2011 International Conference on Telecommunication Technology and Applications
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

Text Classifiers for Cricket Sports News

Tarik Zakzouk and Hassan Mathkour

[1] Ph.D Candidate, College of Computer and Information Sciences, King Saud University (KSU)
[2] Professor, College of Computer and Information Sciences, King Saud University (KSU)

Abstract. Application of Text classification has become an essential part of our daily life. Many algorithms based on Machine Learning (ML) has been tested in the past several decades with different levels of success. This field has since shown maturity and new class of tools has been commonplace. This is a revisit of this field using both commodity software and hardware to show the efficiency and effectiveness of a group of four ML-based methods in classifying Cricket sports news articles.

Keywords: Topical Text Classification, SVM, PERCEPTRON, Decision Tree, Sports Corpus.

1. Introduction

1.1. The Text Classification Process

The classical Text Classification (TC) approach has 5 main steps namely: Feature Extraction (FE), Feature Selection (FS), Feature Valuation (FV), Feature Generation (FG) which has not been applied in our experiments, and Model Generation. The input to this process is the text to be classified shaped as a set of documents named a corpus. The corpus documents are assumed to be structured into folders, the data is ready for classification. The TC stages work as follows: Reading the files, Tokenization (which is equivalent to Feature Extraction), Stop Word removal, stemming, further word filtering and vectorization (Feature Selection), and the Parameter settings for each (word weighting, ranking, …) (Feature Valuation). split, … etc). Then generating the document vectors that feed into the classification algorithm. The Model Generation stage consists mainly of two parts, training and testing. The training part consists of the classification algorithm which generates the required model while the testing part consists of two sub-processes: applying the model and measuring its performance. After the model is generated and perfected, it is being stored as a binary file for further use on different data sets.

1.2. Building the Corpus:

Many corpora (document collections) are available online for research purposes such as Reuters (different variations), OHSUMED …etc. The source of such corpora is the Internet. Building a corpus from scratch used to be a challenging task [4]. We have chosen to build our own corpus to achieve several goals even beyond the scope of this paper. SGSC (Saudi Gazette Sports Corpus) is a textual sports news corpus specifically built for research purposes [6]. It has been motivated by Sports, Medicine, and Religion news classification results [1]. Compared to famous text corpora such as Reuters, it has small size (comprised of only 797 news texts) specialized only in sports. The news text size is found to be between 0.5 KB up to 4.7 KB (3 to 40 lines and between 20 to 900 words not including the title). The following table summarizes the SGSC:

Table 1. SGSC Summary

No. of Documents	Total no. of Words	Without Stop Words	Stemmed
797	18,087	17,770	13,632

[1]Tel.:+966-505-206308; fax:+966-1-491-3634; E-mail address: tzakzouk@gmail.com
[2]E-mail address:binmathkour@yahoo.com

Within a text news web page, there is very limited meta-data such as the title, author or the news agency. The author (if mentioned) has no fixed location. It is sometimes placed right after the title or at the end of the text. Pictures included are mainly non-relevant to the actual news text. The date is found at the main page URL. The most important missing information is the sports name. The process of building SGSC is described in [6] and has been both manual and slow. Later on, the process has been tested for semi-automation using a multi-class text classifier based on SVM. A folder is manually created for each day and the news text is mapped (copy and paste) to a single text file having the name as the date-stamp of the news and additional two digits at the end identifying the order of appearance on the website from 1 to 19.

After downloading and organizing 2 months of news, all files have to be manually classified and copied into new folders representing the sports they address. This has resulted into having 22 different sports news folders ranging from Cricket (with the highest number of news articles around 180) to Swimming with only one article. This shows that sports coverage of Saudi Gazette is not balanced perhaps due to the nature of the readers segment. More folders were added later to create negative examples for each sport with adequate number of examples. Cricket was chosen to be the sports of choice since it had the majority of the news articles (178 positive examples making 22% of the corpus). Separate binary classifiers were built based on 11 different machine learning techniques for the purpose of getting the best model.

The remaining of this paper is organized as follows: in section 2, the experiment of building the text classifiers is explained in detail, section 3 describes the result of each classification method, section 4 compares the results of such classifiers, and finally, section 5 gives the summary.

2. The Binary Text Classifiers for Cricket in SGSC

The setup for building the classifiers (models) is somehow comprehensive and needs careful design. It involves selecting and setting up the right machine learning algorithms, setting up the cases for both learning and testing, and choosing the right measurements for the experiment.

2.1 Text Classification Methods

Rapidminer 5, the data mining tool used in this experiment, provides a wide variety of classification methods. We have chosen 4 for our experiment. Two different SVM implementations namely SVMLight and mySVM in addition to Decision Trees and Neural Net (Perceptron).

