International Arab Conference on Information Technology (ACIT'2016)

Improved Hierarchical Classifiers for Multi-Way Sentiment Analysis

Aya Nuseir,[1] Mohammed Al-Kabi,[2] Mahmoud Al-Ayyoub,[1] Ghasan Kanaan[3] and Riyad Al-Shalabi[3]
[1] Jordan University of Science and Technology, Irbid, Jordan
E-mails: nuseir_aya@yahoo.com, maalshbool@just.edu.jo
[2] Information Technology Department, Al-Buraimi University College, Buraimi, Oman
E-mail: mohammed@buc.edu.om
[3] Amman Arab University, Amman, Jordan
E-mail: gkanaan@aau.edu.jo, shalabi@aau.edu.jo

Abstract: *Sentiment Analysis (SA) is a computational study of the sentiments expressed in text toward entities (such as news, products, services, organizations, events, etc.) using NLP tools. The conveyed sentiments can be quantified using a simple positive/negative model. A more fine-grained approach known as Multi-Way SA (MWSA) uses a ranking system like the 5-star ranking system. Since in such systems, rankings close to each other can be confusing, it has been suggested that Hierarchical Classifiers (HCs) can outperform traditional Flat Classifier (FCs) for MWSA. Unlike FCs which try to address the entire classification problem at once, HCs utilizes tree structures where the nodes are simple 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 with four well-known 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 LABR dataset consisting of 63K book reviews in Arabic. The results show that using some of the proposed HCs yield a significant improvement in accuracy. Specifically, while the best Accuracy and MSE for FC are 45.77% and 1.61, respectively, the best accuracy and MSE for an HC are 72.64% and 0.53, respectively. Also, the results show that, in general, KNN benefitted the most from using hierarchical classification.*

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

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 (Abdulla, et al., 2014), and (Liu & Chen, 2015).

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 look at the problem. One of the interesting variations of SA is known as Multi-Way SA where the sentiment is quantified by a value from a certain range. A good example of MWSA is with 5-star ranking system. This is the focus of this work.

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 the small size of the Arabic NLP community and the many challenges faced when processing Arabic text.

There are many variations of Arabic including Modern Standard Arabic (MSA) and colloquial Arabic. MSA is well-documented with known syntax

and many NLP tools customized for it, which is not the case for colloquial Arabic. This makes analysing MSA posts to identify their polarities easier than analysing colloquial Arabic posts.

This study uses supervised approach (corpus-based) on the document-level, where a 5-star ranking system is used (Bickerstaffe & Zukerman, 2010), (Cao & Zukerman, 2012), (Mehra, et al., 2002). Therefore, a large dataset of more than 63K Arabic book reviews called LABR (Aly & Atiya, 2013) is used in this study. Most of the reviews in this dataset use MSA.

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. This study aims to improve the performance of the classification model by exploring the effects of different hierarchies of classifiers on the performances of different classification models.

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 (Koller & Sahami, 1997) paper proposes 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 (McCallum, et al., 1998) and (Dumais & Chen, 2000).

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 conducted by a number of authors such as the study of (Dumais & Chen, 2000) in which they use Support Vector Machine (*SVM*) to classify large hierarchical structures. Another similar study is conducted by (Pulijala & Gauch, 2004), 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. Ruiz et al. study shows that the hierarchical structure enhances text classification performance relative to an equivalent flat model (Ruiz & Srinivasan, 2002). Cesa-Bianchi et al. 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 (Cesa-Bianchi et al., 2006).

Sun and Lim (Sun & Lim, 2001) 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. Yoon et al. propose an evaluation scheme for internal tree nodes to enhance the development process of hierarchical classifier systems (Yoon et al., 05). The same team of researchers (Yoon et al., 2005) 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 (Yoon et al., 2006).

In his Master's thesis Granitzer utilises the hierarchical structure of classes to enhance accuracy and reduce computational complexity (Granitzer, 2003). 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 Galar et al., especially the most common

decomposition strategies: One-vs-one (OVO)/One-Against-One (1A1) and one-vs-all (OVA)/One-Against-All (1AA). They conclude that OVO methods are generally the best (Galar et al., 2011).

