Co-Training Implementation for Objectivity Detection

Segura Tinoco, Andrés and Cuevas Saavedra, Vladimir
Computer Engineering
Universidad de Los Andes
Bogotá – Colombia
{ga.segura ,ve.cuevas121}@uniandes.edu.co

July 2018

Abstract - In the current era in which we live, many sports articles are published daily on the Internet, by various authors, which are often written objectively, on other occasions subjectively, which may not please the reader or change your perception of the facts. In the present work, we perform a detection analysis of objectivity to a set of 1000 sports articles, previously labeled using the Mechanical Turk tool from Amazon. For this, we conducted 2 experiments in which the predictive ability of using trained statistical models with supervised versus trained learning with semi-supervised learning was compared. The fact of learning from the original file tagged by Amazon Mechanical Turk and one generated with the TMG+ algorithm based on TF-IDF was also evaluated. The results obtained were very encouraging, since the SL approach generated better results for the original tagged file (precision close to 82.9%), while the SSL approach using Co-Training, generated better results for the dataset created with the own algorithm, with accuracy close to 74.4% using 50% of the data tagged for SSL.

Keywords: Machine Learning, Semi-Supervised Learning, Naive Bayes, SVM, Random Forest, Gradient Boosting, Co-Training, Objectivity Detection, Subjectivity Analysis, TF-IDF.

I. INTRODUCTION

The objectivity analysis over articles pretend to detect and predict the writer criteria, the intention is to determine if the article is objective (based on facts or the object) or subjective (based on opinions or the subject). Everyday this kind of analysis is becoming more common on academics or businesses and are the base to create powerful tools that helps readers (to trust articles content) or editors.

Therefore, the intention of this document is to present the results of a subjectivity/objectivity analysis of sport articles with different statistic models; these sport articles are written by different news agencies, it is common to find subjectivity in this topic.

For this, we will explore 2 training approaches such as Supervised Learning and Co-Training on 2 different datasets, which were generated from the same set of sports articles through 2 very different processes: dataset 1 was generated using Amazon Mechanical Turk (crowdfunding) while dataset 2 was generated by a algorithm written in Python, develop by us (called TMG+), that involves the TF-IDF algorithm, the punctuation, stopwords counting, among custom personalization for each output.

The implementation of the algorithms for the classifiers as well as the generation of results ware done with the statistical language R 3.5.1 [1] and the RStudio IDE.

II. DATA DESCRIPTION

On project definition, we analyzed a couple of alternatives, we understood that we required to work on a project related to machine learning with a general topic, this means that we should be able to understand the data easily to avoid adding a learning curve because of the topic complexity. We search over a list of repositories and the dataset that we picked was related to sport articles from the Machine Learning Repositories of the University of California, Irvine (UCI).

Dataset 1

The dataset 1, has two kinds of files, sports articles and features of the same sports articles. The sports articles are a thousand of text files, where each of them contains one sport article in a raw format (plain text). The features file is an excel file composed of a thousand records/rows where each of them correspond to one of the sport articles mentioned before, it is also composed of a series of attributes (59) and a label that states if the article is objective or not.

The features file was manually labeled using Amazon's crowdsourcing tool, Mechanical Turk (MTurk), the UCI does not provide many details of this process. Some of the features are retrieved using the Stanford POS tagger and the tags are as defined in Penn Treebank Project

The information related to the fifty-nine attributes can be found in the Machine Learning Repositories web page (UCI, n.d.).

Dataset 2

Two datasets were created from the raw data (the 1000 sport articles) using term frequency–inverse document frequency (TFIDF) numerical statistic using Python, the difference between them is that one of them was processed with stemming. These datasets are CSV files composed of 5742 and 5231 attributes for simple and stemmed version respectively, these numbers are variable because attributes are auto-generated by the Term Matrix Generator process, more information on this can be found on the next section (Methods and Algorithms).

