ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2015.0136

COMPARISON OF PERFORMANCES OF DIFFERENT SVM IMPLEMENTATIONS WHEN USED FOR AUTOMATED EVALUATION OF DESCRIPTIVE ANSWERS

C. Sunil Kumar[1] and R.J. Rama Sree[2]
[1]*Research and Development Center, Bharathiar University, India*
E-mail: sunil_sixsigma@yahoo.com
[2]*Rashtriya Sanskrit Vidyapeetha, India*
E-mail: rjramasree@yahoo.com

Abstract

In this paper, we studied the performances of models built using various SVM implementations during the multiclass classification task of automated evaluation of descriptive answers. The performances were evaluated on five datasets each with 900 samples and with each of the datasets treated using symmetric uncertainty feature selection filter. We quantitatively analyzed the best SVM implementation technique from amongst the 17 different SVM implementation combinations derived by using various SVM classifier libraries, SVM types and Kernel methods. Accuracy, F Score, Kappa and Area under ROC curve are used as model evaluation metrics in order to evaluate the models and rank them according to their performances. Based on the results, we derived the conclusion that SMO classifier when used with Polynomial kernel is the overall best performing classifier applicable for auto evaluation of descriptive answers.

Keywords:

Descriptive Answers, Auto Evaluation, SVM, LibLINEAR, LibSVM, SMO, Kernels

1. INTRODUCTION

Evaluation of answers and providing a scoring is a hard classification task (assigning a single category to each document) where in the human evaluator or the system is supposed to interpret the answer and classify the answer into one of the possible rubrics pre-allocated for the answer. We believe supervised learning method can be applied to classify the answers into appropriate rubric based on the likelihood suggested by training samples. The supervised learning process requires extracting various text features from the documents meant as training set and then train using a sophisticated machine learning algorithm. Based on the experiments and measurements from our previous researches, it was found that Support vector machines (SVM) - LibLINEAR out performs Naive Bayes, Logistic Regression, Random Forests, Support Vector Machine, Decision Stump and Decision Tree supervised machine algorithms when used for automated evaluation of descriptive answers. Our research about effects of dimensionality reduction on automated evaluation of descriptive answers also revealed that using symmetrical uncertainty attribute evaluation filter yields better prediction accuracies than not using dimensionality reduction therefore we want to continue using symmetrical uncertainty attribute evaluation for this current research as well. While our earlier research focused on evaluating SVM – LibLINEAR with other classifiers, in this current research we want to explore LibLINEAR SVM, LibSVM and SMO implementations and their performances in correctly classifying our answers datasets. LibLINEAR classifies linearly separable data through a hyperplane with maximum distance from the identified support vectors however this can be accomplished through various SVM types such as L2-regularized L2-loss support vector classification (dual), L2-regularized L2-loss support vector classification (primal) etc., Similarly SMO and LibSVM uses a technique called kernels where non-linearly separable data is transformed into linearly separable data by projecting the data into a large dimension plane. There are various types of kernels too such as polynomial kernel; sigmoid kernel etc., Also, LibSVM comes with its own SVM type. We have keen interest to observe how these various implementations, SVM types and kernels affect the performance of the models therefore the current research.

The rest of this paper is organized as follows. Section 2 discusses literature review. Section 3 is the data used, experimental setup, the preliminaries of the tools and techniques used in this paper. Section 4 describes the models built and measurements made during the experiments. Finally, analysis of results, concluding remarks and further research plans are indicated in section 5.

2. LITERATURE REVIEW

While there is enormous amount of literature available on the details of SVMs and their applications in text mining area, it is interesting that there is no literature available that compares the various SVM techniques specific to automated grading of descriptive answers. It is important to derive a general principle in terms of which of these techniques to use for text classification especially in the context of automated grading of descriptive answers. Deriving a general principle eliminates the need to repeat the testing of the kernel techniques every time there is a need to perform the task. Also, it is observed that there is no literature available demonstrating the application of the SVMs to the automated evaluation of descriptive answers domain. These gaps identified are addressed through the research covered under this research paper therefore making the aspects covered under the research paper very unique from the existing literature and a significant contribution to the existing knowledge.

3. EXPERIMENTAL SETUP

The setup in which the experiments are conducted for this paper are specified in this section.