2.2 Generating the Models

Model application was separated from learning to be able to apply to different cases. Models generated from the learning step were applied on fresh data. Results were very promising. In certain cases where feature selection was applied, model application was not straight forward and the new fresh data had to be similarly treated to be able to be handled by the model with some workarounds. For each technique, the plain case (no feature selection) is first applied and then three different combinations of feature selection techniques were employed namely:

- Stop Word Removal,
- Stop Word Removal + Porter Stemming,
- Stop Word Removal + Porter Stemming + Selecting Top 10% Chi-Square Weight Features

A fifth case, aggressive feature selection (Top 1% Chi-Square Feature Weighting) was also tested for some SVM techniques but later abandoned due to weak results. For each experiment, four text classification effectiveness measures were produced in addition to others such as:

- Total time it takes to produce the mode (end to end)
- Size of the produced model (in terms of MB).
- No. of features selected (Dimensions)
- Algorithm specifics such as No. of support vectors for SVM based models.

2.3 Corpus Characteristics:

Two folders with one containing positive examples (Cricket documents) with 178 articles and the other containing negative examples (non-Cricket) with 154 articles. Both have the following characteristics:

Table 2. Corpus Characteristics

No. of Documents	No. of Words	Without Stop Words	Stemmed
332	11,264	10,849	8,371

2.4 The Training/Testing Split:

To produce the model, the data sets (positive and negative) had to be separated into two parts: training set and the testing set. The tool allows the user to control the split either by ratio or by number of examples. Although we tried several splits but we settled with the tool's default ratio which is 70/30 which resulted in generally better classification effectiveness measured by accuracy, precision, recall, and F-Measure.

2.5 Effectiveness Measures:

Four effectiveness measures have been selected. They depend on the confusion matrix (performance matrix) output, which are:

- True Positive (TP): Number of documents correctly assigned to a class
- False Positive (FP): Number of documents incorrectly assigned to a class
- True Negative (TN): Number of documents correctly not assigned to class
- False Negative (FN): Number of documents belong to a class but incorrectly assigned to other

The text classification effectiveness measures used in this study are:

- **Precision** (P) = TP / (TP + FP)
- **Recall** (R) = TP / (TP + FN)
- **Accuracy** (A) = (TP + TN) / (TP + TN + FP + FN)
- **F-Measure** (Micro-averaging) = 2 . (P . R) / (P + R)

These effectiveness measures are available by the tool and need to be selected to be calculated at the end result.

3. MODEL CONSTRUCTION

The following is the results of each one. Note that each experiment was run three times and the results were averaged. The tool had to be restarted after running two cases to avoid crashing.

SVM1: This classifier is based on Java implementation of mySVM by Stefan Ruping. The linear kernel was selected since text categorization examples were found to linearly separable by nature [5]. The four experiments results are summarized in the following table:

Table 3. SVM1 Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	82.76	100	90.57	90	35.9	11264	17
+Stop Words	81.36	100	89.72	89	33.2	10849	17
+Stemming	84.21	100	91.43	91	26.3	8371	13
+Chi-Square	100	100	100	100	4.3	837	11

The numbers in table 3 show that recall does not get affected from any feature selection technique. Precision however, does get affected positively by Stemming and slightly negatively by removing stop words only. By selecting only 10% of the stemmed non-stop words, it shows 100% and does that in 65% of the time. The

model size shrinks by using any feature selection technique and it becomes almost 14% of the original size of the plain case by using all of them. Also, it was observed that the number of support vectors is constant in all four experiments at 332 which is total the number of examples.

SVM2: Based on LibSVM by [2]. Default settings were left untouched such as dot (linear) kernel and cache size. The four experiments results are summarized in the following table:

Table 4. SVM2 Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	100	97.92	98.95	99	33.4	11264	21
+Stop Words	100	97.92	98.95	99	30.4	10849	12
+Stemming	100	97.92	98.95	99	23.9	8371	9
+Chi-Square	100	100	100	100	3.1	837	12

It was found that precision is not affected by any feature selection technique. (Opposite to the first SVM technique). On the other hand recall is positively affected by chi-square feature selection. Both time and especially size of the resultant model benefit from all feature selection used. The number of support vectors varies by case. In the plain case, a total of 252 support vectors were identified. The second case had 257 (slightly increased), the third case decreased to 236, while the last case the number decreased to 118 support vectors.

Decision Trees: This algorithm was employed with its default values such as gain-ration as the criterion and having 4 as the minimal size for tree splitting. The following are the results of the four cases:

Table 5. Decision Tree Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	97.87	95.83	96.84	97	24.9	11264	01:36:00
+Stop Words	97.87	95.83	96.84	97	23.9	10849	01:46:52
+Stemming	100	100	100	100	18.5	8371	00:48:48
+Chi-Square	100	100	100	100	2	837	37

Only stemming and Chi-Square feature selection improve effectiveness measures. The former half's the time to learn while with the later, time shrinks from almost two hours to less than a minute. The model size is directly proportionate to the number of dimensions.

