I.J. Information Engineering and Electronic Business, 2017, 3, 36-42
Published Online May 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2017.03.05

A Comparative Study of Flat and Hierarchical Classification for Amharic News Text Using SVM

Alemu Kumilachew Tegegnie
Bahirdar Instituite of Technology (BiT), Bahirdar University, Ethiopia
Email: alemupilatose@gmail.com

Adane Nega Tarekegn
Bahirdar Instituite of Technology (BiT), Bahirdar University, Ethiopia
Email: nega2002@gmail.com

Tamir Anteneh Alemu
Bahirdar Instituite of Technology (BiT), Bahirdar University, Ethiopia
Email: tamirat.1216@gmail.com

***Abstract*—**The advancement of the present day technology enables the production of huge amount of information. Retrieving useful information out of these huge collections necessitates proper organization and structuring. Automatic text classification is an inevitable solution in this regard. However, the present approach focuses on the flat classification, where each topic is treated as a separate class, which is inadequate in text classification where there are a large number of classes and a huge number of relevant features needed to distinguish between them. This paper aimed to explore the use of hierarchical structure for classifying a large, heterogeneous collection of Amharic News Text. The approach utilizes the hierarchical topic structure to decompose the classification task into a set of simpler problems, one at each node in the classification tree. An experiment had been conducted using a categorical data collected from Ethiopian News Agency (ENA) using SVM to see the performances of the hierarchical classifiers on Amharic News Text. The findings of the experiment show the accuracy of flat classification decreases as the number of classes and documents (features) increases. Moreover, the accuracy of the flat classifier decreases at an increasing number of top feature set. The peak accuracy of the flat classifier was 68.84 % when the top 3 features were used. The findings of the experiment done using hierarchical classification show an increasing performance of the classifiers as we move down the hierarchy. The maximum accuracy achieved was 90.37% at level-3(last level) of the category tree. Moreover, the accuracy of the hierarchical classifiers increases at an increasing number of top feature set compared to the flat classifier. The peak accuracy was 89.06% using level three classifier when the top 15 features were used. Furthermore, the performance between flat classifier and hierarchical classifiers are compared using the same test data. Thus, it shows that use of the hierarchical structure during classification has resulted in a significant improvement of 29.42 % in exact match precision when compared with a flat classifier.

***Index Terms*—**Automatic Text Classification, Flat Classification, Hierarchical Classification, Machine Learning, Support Vector Machine (SVM).

I. Introduction

Humans use classification techniques to organize things in various activities of their life. People make their own judgment to classify things in their everyday life – they classify things based on similarities or likeliness of color, size, concept, and ideas, subject and so on such that searching and information retrieval can be easier.

The need to classify information resources has become an important issue as the production of such resources, in any form and format, increase dramatically from time to time due to the advancement of information technologies. Nowadays, news items are produced every day in digital devices in different languages such as English, French, German, Arabic, Chinese, Amharic, etc. However, most of the time, text classification process is done manually which brings about enormous costs in terms of time and money. In other words, organizing documents by hand or creating rules for filtering is painstaking and labor-intensive.

II. Automatic Text Classification

In a system where there is large collection of documents, retrieval of a given document or set of documents is necessitated proper and systematical classification of such resources. Automatic classification systems are very desirable since they minimize such classification problems [1]. Text classification is the task

of automatically assigning a set of documents into categories (or classes, or topics) from a predefined set [2]. If C= {c1, c2…cm} is a set of categories (classes) and D= {d1,d2,…,dn} is a set of documents, the purpose of text classification is assigning ci to dj ($1 \le i \le m$ and $1 \le j \le n$) a value of 0 if the document dj does not belong to ci ; otherwise a value of 1.

Automatic document classification can be done either by treating each category/topic independently of one another (flat classification) or by considering the structural relationship among a given categories (hierarchical classification).

A. Flat classification

In a flat classification, the predefined categories are treated individually and equally so that no structures exist to define relationships among them [3]. A single huge classifier is trained which categorizes each new document as belonging to one of the possible predefined classes.

In a flat classification approach, as we use a large corpus we may have hundreds of classes and thousands of features, the borderlines between document classes are blurred; and the computational cost of training a classifier for a problem of this size is prohibitive. In such approaches, there will have many thousands of parameters which need to be estimated, the resulting classifier will have also a large variance, and thus it will lead to overfitting of the training data.