3.1 DATA COLLECTION

In February 2012, The William and Flora Hewlett Foundation (Hewlett) sponsored the Automated Student

Assessment Prize (ASAP) [10] to machine learning specialists and data scientists to develop an automated scoring algorithm for student-written essays. As part of this competition, the competitors are provided with hand scored essays under 8 different prompts (questions). 5 of the 8 essays prompts are used for the purpose of this research.

All the graded essays from ASAP are according to specific data characteristics. All responses were written by students of Grade 10. On average, each essay is approximately 50 words in length. Some are more dependent upon source materials than others. The number of training essays for each prompt varies. For example, the lowest amount of training data is 1,190 essays, randomly selected from a total of 1,982. The data contains ASCII formatted text for each essay followed by one or more human scores, and (where necessary) a final resolved human score. Where it is relevant, more than one human score exists, so as to signify the reliability of the human scorers [11]. For the purpose of evaluation of the performance of the model, we considered the score predicted by the model to comply with the final resolved human score in training example.

The data used for training, validation and testing the models are answers written by students for 5 different questions. Data for a question is considered as one unique dataset. So, we have a total of 5 datasets. The questions that students are asked to provide responses to are from diversified fields of Chemistry, English Language Arts and Biology.

3.2 DATA CHARACTERISTICS

In each of the 5 training data sets used for our research, the training set is 900 samples in size. Our previous research for determining appropriate sample size for automated essay scoring using SMO revealed that using 900 samples for training proved to yield slightly better results than using other sample sizes therefore the decision to use 900 samples as the training sample size [1].

3.3 WEKA WORKBENCH

For the purpose of designing and evaluating our experiments, we have used a machine learning workbench called Weka. Weka (Waikato Environment for Knowledge Analysis) is a free offering from University of Waikato, New Zealand. This workbench has a user-friendly interface and it incorporates numerous options to develop and evaluate machine learning models [2, 3]. These models can be utilized for a variety of purposes, including automated essay scoring.

All experiments performed were executed on a Dell Latitude E5430 laptop. The laptop is configured with Intel Core i5 - 3350M CPU @ 2.70 GHz and with 4 GB RAM however Weka workbench is configured to use a maximum of 1 GB. The laptop runs on Windows 7 64 bit operating system.

3.4 STATISTICAL FEATURE EXTRACTION

Below features are focused on from input training data set to build feature table:

- Unigrams - An n-gram of size 1 is referred to as a “unigram”.
- Bigrams - An n-gram of size 2 is a “bigram” (or, less commonly, a “digram”).
- Trigrams - An n-gram of size 3 is a “trigram”.
- Stop words - The most common, short function words, such as the, is, at, which, and on.
- Stemming - It is a process of reducing inflected (or sometimes derived) words to their stem, base or root form-generally a written word form. Porter stemmer is used for stemming purpose.
- Punctuations - unigrams representing things like periods, commas, or quotation marks.

Included features - Unigrams, Bigrams, Trigrams and Stemming.

Excluded features -Stop words, Punctuations.

3.5 SYMMETRICAL UNCERTAINTY (SU) ATTRIBUTE EVALUATION BASED FEATURE SELECTION

SU filter evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class. SU filter Entropy represents a measure of the system’s randomness. Entropy is generally represented by H which stands for the Greek Alphabet Eta. This attribute evaluation filter compensates for the bias in Information gain [4]. It does this by dividing the Information gain by entropies obtained on the class and the attribute as shown in Eq.(1) .

$$SU(Class, Attribute) = \frac{2*(H(Class) - H(Class|Attribute))}{H(Class) + H(Attribute)} \quad (1)$$

More information on this filter can be found in [6].

Once the filter is applied, we observed significant reduction in the number of features that are really meant for relevant and non-redundant use for model building. The reduced features and original features prior to feature selection in the datasets are shown in Table.1.

Table.1. Reduction of number of features on attribute selection filter application on datasets

Data Set	Number of features with no Attribute selection applied	Number of features with Symmetrical uncertainty filter applied
1	25190	254
2	22847	126
3	29475	400
4	20915	378
5	19599	373

3.6 MODEL BUILDING

For all the five datasets treated through feature selection filter, models are built using LibLINEAR, LibSVM and SMO.

LibLINEAR is an open source library and a family of linear SVM classifiers for large scale linear classification which supports logistic regression and linear SVM. A detailed description can found in [5].