PERCEPTRON: It is simply a Neural Network linear classifier based. The settings are rounds = 3 and learning rate = 0.05 which are the default settings. The following are the results of the four cases:

Table 6. PERCEPTRON Experiment Results

Combination	Precision %	Recall %	F-Measure%	Accuracy %	Model Size	Dimensions	Time
Plain	55.81	100	71.64	62	61.9	11264	10:24
+Stop Words	55.17	100	71.11	61	58	10849	11:12
+Stemming	55.81	100	71.64	62	47	8371	08:02
+Chi-Square	55.81	100	71.64	62	73.2	837	00:56

PERCEPTRON is almost feature selection blind. Effectiveness measures are the same. Time is the only factor that is positively affected and it is directly proportional to the number of dimensions. The surprise is that size increases by using Chi-Square feature selection instead of shrinking.

4. DISCUSSION

By taking the F-Measure effectiveness measure we construct the following table for all algorithms used in the experiment:

Table 7. F-Measure Effectiveness

Combination	SVM1	SVM2	Decision Tree	PERCEPTRON
Plain	90.57	**98.95**	96.84	71.64
+SW	89.72	**98.95**	96.84	71.11
+Stem	91.43	98.95	**100**	71.64
+Chi-2	**100**	**100**	**100**	71.64

- All but the PERCEPTRON are solid performers with accuracy well over 90%. We have to analyze the best algorithm by case;
- In the plain case (no feature selection) SVM2 and Decision Tree lead the pack with over 96 % F-measure.
- With stop-word removal SVM2 and Decision Tree also are clear winners with F-Measure over 96%
- Stemming has the most effect on Decision Trees at 100% f-measure. SVM2 and SVM1 values slightly decrease.
- By employing a 10% Chi-Square Feature selection 3 algorithms namely SVM1, SVM2, and decision trees have f-measure at 100%. PERCEPTRON is unchanged.

Such high values of f-measure do come at an expense in both execution (learning) time and resultant model size. Observe the following two tables:

Table 8. Performance (Times)

Combination	SVM1	SVM2	Decision Tree	PERCEPTRON
Plain	**00:17**	00:21	01:36:00	10:24
+SW	00:17	**00:12**	01:46:52	11:12
+Stem	00:13	**00:09**	00:48:48	08:02
+Chi-2	**00:11**	00:12	00:00:37	00:56

- Based on the discussion on F-Measure, we combine it with the timing to determine the best effectiveness/performance ratio for each case:
- The plain case, SVM2 and Decision Tree had the best F-Measure. However, the F-Measure/Time ratio shows that SVM1 a clear winner.
- The Stop-Word Removal case: SVM2 has the best F-Measure and the best time which is 12 seconds
- The stemming case: decision Trees had 100% F-Measure but at a heavy price. It requires 48 minutes and 48 seconds to execute vs. 9 seconds for SVM2.
- The Chi-Square Feature Selection case: SVM1, SVM2, and Decision Trees have perfect F-Measure scores but with the following timings: 11, 12, and 37 seconds respectively. SVM1 and SVM2 have the best timings.

- By a quick look at the size of the resultant models, SVM2 has the smallest model foot-print than SVM1 in all cases. This result is consistent with several previous studies [3]. It is also important to mention that feature selection is not recommended with PERCEPTRON.

5. CONCLUSION

Four binary text classifiers were built to test the cricket class of SGSC. Their effectiveness was measured using four classical measures namely, Precision, Recall, Accuracy, and F-Measure. Additional measures such as time and model size were discussed to find the most suitable algorithm. 3 variations of feature selection cases were performed along with a plain case. SVM2 which is based on LibSVM lead the pack with best performance/effectiveness ratios overall. This experiment has been undertaken on a Pentium dual core 4 GB laptop running UBUNTU Linux 10.04. More powerful HW has been tested using a 4-core CPU

laptop having 8 threads which showed potential scalability. This experiment also demonstrates that such experiments are possible using COTS and open-source SW running on mainstream HW to conduct what used to be specialized controlled only experiments.

6. ACKNOWLEDGMENT

The authors wish to thank the research center in the College of Computer and Information Sciences and the College of Computer and Information Sciences, King Saud University. We also thank for Eng. Mohammad Amin for his support and knowledge using RapidMiner.

7. References

[1] Al-Harbi, S, Almuhareb, A, Al-Thubaity, A, Khorsheed, N and Al-Rajeh, A. Automatic Arabic Text Classification. JADT 2008.

[2] Chang, C.-C and Lin, C.-J. LIBSVM: a library for support vector machines, 2001, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] Lewis, D, Yang, Y, Rose, T and Li, F. RCV1: A new Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5 (2004): 361 – 397.

[4] Rose, T, Stevenson and M, Whitehead, M. The Reuters Corpus Volume 1 – from Yesterday's News to Tomorrow's Language Resources. Reuters.

[5] Yang, Y and Pedersen J. A Comparative Study on Feature Selection in Text Categorization. In *International Conference on Machine Learning*, pages 412–420, 1997.

[6] Zakzouk, T and Mathkour, H. Building A Corpus for Text Classification. The 2010 International Conference on Intelligent Network and Computing (ICINC 2010).