B. Hierarchical classification

Hierarchical text classification is proposed by [4], as divide-and-conquer approach that utilizes the hierarchical topic structure to decompose the classification task into a set of simpler problems, one at each node in the classification hierarchy. Rather than ignoring the topical structure and building a single huge classifier for the entire task, we use the structure to break the problem into manageable size pieces. As shown in figure 1, in such a hierarchical structure document types become more specific as we go down in the hierarchy.

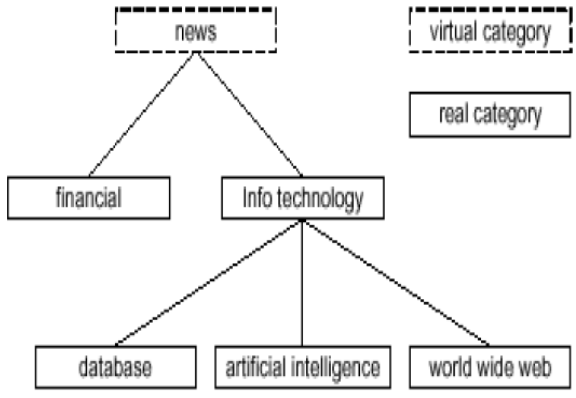

Fig.1. Sample hierarchical structure

III. Related Literature

The study is conducted on large collection of Amharic news text released by Ethiopian News Agency (ENA). Amharic is the native language of people living in the north central part of Ethiopia. It is the working language of the Federal Republic of Ethiopia. There are significant number of immigrants who speak Amharic in the Middle East, Asia, Western Europe and North America [5]. The language has its own writing system that uses the Ge'ez alphabet.

Recently, there are numerous electronic documents produced and stored in Amharic language. More specifically, ENA produces and stores more than 100,000 news articles [6] with a total of 110 categories.

However, using manual classification for such large number of classes brings about enormous costs in terms of time and money. To cope up with these challenges a couple of studies have been done [7], [8], and [9]. However, all attempts are done with flat classification approach. In such a situation, the result of their studies showed that the accuracy of the classifier degrades as the number of categories (classes) and the number of document features increases. In a hierarchical classification approach, however, the classifier is only focus on the relevant features of the classes since the number of classes and documents decreases as we go down the classification tree. The aim of this study is, therefore, to explore the possibility of designing and developing hierarchical news text classifier that is effective and efficient in classifying large, heterogeneous collection of Amharic news text.

IV. Methodology

A. Data source

Table 1. Statistics of data collected from ENA (2007-2010)

No	Major class	No. of Subclass	No. of docs (2007-2010)
1	Culture and Tourism (ባህልና ቱሪዝም)	9	791
2	Economy (ኢኮኖሚ)	11	1134
3	Education (ትምህርት)	14	1117
4	Health (ጤና)	11	1067
5	Law and Justice (ህግናፍትህ)	6	704
6	Politics (ፖለቲካ)	9	1136
7	Social (ማህበራዊ)	11	1064
8	Sport (ስፖርት)	7	523
9	Accident (አደጋ)	3	2102
10	Weather and Environmental Protection (የአካባቢ ጠበቃና የአየር ሁኔታ)	5	2203
11	Relations, Defense, and Security (ግንኙነት፣ መከላከያና ደህንነት)	8	2017
12	Science and Technology (ሳይንስና ቴክኖሎጂ)	4	1150
	Total	98	15008

The data used in the study is taken from Ethiopian News Agency (ENA). As shown in table 1, a total of 15008 Amharic news texts from 12 major classes and 98 subclasses have been collected.

B. Document Preprocessing

The original data was a two level category, where documents were assigned in sub categories. The sub categories in ENA were used as leaf nodes (last category) in the hierarchy where the actual news items are assigned. However, this study needs to have more than two levels of categorical data to achieve its defined purpose. Generation of more than two categorical data is done through the help of expert's judgment and document-similarity matrix methods (see table 2 below) using cosine similarity method. Cosine similarity produces a value between [-1, 1] and 0.5 was used as a threshold document similarity value among those sub-categories. Thus, Health class will have three categories. Each category is given appropriate names and each with 4, 4 &2 subcategories as shown by shaded cells. As a result, 8 major (level-0) classes, 20 level- 1 classes and 69 level-2 classes with 5100 documents are generated.