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 Sun et al. (Sun et al., 2004) 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. conclude that their methods are beneficial to reduce the blocking problem, and Restricted Voting is the best method (Sun et al., 2004).

A detailed study of an evaluation of hierarchical classification systems is presented by Kosmopoulos et al. (Kosmopoulos et al, 2015). They 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.

The field of Arabic SA has been growing significantly over the past years (Abdulla, et al., 2014). However, MWSA has not been getting enough attention. The most interesting work on Arabic MWSA is (Aly & Atiya, 2013), in which the authors created the Large Arabic Book Reviews (LABR) dataset consisting of 63K Arabic reviews. The LABR has been used in other works; however, all these efforts employ FC. The only work attempting to address the MWSA problem using hierarchical classification is a prior work of ours (Al-Ayyoub et al., 2016), 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 (Al-Ayyoub et al., 2016).

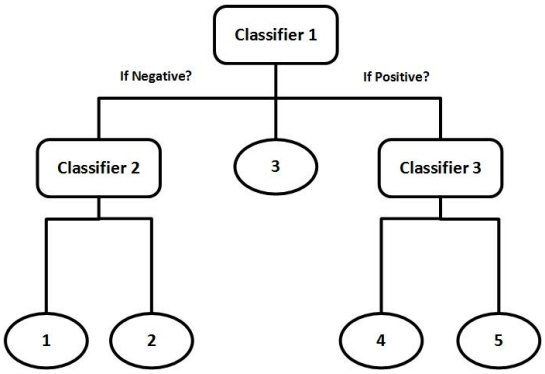

Figure 1: HC#1

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 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. Classifier 2 task is to classify the instances as class 4 or 1-3 and so on.

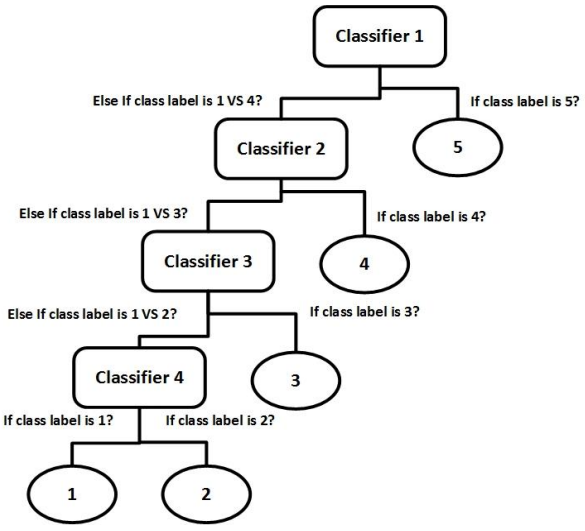

Figure 2: HC#2

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.

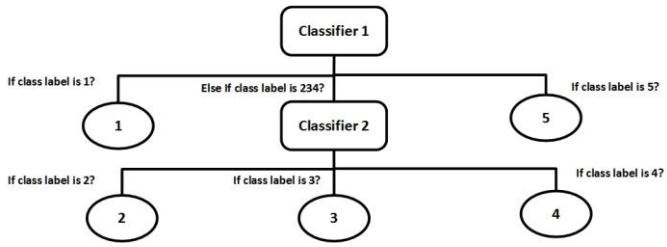

Figure 3: HC#3

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.

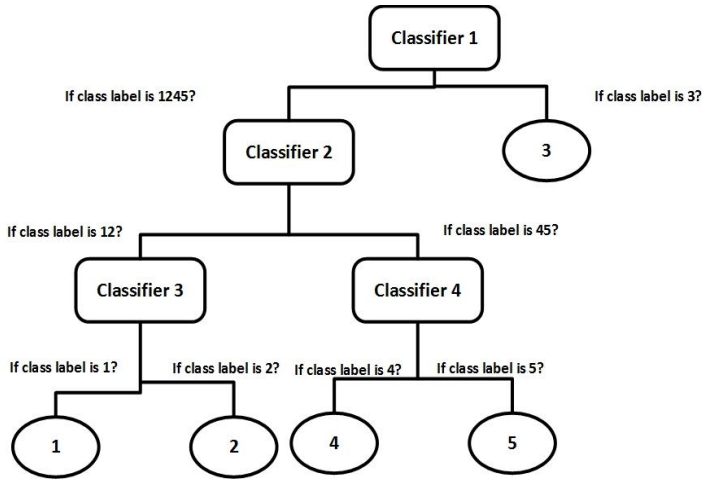

Figure 4: HC#4

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.