LibSVM is a classical SVM implementation based on the original work of Vapnik; a detailed description of this algorithm can be found in [6, 7].

SMO is a Sequential Minimal Optimization principle based SVM method, introduced by Platt in 1997 [8].

We identified that for multiclass classification task that is on hand for this research, there are 17 combinations possible through LibLINEAR, LibSVM and SMO options. The combinations are derived through usage of various SVM types and kernels as appropriate. The Table.2 shown below lists the combinations.

In Table.2 above, we named each of the classifier - SVM - kernel type combinations with a classifier nickname. This nickname will be the one used to report various measurements throughout the rest of this paper. This arrangement is made for ease of reporting.

4. MODELS BUILT AND MEASUREMENTS

Various models are built during the experiments, the measurements obtained and various conclusions made through analysis of the measurements done during the experiments are described in this section.

Models are built on Weka workbench, we used randomized 10-fold cross-validation in order to test the performance the models.

We made measurements under two broad categories named calibration scores and discriminatory scores. Calibration scores measure whether the model assigns the correct class value to the test instances. Many of these scores can be computed solely from the confusion matrix obtained from the result of the classifications done by the model. Discriminatory scores measure how good can the prediction model separate instances with different classes are called discriminatory scores [9, 12].

Under the calibration scores umbrella, accuracy, F score and kappa are compared for the datasets across the classifier - SVM - kernel types. Area under the ROC curve is captured as part of discrimination of models.

Table.2. Various combinations of Classifiers, SVMs and kernel combinations used for this research

Classifier Nickname	Classifier	SVM Type	Kernel Type	Reference number used for the classifier through rest of the paper
LibLINEAR1	LibLINEAR	L2-regularized L2-loss support vector classification (dual)	Not applicable	1
LibLINEAR2	LibLINEAR	L2-regularized L2-loss support vector classification (primal)	Not applicable	2
LibLINEAR3	LibLINEAR	L2-regularized L1-loss support vector classification (dual)	Not applicable	3
LibLINEAR4	LibLINEAR	Support vector classification by Crammer and Singer	Not applicable	4
LibLINEAR5	LibLINEAR	L1-regularized L2-loss support vector classification	Not applicable	5
LibSVM1	LibSVM	C-SVC	Linear	6
LibSVM2	LibSVM	C-SVC	Polynomial	7
LibSVM3	LibSVM	C-SVC	Radial Basis Function	8
LibSVM4	LibSVM	C-SVC	Sigmoid	9
LibSVM5	LibSVM	nu-SVC	Linear	10
LibSVM6	LibSVM	nu-SVC	Polynomial	11
LibSVM7	LibSVM	nu-SVC	Radial Basis Function	12
LibSVM8	LibSVM	nu-SVC	Sigmoid	13
SMO1	SMO	Not applicable	Normalized Polynomial	14
SMO2	SMO	Not applicable	Polynomial	15
SMO3	SMO	Not applicable	Pearson VII function-based universal	16
SMO4	SMO	Not applicable	Radial Basis Function	17

Accuracy is measured by percentage of correctly predicted instances by the model divided by the total number of instances [13]. Accuracy is given by the formula shown in the Eq.(2).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \qquad (2)$$

The terms TP, TN, FP, FN in Eq.(2) and Eq.(3) stands for True positives, true negatives, false positives and false negatives respectively.

The Table.3 shows the accuracies measured from the models built and tested through 10 fold cross validation across datasets.

The F score measures accuracy using the statistics precision p and recall r [14]. Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives to all actual positives (tp + fn). The F score is given by Eq.(3) shown below.

$$F\ Score = \frac{2PR}{(P + R)}\ where\ P = \frac{TP}{(TP + FP)}\ and\ R = \frac{TP}{(TP + FN)} \qquad (3)$$

The Table.4 shows the F Scores measured from the models built and tested through 10 fold cross validation across datasets.