Table 2. Results of the flat classifier when 8, 20, 69 classes are used

No.of classes	No. of documents			Accuracy (%)
	Training set	Testing set	Total	
8	1785	765	2550	80.34
20	2499	1071	3570	66.09
69	3213	1377	4590	50.32

Moreover, the stemming and stop word removal algorithm of [10] have been implemented and used in the study. As a result, a total of 189877 features are generated in all document preprocessing techniques. Such feature are used in a tf*idf calculation. More specifically, the architecture in figure 2 has explained the efforts made while undertaking the whole experiment up to building the model/classifier.

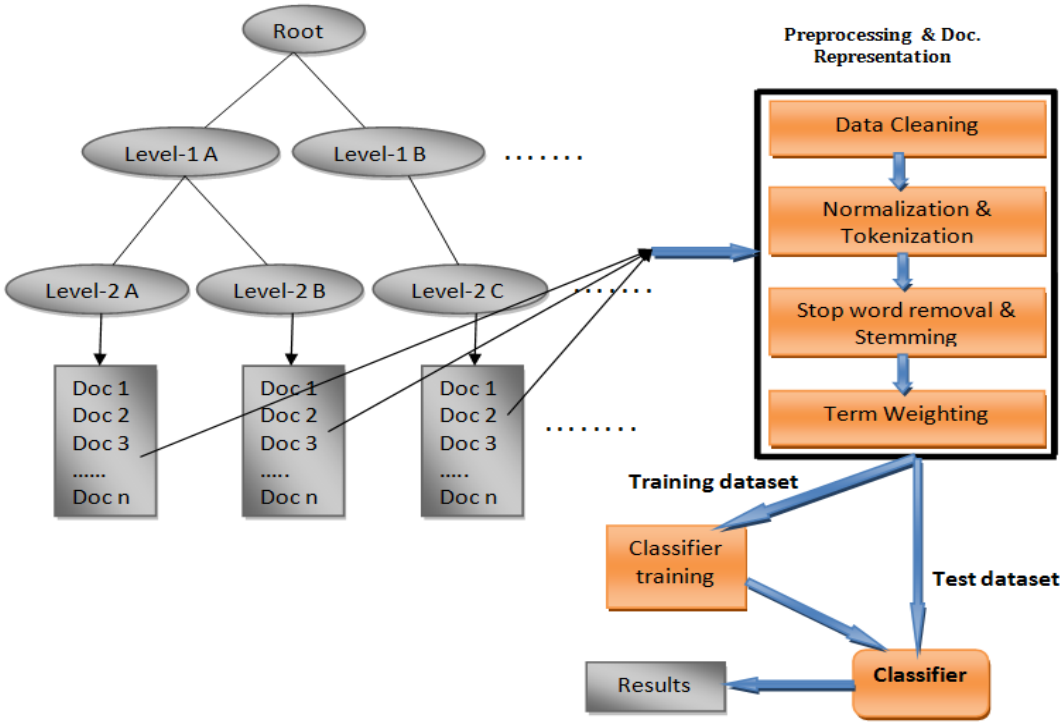

Fig.2. Architecture of the Hierarchical Amharic News Text Classifier

C. Support Vector Machine (SVM)

SVM is a method for supervised learning, applicable to both classification and regression problems. SVM uses a nonlinear mapping to transform the original training data into a higher dimension [12][13]. Within this new dimension, it searches for the linear optimal separating hyperplane (a decision boundary separating the instances of one class from another). Data from two classes can always be separated by a hyperplane, with an appropriate nonlinear mapping to a sufficiently high dimension. The SVM finds this hyperplane using essential instances from the training set called support vectors [11].

A growing number of machine learning methods have been applied to text categorization, such as Naïve Bayesian, Bayesian Network, Decision Tree, Neural network, Linear Regression, K-Nearest Neighbor, and Boosting. However, most machine learning methods overfit the training data when many features (high dimension vectors) are given [14]. Whereas, most researches [15][16] indicate that SVM is robust even when the number of features is large. Hence, SVM is selected for this study due to its capability of support high dimensional input space so that it can deal with large data sets; and tend to be less prone to over fitting

since the learned classifier is characterized by the number of support vectors rather than the dimensionality of the data.