III. DEFINITION OF TECHNIQUES AND ALGORITHMS

The selection of the algorithm to solve a specific problem is a critical task that requires time and knowledge, since often non-functional requirements must be considered such as: the size, quality and nature of the data, the variables to be selected, the amount of data tagged, the purpose of the algorithm, the implementation, accuracy and performance of this, etc.

Therefore, given our scenario, where we have as initial data, multiple quantitative input variables (different for each dataset) and a single qualitative output variable, we chose to use Supervised Statistical Learning techniques to solve the problem, focusing the solution in the prediction of articles that have been written subjectively, through the learning and classification of a set of characteristics registered for them.

Below, we summarize the most important algorithms that we use, to offer a better understanding of these algorithms and their results:

Support Vector Machine (SVM) is a maximum margin classification algorithm based on the theory of statistical learning. SVM performs classification tasks by maximizing the margin that separates both classes while minimizing classification errors, through a hyperplane. SVM can use different kernels such as: linear, polynomial, radial and sigmoid [4]

Random Forest constructs a series of decision trees such that each tree depends on the values of a randomly tested vector independently and with the same distribution for each of these. When constructing these decision trees, whenever a division is considered in a tree, a random sample of m predictors is selected as candidates from the full set of p predictors. A fresh sample of m predictors is taken in each division, and we typically choose $m \approx \sqrt{p}$, that is to say, the number of predictors considered in each division is approximately equal to the square root of the total number of predictors. Random forests promote divisions by considering only a subset of the predictors for each decision tree [4].

Gradient Boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed by Jerome H. Friedman simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett and Marcus Frean. The latter two papers introduced the view of boosting algorithms as iterative functional gradient descent algorithms. That is, algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification [5].

Term Matrix Generator+ (TMG+) Because of the requirements and the co-training approach that we are using, a program was created to generate a second dataset that can be personalized according to the requirements and the topics. This algorithm takes raw files, read them, clean the text, stemming can be applied if required, TF-IDF is calculated for all the words on each document. From the results a Top 10 is generated for each file using unigrams only, once all Top 10 are obtained a matrix is created showing the frequency of the word on each document. The matrix follows these orders for each row: document x vocabulary x frequencies.

The algorithm supports arguments where we can specify the name of the output file that will be a CSV file. As it can be seen, even though for generating the Top 10 unigrams were used, once the Top K is gotten, the 10 words with frequencies for each document can be considered an n-gram (n = 10).

Figure 1 shows the workflow that is followed to extract the attributes and generate the CSV that then is used by the classifiers.

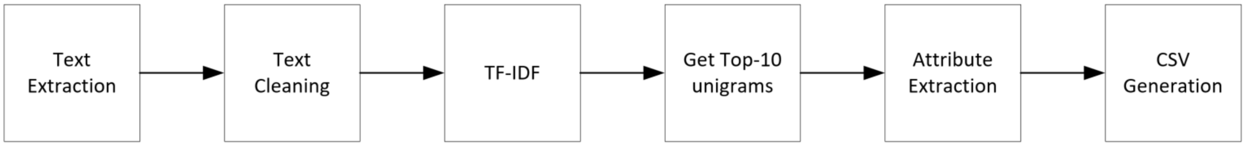

Figure 1. Term Matrix Generator - Workflow of automated TF-IDF word attribute generation.

Co-Training is a type of semi-supervised learning (SSL), which can be used when there are two sets of independent and compatible attributes for the data (two views of the data), where each set of data is sufficient to train a classifier. Because two independent and redundant sets of tagged attributes cannot always be found, another option is to train two different classifiers for the same training set. The main idea behind this technique is to train a classifier with each subset of attributes and use it to classify data for the other classifier and thus increase your training set. The Co-Training algorithm that will be used in this work is:

```
Given: Labeled data L, unlabeled data U
Loop:
    • Train C1 using L
    • Train C2 using L
    • Allow C1 to label p positive, n negative examples from U
    • Allow C2 to label p positive, n negative examples from U
    • Add these self-labeled examples to L.
```

IV. RELATED WORKS

Objectivity/subjectivity analysis is as very active domain of research these days where social networks and information has become relevant and easy to transfer because of the current advances in technology and communication.