Table.3. Accuracies measured and ranking of the models built using various classifier - SVM type - kernel type combinations

Classifier Nickname	Dataset 1	Rank	Dataset 2	Rank	Dataset 3	Rank	Dataset 4	Rank	Dataset 5	Rank
LibLINEAR1	56%	7	66%	1	76%	3	85%	3	90%	1
LibLINEAR2	57%	3	66%	1	76%	3	85%	3	90%	1
LibLINEAR3	56%	7	64%	6	77%	1	85%	3	90%	1
LibLINEAR4	57%	3	65%	3	75%	7	85%	3	90%	1
LibLINEAR5	57%	3	65%	3	76%	3	84%	8	90%	1
LibSVM1	57%	3	64%	6	76%	3	86%	2	90%	1
LibSVM2	32%	17	53%	13	54%	17	76%	15	83%	15
LibSVM3	52%	12	53%	13	65%	12	81%	9	84%	12
LibSVM4	50%	16	53%	13	55%	15	78%	13	84%	12
LibSVM5	56%	7	63%	10	67%	11	81%	9	89%	8
LibSVM6	52%	12	47%	17	55%	15	45%	17	79%	16
LibSVM7	56%	7	63%	10	69%	10	79%	12	89%	8
LibSVM8	56%	7	64%	6	56%	14	61%	16	49%	17
SMO1	58%	2	65%	3	75%	7	80%	11	86%	11
SMO2	60%	1	64%	6	77%	1	87%	1	90%	1
SMO3	52%	12	61%	12	70%	9	85%	3	87%	10
SMO4	51%	15	53%	13	58%	13	78%	13	84%	12

Table.4. F Score measured and ranking of the models built using various classifier - SVM type - kernel type combinations

Classifier Nickname	Dataset 1	Rank	Dataset 2	Rank	Dataset 3	Rank	Dataset 4	Rank	Dataset 5	Rank
LibLINEAR1	0.56	5	0.622	1	0.752	4	0.837	4	0.888	7
LibLINEAR2	0.563	4	0.622	1	0.752	4	0.834	5	0.893	4
LibLINEAR3	0.554	9	0.594	6	0.763	2	0.834	5	0.894	2
LibLINEAR4	0.559	7	0.604	4	0.751	7	0.844	3	0.897	1
LibLINEAR5	0.567	2	0.612	3	0.752	4	0.83	8	0.893	4
LibSVM1	0.565	3	0.588	7	0.762	3	0.853	2	0.891	6
LibSVM2	0.155	17	0.372	14	0.383	17	0.656	15	0.759	16
LibSVM3	0.46	13	0.372	14	0.618	12	0.766	11	0.779	13
LibSVM4	0.428	16	0.372	14	0.388	16	0.712	14	0.764	15
LibSVM5	0.555	8	0.583	9	0.666	10	0.81	9	0.88	8
LibSVM6	0.503	12	0.465	13	0.552	14	0.494	17	0.785	12
LibSVM7	0.549	11	0.573	11	0.688	9	0.794	10	0.878	9
LibSVM8	0.552	10	0.583	9	0.564	13	0.642	16	0.579	17

SMO1	0.56	5	0.601	5	0.732	8	0.761	12	0.815	11
SMO2	0.595	1	0.584	8	0.767	1	0.864	1	0.894	2
SMO3	0.456	14	0.543	12	0.661	11	0.831	7	0.846	10
SMO4	0.432	15	0.372	14	0.46	15	0.714	13	0.777	14

Kappa statistic is used to measure the agreement between predicted and observed categorizations of a dataset, while correcting for an agreement that occurs by chance. However, like the plain success rate, it does not take costs into account. Better models will have Kappa closer to 1 [15].

The Table.5 shows the Kappas measured from the models built and tested through 10 fold cross validation across datasets.

Area under the receiver operating characteristics curve (AUC) is a single scalar that represents models performance based on two dimensional ROC representations. A perfect model will have an AUC value of 1 where as a random guessing model will have a value of 0.5. Further details on AUC can be found in [16].

The Table.6 shows the AUCs measured from the models built and tested through 10 fold cross validation across datasets.