D. Tools and Input File Preparation

Support Vector Machine (SVM), is a method for supervised learning, applicable to both classification and regression problems. It has been implemented using LibSVM multiclass tool. For a training set (x1,y1) ... (xn,yn) with labels yi in [1..k], it finds the solution of the following optimization problem during training.

$$\begin{aligned} &\min_{w,b,\xi} \quad \frac{1}{2}W^T W + C\sum_{i=1}^{l}\xi_i \\ &subject\ to \quad y_i(W^T\phi(0X_i)+b) \geq 1-\xi_i, \qquad (1) \\ &\qquad\qquad \xi_i \geq 0. \quad for\ all\ y\ in\ [1\ldots k] \end{aligned}$$

The data was prepared according to the requirement of LibSVM in the following format.

[line].=. [label] [index1]:[value1] [index2]:[value2] ...
[line] .=. [label] [index1]:[value1] [index2]:[value2] ...
…….

Where,

Label (target value): Sometimes referred to as 'class', the class (or set) of classification and its values are integers.

Index (feature number): Ordered indexes. Usually continuous integers

Value (feature value): The data for training which contains term weights.

E. Running LibSVM

SVMmulticlass consists of a learning module (svm_multiclass_learn) and a classification module (svm_multiclass_classify). The learning module takes the files to be learned. It learns the characteristics of the data and develops the model or classifier. It has the following format:

svm_multiclass_learn[options]training_example_file model_file

Where,

- *svm_multiclass_learn* is the learning module.
- *options* are the kernel functions and their parameters given to learning module to train the example file. Linear kernel and C=0.01 are some of the default values in LibSVM
- *training_example_file* is a file containing training instances (a file to be learned)
- *model_file* is the learned rule generated by the classification module using the selected parameters

In the other hand, the classification module can be used to apply the learned model to new examples. It has the following format:

svm_multiclass_classify [options] test_example_file model_file output_file

Where,

- *svm_multiclass_classify* is the classification module.
- *options* are functions and parameters
- *test_example_file* a file containing test instances (a file used to test a learned model).
- *model_file* the learned rule generated by the classification module on which the *test_example_file* is tested.
- *output_file* is a standard output of a classification/prediction result.

For all test examples in *test_example_file* the predicted classes (and the values of $x \bullet wi$ for each class) are written to *output_file*. There is one line per test example in *output_file* in the same order as in *test_example_file*. The first value in each line is the predicted class, and each of the following numbers is the discriminate values for each of the k classes. For e.g, given a testing file,

2 1:1.08889 2:2.1978 3:0.9634

the *output_file* results the following output

Class	*1*	*2*	*3*
Prediction 2	*-0.067107*	*0.112766*	*-0.045659*

SVM compares the prediction of each class and then the class with the maximum value is assigned to the test file. Thus, the above example shows that the test example is correctly predicted as class 2 among 3 classes.

F. Performance Measures

After models are trained by solving the above optimization problems, users can apply LibSVM to predict labels (target values) of test data. Let x1,….,xn be test data and f(x1),…., f(xn) be LibSVM's predicted decision values (target values for classification). If the true labels (target values) of test data are known and denoted as y1,….,yn, the predictions are evaluated by the following measures.

$$\text{accuracy} = \frac{\text{no of correctly classified data}}{\text{no of total data set}} \text{x}100 \qquad (2)$$

G. Training data

The number of news documents used in this experiment was 5100. Since hierarchical classification emphasizes the relationship among classes, rather than building single huge classifier, a classification is accomplished with the cooperation of classifiers built at each level of the tree. The training data is organized into 3 levels: from level-0(root level) to level-2. Each level represents classes or subclasses in a classification tree. Thus, there were 8 classes at level-0, 20 classes at level-1, and 69 classes at level-2 with at least 14 documents in them. The classifiers at each level were trained using the associated documents of all subclasses of that class. Thus, the level-0 classifier was trained using documents of all subclasses of that class from level-1 through 2. In contrast, each level-1 classifier was trained with documents from the appropriate level-1 subclasses up level-2.

H. Test data

We tested the accuracy of the classifier using the test data selected from each level-3 documents. These documents were excluded from the training process and were selected from different level-3 classes. Since the class from which the test documents were selected is known, the accuracy of the classifier is evaluated how often the classifier assign the test documents to the classes from which they originally came. Moreover, we used the accuracy of classification (see equation 2) as an evaluation measure.