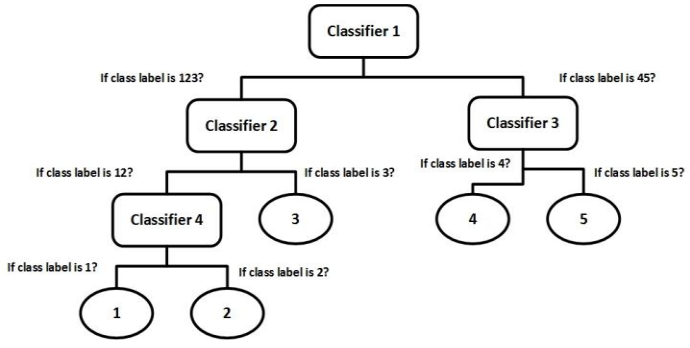

Figure 5: HC#5

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.

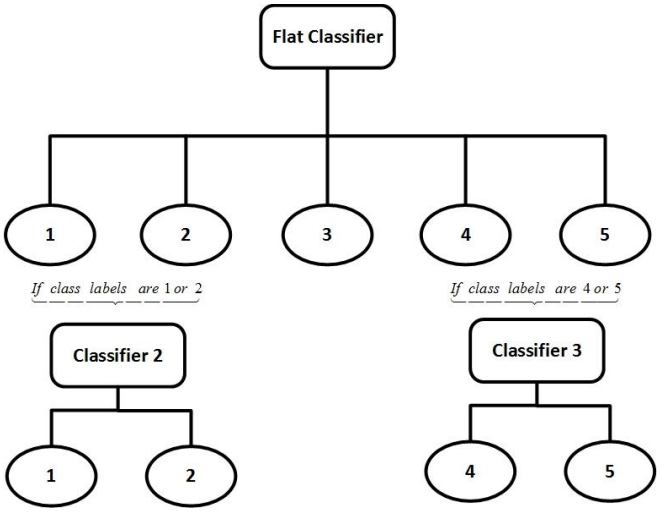

Figure 6: HC#6

4. Experiments and Results

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

4.1. Experiment Set-up

The Large-scale Arabic Book Review (LABR) dataset is used in this work. It was constructed by (Aly & Atiya, 2013) 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 (Al-Ayyoub et al., 2016).

In this work, the core classifiers (used as FC or as the nodes within HCs) we consider are Support Vector Machine (SVM), Naive Bayes (NB), 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). Three 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*). Both of these types are used in our experiments; however, only the best results (obtained by the RBF kernel) are reported (Trivedi & Dey, 2013), (Begg & Kamruzzaman, 2005).

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 yields the best performance according to all accuracy measures. As for the worst overall performance, it is obtained by KNN.

Table 1: FC Results

	Pr	Rc	F1	Ac	MSE
SVM	45.66%	45.77%	45.72%	45.77%	1.61
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	54.80%	45.20%	49.54%	45.20%	0.87
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. What is interesting in this hierarchy is the comparison between NB on one side and SVM 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	65.67%	47.48%	55 .11%	47.48%	1.16
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 yields the best performance in terms of MSE. As noted in the previous paragraph, it looks like SVM 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.

Table 4: HC#3 Results

	Pr	Rc	F1	Ac	MSE
SVM	48.00%	31.41%	37.97%	31.41%	1.39
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 yields the best performance in terms of MSE. As for the worst performance in terms of accuracy and F1, it is obtained by DT.