Table.5. Kappa measured and ranking of the models built using various classifier - SVM type - kernel type combinations

Classifier Nickname	Dataset 1	Rank	Dataset 2	Rank	Dataset 3	Rank	Dataset 4	Rank	Dataset 5	Rank
LibLINEAR1	0.4125	7	0.3581	1	0.5572	6	0.5717	5	0.6083	7
LibLINEAR2	0.4152	5	0.3581	1	0.5569	7	0.5637	6	0.6275	3
LibLINEAR3	0.4123	8	0.3119	6	0.5792	2	0.5633	7	0.6308	2
LibLINEAR4	0.4149	6	0.3325	4	0.5575	5	0.5838	3	0.6416	1
LibLINEAR5	0.424	2	0.3397	3	0.5594	4	0.5523	8	0.6236	5
LibSVM1	0.4208	3	0.3033	7	0.5772	3	0.6063	2	0.6167	6
LibSVM2	0	17	0	14	0	17	0	17	0	17
LibSVM3	0.3388	13	0	14	0.316	12	0.3561	11	0.1319	13
LibSVM4	0.3081	16	0	14	0.0057	16	0.1883	15	0.0372	16
LibSVM5	0.4107	9	0.2923	9	0.3955	11	0.5074	9	0.5781	8
LibSVM6	0.3515	12	0.1159	13	0.2088	14	0.0161	16	0.243	12
LibSVM7	0.4019	11	0.2756	11	0.4369	9	0.4597	10	0.573	9
LibSVM8	0.4065	10	0.2923	9	0.2127	13	0.2035	14	0.0664	15
SMO1	0.4188	4	0.3218	5	0.5311	8	0.3348	12	0.2678	11
SMO2	0.4613	1	0.2985	8	0.5862	1	0.6374	1	0.6266	4
SMO3	0.3242	14	0.246	12	0.4009	10	0.5774	4	0.5097	10
SMO4	0.3169	15	0	14	0.0971	15	0.2038	13	0.1156	14

Table.6. AUC measured and ranking of the models built using various classifier - SVM type - kernel type combinations

Classifier Nickname	Dataset 1	Rank	Dataset 2	Rank	Dataset 3	Rank	Dataset 4	Rank	Dataset 5	Rank
LibLINEAR1	0.703	10	0.657	3	0.776	7	0.791	5	0.822	5
LibLINEAR2	0.705	7	0.657	3	0.775	8	0.786	6	0.83	4
LibLINEAR3	0.704	8	0.633	8	0.788	2	0.786	6	0.831	3
LibLINEAR4	0.704	8	0.644	6	0.777	5	0.792	4	0.848	1
LibLINEAR5	0.709	5	0.648	5	0.777	5	0.782	8	0.82	6
LibSVM1	0.707	6	0.628	9	0.787	3	0.797	3	0.816	7
LibSVM2	0.5	17	0.5	15	0.5	17	0.5	17	0.5	17
LibSVM3	0.665	15	0.5	15	0.648	12	0.667	11	0.553	15
LibSVM4	0.65	16	0.5	15	0.502	16	0.588	15	0.517	16
LibSVM5	0.702	11	0.624	10	0.691	11	0.773	9	0.803	10
LibSVM6	0.674	14	0.554	13	0.601	13	0.516	16	0.646	11

LibSVM7	0.697	13	0.617	12	0.711	10	0.741	10	0.806	9
LibSVM8	0.7	12	0.624	10	0.598	14	0.634	13	0.557	14
SMO1	0.79	2	0.669	1	0.782	4	0.647	12	0.608	12
SMO2	0.808	1	0.666	2	0.81	1	0.823	1	0.839	2
SMO3	0.749	4	0.638	7	0.731	9	0.807	2	0.814	8
SMO4	0.752	3	0.529	14	0.577	15	0.607	14	0.57	13

Table.7. Time taken to build the models using various classifier - SVM type - kernel type combinations

Classifier Nickname	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
LibLINEAR1	0.32	0.08	0.17	0.13	0.08
LibLINEAR2	0.06	0.02	0.04	0.05	0.04
LibLINEAR3	0.14	0.04	0.11	0.08	0.07
LibLINEAR4	0.48	0.13	0.2	0.15	0.14
LibLINEAR5	0.07	0.01	0.04	0.04	0.03
LibSVM1	0.33	0.21	0.25	0.18	0.2
LibSVM2	0.31	0.12	0.25	0.19	0.22
LibSVM3	0.29	0.17	0.26	0.2	0.19
LibSVM4	0.32	0.15	0.26	0.24	0.28
LibSVM5	0.3	0.2	0.39	0.32	0.16
LibSVM6	0.27	0.12	0.15	0.12	0.14
LibSVM7	0.33	0.19	0.23	0.17	0.16
LibSVM8	0.33	0.19	0.19	0.14	0.16
SMO1	1.16	1.33	0.97	0.93	0.73
SMO2	0.4	0.24	0.27	0.24	0.18
SMO3	0.7	0.55	0.66	0.5	0.37
SMO4	0.58	0.37	0.55	0.32	0.21