V. Results and Discussions

The experiment has been done based on the concepts explained at section IV part (G) and (E) and the result is discussed as follows:

A. Effects of the number of classes and documents on flat classification

We created a single classification system by training a flat classifier for all classes in the top 3 levels of the classification tree, ignoring structure. In other words, each of the 97 classes was trained using 70% of the documents from each class. We had broken the classification process into pieces of classes taken separately at a time to see the performance of the classifier while increasing number of classes and documents (features). Since each document is assigned in the leaf node of the classification tree; level-0 classes will have the same number of documents as that of level-1 and level-2 when used separately for the next corresponding experiment. Hence, we selected documents using 50% of the collection to experiment on level-1 classes, and 70% of collection for the second experiment and 90% of the document collection for the third experiment.

From the above three experiments, it was found that the accuracy decreases when the number of classes increased from 8 to 20 and then to 69; and the number of documents increased from 2550 to 3570 and then to 4590. Moreover, the average accuracy obtained from the above three experiments was 65.58%, which were decreased at each steps of the experiment. This shows that as the number of classes and documents increase, the performance of a flat classifier decreases. This is because, as the number of documents is increasing, the number of support vectors increases. Since the classification is done in a multidimensional plane where we can draw a number of hyper planes, the increasing number of support vectors causes to narrow the margin between these hyper planes. The smaller the marginal hyper plane then causes maximum classification error on unseen test instances apart from the difficulty to get the maximum marginal hyper plane (MMH).

B. Effects of Number of Top Features on Flat Classification

All the 97 classes and total documents in the collection were used to see the performance of the flat classifier at increasing number of top features which were extracted from the test documents. Thus, the top features up to 20 words were considered where the features were selected based on their *tfidf* weights. The peak accuracy in this experiment was 68.84 % when the top 3 features were used. This means that least number of features has high discriminating power among classes than more number of features, which are found across many classes. Figure 3 shows effects of top features on flat classifier.

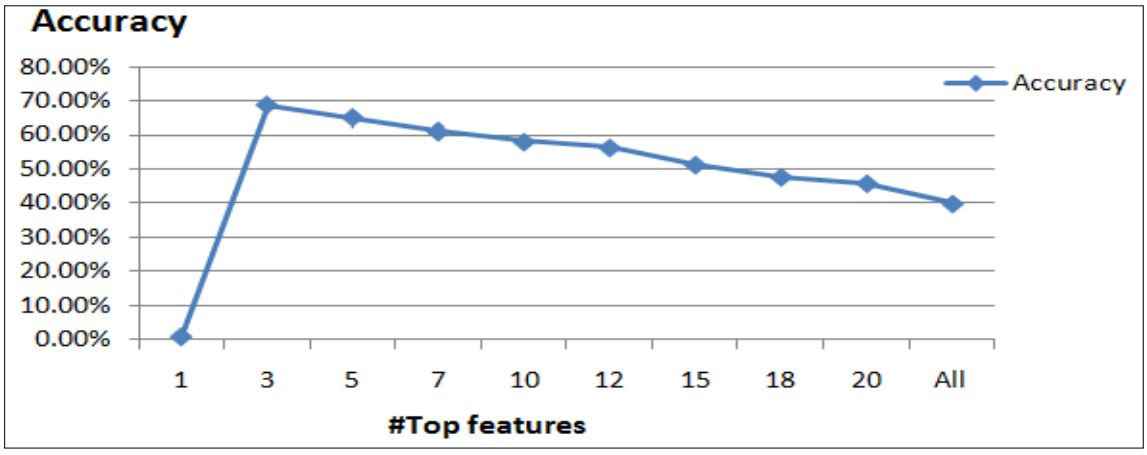

Fig.3. Effects of the number of top features on the performance of flat classification

C. Effects of the number of classes and documents on hierarchical classification

For the hierarchical classifier, we constructed a set of classifiers, one at each level of the classification tree. Thus, there was one classifier at level-0 (trained on the 8 level-1 classes), 8 classifiers for level-1 (one for each level- 1 class), and 20 classifiers for level-2(one for each level-2 class).