Even though the topic of this project is sports articles, the approach can be extended to other domains, e.g other types of entertainment, politics, stock market investments. We find various machine learning approaches for subjectivity classification. We'll mention a few of the findings relevant to this study.

A Subjectivity Classification Framework for Sports Articles using Improved Cortical Algorithms (Nadine Hajj, Yara Rizk, Mariette Awad, PhD American University of Beirut, Beirut, Lebanon), May 2018.

A team of PhD from Lebanon created a framework that tested the same database/dataset that we are using on our project achieving testing accuracy of 85.6% on a 4'fold cross validation with 40% reduction in features using cornical algorithm (CA*).

The contribution made by them are: The sports articles' dataset for subjectivity analysis used in (Rizk and Awad, 2012) was extended to include a larger number of diverse articles.

They express that there have been several approaches to classify objectivity/subjectivity that have included NLP, use of SENTIWORDNET (this one uses

SVM and Rocchio classifiers), use of n-grams (n=1, 2 and 3), Naïve Bayes, semantic orientation count. All of these approaches were classifying phrases or words.

For our project, we are classifying whole documents, using attributes, frequencies, TF-IDF and n-grams.

Per the following two researches, current models can run in real-time, e.g **Wang et al. (2012)** developed a Twitter sentiment analysis system that can process 72 tweets per minute and correctly classify 59% of the tweets into positive, negative, neutral and unsure. Based on twitter, **Guerra et al. (2011)** also adopted a transfer learning approach that develop real-time sentiment analysis system applied to tweets and produced prediction accuracies between 80% and 90%.

Yang et al. (1997) conducted a comparative study of five feature selection techniques in the context of text categorization using: document frequency (DF), information gain (IG), mutual information (MI), chi-square test (CHI) and term strength (TS). In this study, DF was the most computationally efficient with acceptable performance, while other techniques performed similarly better at the expense of an increased complexity. This is another interesting point because we are using TF-IDF instead of just the DF.

V. EXPERIMENTAL RESULTS

Experiment 1 – SL vs SSL for Dataset 1: In this first experiment, the pre-processed data was used by Amazon Mechanical Turk, which is the features.csv file (also called Dataset 1). After an initial exploratory analysis, it was found that Dataset 1 contains 1000 records, with 58 input variables (X) and an objective variable (Y) called Label. The 2 classes of the Label variable are well represented, since the number of records labeled as objective is 635, while the number of records labeled as subjective is 365. In addition, the data has 100% completeness for all characteristics (variables).

After the process of data exploration, we proceeded to create and analyze a first scenario of Supervised Learning (SL), creating 6 different models of Machine Learning from the tagged data. The models used are: KNN (K-nearest neighbor) with a k equal to 5, SVM with linear kernel, Naïve Bayes, Decision Trees, Random Forests with 200 internal trees and Ada Boosting. To measure the learning accuracy, the confusion matrix was calculated and the Cross-Validation technique was used with 5 folds, and to avoid that randomness influenced the results, Cross-Validation was executed 5 times. The results were averaged and can be seen in table 1.

Table 1. Average precision of the models for experiment 1 - first scenario.

Model	***Accuracy***	***Error***	***# Objective***	***# Subjective***	***% Objective***	***% Subjective***
Base Line	**63.50**	**36.50**	**635**	**365**	**100.00**	**100.00**
SVM Linear	82.82	17.18	580	248	91.37	67.95
KNN	77.56	22.44	536	240	84.35	65.75
Bayes	79.60	20.40	563	233	88.60	63.95
Decision Tree	79.58	20.42	553	243	87.06	66.58
Random Forest 200	**82.80**	**17.20**	**559**	**269**	**87.97**	**73.81**
Ada Boost	80.84	19.16	549	259	86.46	71.07

From the previous table, the following information can be highlighted: all models achieved an overall accuracy greater than the baseline, highlighting the model generated by the SVM algorithm with 82.82% accuracy; in counterpart, the model generated at from KNN had the worst accuracy with 77.56%. In addition, the SVM model had the highest accuracy predicting the records classified as targets, however, the model that predicted with greater accuracy, classifying the records as subjective, was Random Forest 200 with 73.81% followed by Ada Boost with 71.07 %. No other model exceeded the threshold of 70%.