Time to build the models is another important factor that is also captured across the models for comparison purposes. However, we observed that all models were built within our self-imposed threshold value of 1 minute therefore we did not include the time to build models as the criteria for ranking. Also, Linear classifiers such as LibLINEAR does not use any kernel as the data is assumed to be linearly separable whereas the other kernel based classifiers need to transform the data into a high dimensional plane so as to ensure linear seperability in the data . This additional step of transforming the data into higher dimension plane obviously will take additional time therefore we decided to eliminate the time taken to build the model from the criteria of model evaluation.

The Table.7 shows the time taken to build the models through various SVM implementations.

For the measurements captured, in order to objectively compare the performance of various models built, we ranked each the measurements separately using Rank.EQ excel function [17]. The ranking is done across each of the five datasets comparing across the 17 different algorithms used for this research purpose. Ranks are assigned in descending order i.e., the highest value in the comparison range gets the rank 1 and the lowest value gets the last rank.

Post ranks were obtained we summed the ranks across all the evaluation criteria by datasets. Again, we applied the Rank.EQ excel function to rank the overall sums obtained but this time the ranks were assigned in ascending order i.e., the lowest sum in the comparison range gets rank 1 and the highest range gets the last rank. Fig.1 is the comparison of various measurements, overall rank sums obtained across datasets and by SVM implementation. X axis shown on the left side of the Fig.1 is the scale to compare the various measurements. Y axis shown on the bottom of the Fig.1 is the SVM types referenced from Table.2. The axis on the right hand side of Fig.1 is the scale to show the final ranks.

Finally, the Algorithm with the rank 1 is concluded as the best performing SVM classifier - Kernel combination.