Table 3 shows the performance of hierarchical classifier as it improves down the hierarchy for randomly selected classes (Education (code 2), Health (code 4) & Politics (code 6)). This is because each classifier deals with the documents associated to only that class or

subclasses of that class and it concentrates on a smaller set of documents, those relevant to the task at hand.

Table 3. Results of the hierarchical classifier along the hierarchy for sample classes

Classifier		Training set	Testing set	Accuracy (%)
Level-0 Classifier		3570	1530	63.03
Level-1 Classifier	Code 2	394	168	78.76
	Code 4	399	170	81.155
	Code 6	595	256	79.76
Level-2 Classifier	Code 2.1	163	70	87.93
	Code 2.2	161	68	90.37
	Code 2.3	70	30	88.23
	Code 4.1	176	75	85.71
	Code 4.2	33	14	89.37
	Code 4.3	190	81	86.01
	Code 6.1	475	204	82.62
	Code 6.2	120	52	85.56

D. Effects of Number of Top Features in Hierarchical Classifiers

The documents were initially classified at level-0 using a varying number of features per document where the features were selected based on their *tf*idf* weights. The first run used only the highest weighted feature for classifying the documents and number of features was increased in each subsequent run until a maximum of 20 features. The level-0 classifier had a peak accuracy of 81.50% when the top 5 features were used.

The test documents were then classified at level-1 while again varying the number of top features from 1 to 20. At level-1, the classification process is same as above, but it is constrained to consider only the subclasses of the best matching class at level-0. Hence, the level-1 classifier had a peak accuracy of 85.07% when the top 10 features were used.

Finally, the test documents were classified at level-2 with the classification process now constrained to consider only the subclasses of the best matching class at level-1. Since all the test documents originally came from level-2 classes, the accuracy of the classifier overall is best judged by the accuracy at level-2. The level-2 classifier had an exact match precision of 89.06% when the top 15 features were used. This means that, from a set of 97 classes, the hierarchical classifier correctly classified 89.06% of documents to their original class. The overall results of the experiment are shown in figure 4 below.

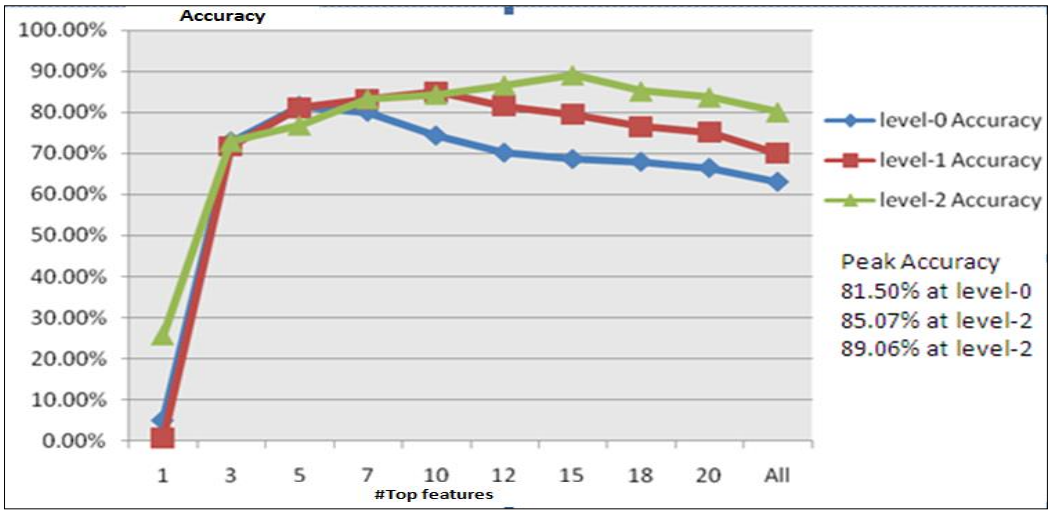

Fig.4. Effects of Number of Top Features in Hierarchical Classifiers

It is interesting to note that, as we move down the hierarchy, the classifiers perform better with more features extracted from test documents. This is because they need more information in order to make finer-grained distinctions between the classes.

