "Improving Text Classification by Shrinkage in a Hierarchy of Classes," *in Proceeding of the 15th International Conference on Machine Learning*, San Francisco, pp. 359-367, 1998.

[17] Mehra N., Khandelwal S., and Patel P., "Sentiment Identification using Maximum Entropy Analysis of Movie Reviews," Available: http://web.stanford.edu/class/cs276a/projects/reports/nmehra-kshashi-priyank9.pdf, Last Visited 2002.

[18] Pulijala A. and Gauch S., "Hierarchical Text Classification," Available: https://pdfs.semanticscholar.org/2229/a8b5bf3f2c6622d4c3fd6253c1f8f4f3510b.pdf, Last Visited 2004.

[19] Refaee E. and Rieser V., "Subjectivity and Sentiment Analysis of Arabic Twitter Feeds with Limited Resources," *in Proceeding of the Language Resources and Evaluation Conference*, Lisbon, pp. 16-21, 2014.

[20] Ruiz M. and Srinivasan P., "Hierarchical Text Categorization Using Neural Networks," *Information Retrieval*, vol. 5, no. 1, pp. 87-118, 2002.

[21] Singh A. and Nakata K., "Hierarchical Classification of Web Search Results using Personalized Ontologies," *in Proceeding of the 3rd International Conference on Universal Access in Human-Computer Interaction*, Las Vegas, pp. 1-10, 2005.

[22] Sun A. and Lim E., "Hierarchical Text Classification and Evaluation," *in Proceeding of IEEE International Conference on Data Mining*, San Jose, pp. 521-528, 2001.

[23] Sun A., Lim E., Ng W., and Srivastava J., "Blocking Reduction Strategies in Hierarchical Text Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1305-1308, 2004.

[24] Tikk D. and Biró G., "Experiment with a Hierarchical Text Categorization Method on the

WIPO-Alpha Patent Collection," *in Proceeding of 4th International Symposium on Uncertainty Modeling and Analysis*, Maryland, pp. 104-109, 2003.

[25] Trivedi S. and Dey S., "Effect of Various Kernels and Feature Selection Methods on SVM Performance for Detecting Email Spams," *International Journal of Computer Applications*, vol. 66, no. 21, pp. 18-23, 2013.

[26] Xue G., Xing D., Yang Q., and Yu Y., "Deep Classification in Large-scale Text Hierarchies," *in Proceeding of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, pp. 619-626, 2008.

[27] Yoon Y., Lee C., and Lee G., "An Effective Procedure for Constructing a Hierarchical Text Classification System," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 431-442, 2006.

[28] Yoon Y., Lee C., and Lee G., "Systematic Construction of Hierarchical Classifier in SVM-based Text Categorization," *in Proceeding of International Conference on Natural Language Processing*, Hainan Island, pp. 616-625, 2005.

[29] Yuan X., Lai W., Mei T., Hua X., Wu X., and Li S., "Automatic Video Genre Categorization Using Hierarchical SVM," *in Proceeding of IEEE International Conference on Image Processing*, Atlanta, pp. 2905-2908, 2006.

Aya Nuseir earned her MSc in 2015 and BSc in 2012 in Computer Science from Jordan university of Science and Technology (JUST). She worked as a programmer at NCARE in 2015. She is currently a part time lecturer at JUST. Her interests include data mining and Arabic text categorization.

Mahmoud Al-Ayyoub Received his Ph.D. in computer science from Stony Brook University in 2010. He is currently an associate professor of computer science at Jordan University of Science and Technology (JUST). His research interests include high performance computing, machine learning and AI.

Mohammed Al-Kabi is an assistant Professor in the Information Technology Department at Al-Buraimi University College, Buraimi, Sultanate of Oman. Prior to joining Al-Buraimi University College, he worked 4 years in Computer Science Department, Faculty of IT, at Zarqa University, he worked 11 years in IT faculty at Yarmouk University in Jordan, Nahrain University and Mustanserya University in Iraq for six years. He also worked as a part-time lecturer at Jordan University of Science and Technology (JUST), Princess Sumaya University for Technology (PSUT) and Sunderland University. AL-Kabi's research interests include Sentiment Analysis and Opinion Mining, Big Data, Information Retrieval, Web search engines, Data Mining, Social media, and Natural Language Processing. He is the author of more than 97 peer-reviewed articles on these topics. His teaching interests focus on Information Retrieval, Big Data, Web Programming, Data Mining, DBMS (MYSQL, ORACLE & MS Access). Furthermore, AL-Kabi has an interest in the quality of higher education and university rankings and published many articles in this field.

Ghassan Kanaan received his Ph.D. in computer science from Illinois Institute of Technology in 1996. He is currently a full professor of computer science and the Acting President of Amman Arab University, Amman, Jordan. His research interests include information retrieval and natural language processing.

Riyad Al-Shalabi received his Ph.D. in computer science from Illinois Institute of Technology in 1996. He is currently a full professor of computer science and the Dean of Scientific Research and Graduate Studies at Amman Arab University, Amman, Jordan. His research interests include information retrieval and natural language processing.