Table 5: HC#4 Results

	Pr	Rc	F1	Ac	MSE
SVM	00.00%	42.72%	00.00%	42.72%	0.88
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 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	60.47%	62.63%	61.53%	62.63%	0.38
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	74.83%	69.26%	71.94%	69.25%	0.56
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	-1.25%	3.73%	-31.38%	-6.67%	36.82%	51.31%
DT	9.25%	18.24%	-32.16%	4.07%	25.95%	65.27%
NB	4.69%	28.17%	13.41%	13.49%	-	38.2%
KNN	19.73%	49.75%	0.83%	17.25%	30.58%	87.98%

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. 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	45.92%	27.49%	13.46%	45.18%	76.56%	65.08%
DT	51.84%	11.56%	-48%	49.35%	62.89%	68.82%
NB	-8.87%	-45.07%	-34.17%	30.8%	-	47.98%
KNN	55.71%	53.01%	10.24%	26.58%	56.66%	74.01%

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 benefitted the most from HCs (HC#1, HC#2 and HC#6 for KNN and HC#3 and HC#5 for SVM). 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 exception is SVM, which saw the best improvement with HC#5.

5. Conclusion 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

Abdulla, N., et al. (2014). An extended analytical study of Arabic sentiments. *IJBDI*, *1*(1/2), 103.

Al-Ayyoub, M., et al. (2016). Hierarchical Classifiers for Multi-Way Sentiment Analysis of Arabic Reviews. *IJACSA*, *7*(2), 531–539.

Aly, M., & Atiya, A. (2013). LABR: A Large Scale Arabic Book Reviews Dataset. *Aclweb.Org*, 494–498.

McCallum, A., et al. (1998). Improving Text Classification by Shrinkage in a Hierarchy of Classes (pp. 359–367).

Begg, R., & Kamruzzaman, J. (2005). A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data. *Journal of Biomechanics*, *38*(3), 401–408.

Bickerstaffe, A., & Zukerman, I. (2010). A Hierarchical Classifier Applied to Multi-way Sentiment Detection. *Computational Linguistics*, (August), 62–70.

Cao, M. D., & Zukerman, I. (2012). Experimental Evaluation of a Lexicon- and Corpus-based Ensemble for Multi-way Sentiment Analysis. In *ALTA 2012* (pp. 52–60).

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006). Hierarchical Classification: Combining Bayes with SVM. In *ICML '06*, 177–184.

Dumais, S., & Chen, H. (2000). Hierarchical classification of Web content. In *SIGIR '00*, 256–263.

Galar, M. et al. (2011). 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*, *44*(8), 1761–1776.

Granitzer, M. (2003). *Hierarchical text classification using methods from machine learning*. *Master's Thesis, Graz University of Technology*.

Koller, D., & Sahami, M. (1997). Hierarchically Classifying Documents Using Very Few Words. In *ICML 2015*, 170–178.

Kosmopoulos, A. et al. (2015). Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, *29*(3), 820–865.

Liu, S. M., & Chen, J.-H. (2015). A multi-label classification based approach for sentiment classification. *ESwA*, *42*(3), 1083–1093.

Mehra, N. et al. (2002). Sentiment identification using maximum entropy analysis of movie reviews.

Pulijala, A., & Gauch, S. (2004). Hierarchical text classification. *CITSA 2004*, *1*, 257–262.

Ruiz, M. E., & Srinivasan, P. (2002). Hierarchical Text Categorization Using Neural Networks. *Information Retrieval*, *5*(1), 87–118.

Sablatnig, R. et al.. (1998). Hierarchical classification of paintings using face- and brush stroke models. *Pattern Recognition*, *1*(16), 172–174.

Sun, A., & Lim, E. (2001). Hierarchical Text Classification and Evaluation. In *ICDM*, 521–528.

Sun, A. et al. (2004). Blocking reduction strategies in hierarchical text classification. *IEEE Transactions on Knowledge and Data Engineering*, *16*(10), 1305–1308.

Trivedi, S., & Dey, S. (2013). Effect of Various Kernels and Feature Selection Methods on SVM Performance for Detecting Email Spams. *IJCA*, *66*(21), 18–23.

Yoon, Y. et al. (2005). Systematic construction of hierarchical classifier in SVM-based text categorization. *IJCNLP 2004*, 616–625.

Yoon, Y. et al. (2006). An effective procedure for constructing a hierarchical text classification system. *JASIST*, *57*(3), 431–442.