VI. Conclusion

As the advancements of technology, the mass production Amharic documents necessitated proper classification to meet the information needs of users. Hierarchical Amharic text classification approach shows good result in classifying documents into their predefined categories. Since, a hierarchical approach manages a huge collection of document in a divide-and-conquer approach; it simplifies a classification task focus on a smaller set of documents, those relevant to the task at hand. The experimental result also shows that the use of hierarchy for text classification results in a significant improvement of 29.42 % in exact match accuracy over the flat classifier.

References

[1] Rennie, Jason D. M. Improving Multi-Class Text Classification with Naive Bayes. Massachusetts Institute of Technology, Masters Thesis, 2001.

[2] Klein, B. Text Classification Using Machine Learning. Journal of Theoretical and Applied Information Technology. 2004.

[3] D'Alessio, S., Murray, K., Schiaf_no, R., & Kershenbaum, A. The effect of using hierarchical classifiers in text categorization. Proceeding of RIAO-00, 6 International Conferences, 2000.
[4] Koller & Sahami. Hierarchically classifying documents using very few words. The 14th national conference on machine learning. Computer Science department, Stanford University , 1997
[5] Ethnologue. 2004. Languages of the World, 14th Edition.
[6] Yohannes A. Automatic Amharic news text classification using Support Vector Machine approach. Department of Information Science, Addis Ababa University, Master's Thesis, 2007.
[7] Zelalem Sintayehu. Automatic Classification of Amharic News Items: The Case of Ethiopian News Agency. School of Information Studies for Africa, Addis Ababa University, Addis Ababa, 2001.
[8] Surafel Teklu. Automatic categorization of Amharic news text: A machine learning approach. Department of Information Science, Addis Ababa University, Masters Thesis, 2003.
[9] Worku K. Automatic Amharic News Text Classification: A Neural network approach. Department of Information Science, Addis Ababa University, Master's Thesis, 2009.
[10] Nega, Stemming of Amharic Words for Information Retrieval university of Sheffield, Sheffield, UK , 2002.
[11] J. Han and M. Kamber. Data Mining: Concepts and techniques (2nd ed.). Morgan Kaufmann Publishers, 2006.
[12] Yuchen Fu,Yuanhu Cheng. "Application of an integrated support vector regression method in prediction of financial returns". International Journal of Information Engineering and Electronic Business (IJIEEB), Vol.3, No.3, June 2011.
[13] Muhsin Hassan, et al." Reducing Support Vector Machine Classification Error by Implementing Kalman Filter", International Journal of Intelligent Systems and Applications(IJISA), Vol. 5, No. 9, August 2013.
[14] Le Hoang,et al. "Image Classification using Support Vector Machine and Artificial Neural Network", International Journal of Information Technology and Computer Science(IJITCS), Vol. 4, No. 5, Mayl 2012.
[15] F. Sebastiani. Machine learning in Automated Text Categorization-in ACM Computing surveys 34(1), 2002, pages 1-47.
[16] T. Joachims, Text categorization with support vector machines: learning with many relevant features, *Proceedings of ECML 98, 10th European Conference on Machine Learning* (Chemnitz, Germany, 1998). Pages 137–142.

Authors' Profiles

Mr. Alemu Kumilachew obtained his M.S.c degree in Information Science from Addis Ababa University and B.S.c degree in Information Technology from Jimma University, Ethiopia. He is currently working as a lecturer at the faculty of computing, Bahir Dar University. His area of research interests include information systems and retrieval, Artificial intelligence, question answering systems, machine learning.

Mr. Adane Nega obtained his M.S.c degree in information technology from university of Gondar and B.S.c degree in computer science from Bahir Dar University, Ethiopia. He is currently working as a lecturer at the faculty of computing, Bahir Dar University. His area of research interests include Artificial intelligence and soft computing, Data mining, big data analysis, machine learning.

Mr. Tamir Anteneh obtained his M.S.c degree in information Science from Addis ababa University and B.S.c degree in Information Technology from Bahir Dar University, Ethiopia. He is currently working as a lecturer at the faculty of computing, Bahir Dar University. His area of research interests include Information retrieval, machine leaning, cloud computing, Databases and knowledge based systems.

How to cite this paper: Alemu Kumilachew Tegegnie, Adane Nega Tarekegn, Tamir Anteneh Alemu,"A Comparative Study of Flat and Hierarchical Classification for Amharic News Text Using SVM", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.9, No.3, pp.36-42, 2017. DOI: 10.5815/ijieeb.2017.03.05