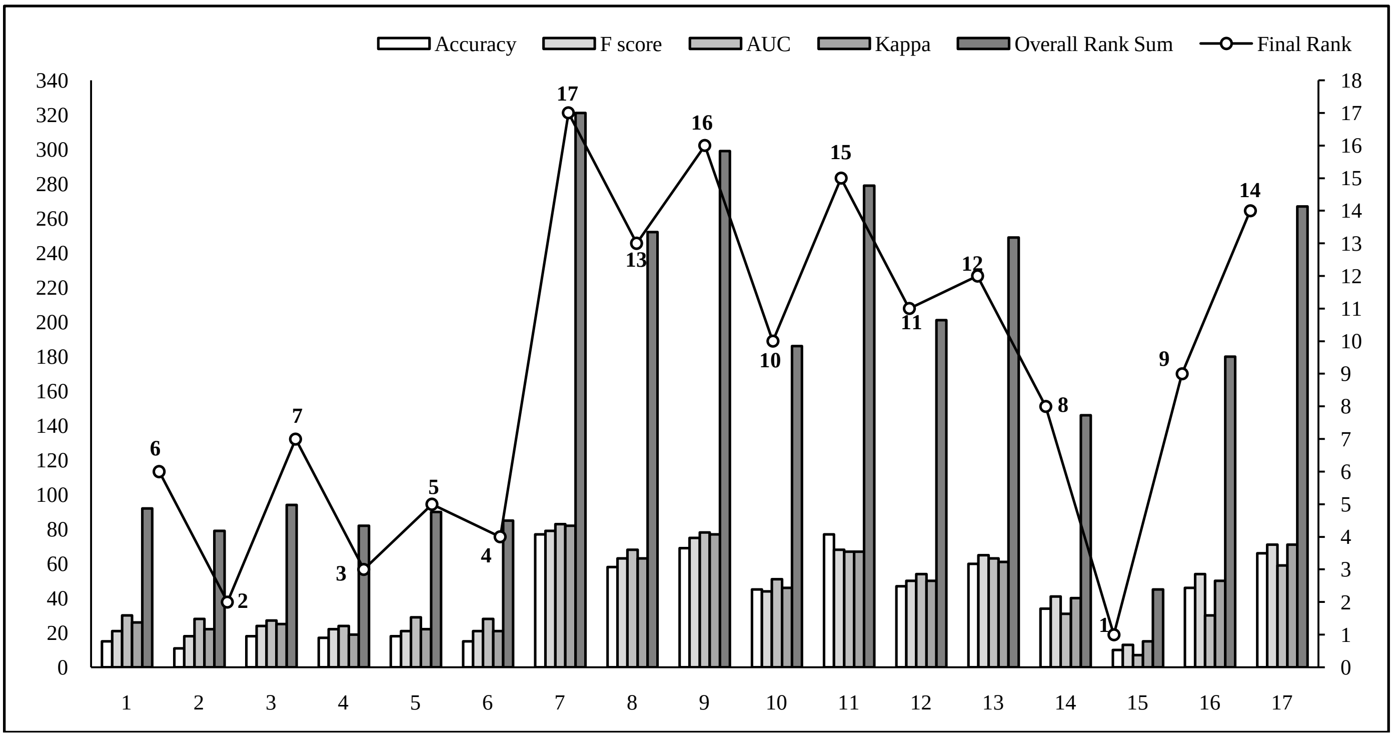

Fig.1. Sum of ranks obtained on all datasets and ranking of the models built using various classifier – SVM type – kernel type combinations

5. RESULTS DISCUSSIONAND NEXT STEPS

5.1 CONCLUSIONS DERIVED FROM RESULTS

From the measurements and ranking tables shown in the previous section, it is very evident that SMO2, LibLINEAR2 and LibLINEAR3 are in the top three positions. SMO2 is a SMO classifier that uses polynomial kernel and from the experiment this classifier - kernel combination is chosen as the best SVM to use on datasets meant for automated evaluation of descriptive answers.

5.2 FUTURE DIRECTIONS

In this paper we were able to apply various SVM implementations to datasets to identify the best SVM technique. Further research is required to apply dimensionality reduction techniques such as Principal component analysis and perform feature transformation to verify if the model's performance can be improved. Wrappers are another feature selection technique similar to filters applied in this paper however wrappers are optimized to use with a specific classification algorithm. Now that SMO - Polynomial kernel is confirmed as the best classification algorithm to use, Wrappers is one area to research and confirm if the classification performances can be improved. Ensembling techniques such as bagging, boosting, stacking etc., for classification performance improvement is another prospect for further research.

REFERENCES

[1] Sunil Kumar and R. J. Rama Sree, "Experiments towards determining best training sample size for automated evaluation of descriptive answers through sequential minimal optimization", *ICTACT Journal on Soft Computing*, Vol. 4, No. 2, pp. 710 -714, 2014.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H. Witten, "The WEKA Data Mining Software: An Update", *ACM SIGKDD Explorations Newsletter*, Vol. 11, No. 1, pp. 10-18, 2009.

[3] Ian H. Witten and Eibe Frank, "*Data Mining: Practical Machine Learning Tools and Techniques*", Second Edition, Morgan Kaufmann, 2005.

[4] Lei Yu and Huan Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 856-863, 2003.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification", *Journal of Machine Learning Research*, Vol. 9, pp. 1871-1874, 2008.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, Article. 27, 2011

[7] V. Vapnik, "*The Nature of Statistical Learning Theory*", Second Edition, Springer-Verlag New York, 1995.

[8] John C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Technical Report MSR-TR-98-14, 1998.

[9] Scikit learn, “Model evaluation: quantifying the quality of predictions”, http://scikit-learn.org/ stable/modules/ model_evaluation.html

[10] Kaggle, http://www.kaggle.com/c/asap-ses

[11] Evaluation, http://www.kaggle.com /c/asap-sas/details /evaluation, 2012.

[12] Orange, “Method scoring”, http://orange.biolab.si/ docs/latest/ reference/rst/ Orange.evaluation.scoring/

[13] Machine Learning Corner, “Evaluation of Classifier’s Performance”, http://mlcorner.wordpress.com /2013/04/30/ evaluation-of-classifiers-performance/, 2013.

[14] Kaggle, “Mean FScore, http://www.kaggle.com /wiki/MeanFScore.

[15] Steve Simon, “What is Kappa coefficient (Cohen's Kappa)”, http://www.pmean.com/definitions/kappa.htm.

[16] kaggle.com, “Area under the curve”, https://www.kaggle.com/wiki/AreaUnderCurve.

[17] RANK.EQ function, http://office.microsoft.com/en-in/excel-help/rank-eq-function-HP010335687.aspx, 2014