Therefore, since the models that offered the best results were those generated from SVM and Random Forest algorithms, in the scenario we proceeded to explore other variants of them and eliminate algorithms that offered worse results from the experiment, these were KNN and Decision Tree. In scenario 2 the models used were: Naïve Bayes, SVM with linear kernel, SVM with polynomial kernel, Random Forests with 200 trees, Random Forests with 300 trees and Ada Boosting. This time the Cross-Validation with 10 folds was used and the learning was executed 10 times and the results were averaged again, which can be seen in table 2.

Table 2. Average precision of the models for experiment 1 - second scenario.

Model	***Accuracy***	***Error***	***# Objective***	***# Subjective***	***% Objective***	***% Subjective***
Base Line	**63.50**	**36.50**	**635**	**365**	**100.00**	**100.00**
Bayes	79.90	20.10	567	232	89.29	63.56
SVM Linear	83.40	16.60	583	251	91.75	68.88
SVM Poly	73.53	26.47	497	238	78.33	65.18
Random Forest 200	**82.89**	**17.11**	**557**	**272**	**87.78**	**74.38**
Random Forest 300	82.75	17.25	558	270	87.81	73.95
Ada Boost	81.04	18.96	545	265	85.86	72.66

From the second SL scenario, it was obtained that the models that continued to offer the best precision were SVM with linear kernel and Random Forest, with very similar results for the models generated with 200 and 300 decision trees. The precision of the SVM model with polynomial kernel was greatly degraded with respect to that obtained with the linear kernel (radial and sigmoid kernel tests were also performed but the results were even worse). Therefore, the 2 models that learned better from the data of View 1 and had the highest prediction accuracy were SVM Linear and Random Forest. However, since Random Forest 200 had the highest percentage of cases correctly predicted as subjective, it was the model selected to be used in experiment 3 (co-training approach).

In addition, it was observed that the results practically do not change when 5 iterations and 5 folds are used versus 10 iterations and 10 folds, in counterpart, the learning process took longer (approximately 4 times more). Therefore, for the next experiments, only 5 iterations and cross-validation with 5 folds were used.

In the last phase of this experiment, we proceeded to evaluate for the same dataset with a Semi-Supervised Learning (SSL) approach and Co-Training, and later, to compare the results with those obtained by the Random Forest 200 SL model. The algorithm used in the Co-Training for the 2 algorithms was Gradient Boosting.

To correctly evaluate the effect of the Co-Training, several combinations of dataset sizes were made (5 in total), with the following percentage from the total data remaining in each subset: training {10%, 20%, 30%, 40%, 50%} and tests {90%, 80%, 70%, 60%, 50%}. In addition, each of the 5 sub-experiments of Co-Training was repeated 5 times and the results of the overall precision of the model were averaged, which can be seen in the following table (table 3).

Table 3. Gradient Boosting Accuracy with Co-Training for Dataset 1.

	Size of Training Data				
Gradient Boosting with CT	**10%**	**20%**	**30%**	**40%**	**50%**
Global Accuracy	76.60	77.11	78.13	78.43	78.72

When the results obtained by the Co-Training approach against those obtained by the best model of the Supervised Learning approach were compared, it was observed that, although they are very good, it did not exceed the overall accuracy obtained by the Random Forest algorithm with 200 trees. In addition, the results of the Co-Training approach were very constant regardless of the amount of tagged data that was used to learn. This may be because the data in each sub-experiment was stratified (they maintained the 63.5-37.5% relationship between the objective / subjective labels). Figure 2 shows the comparison between the average global precision of the 2 training approaches (SL vs. SSL).

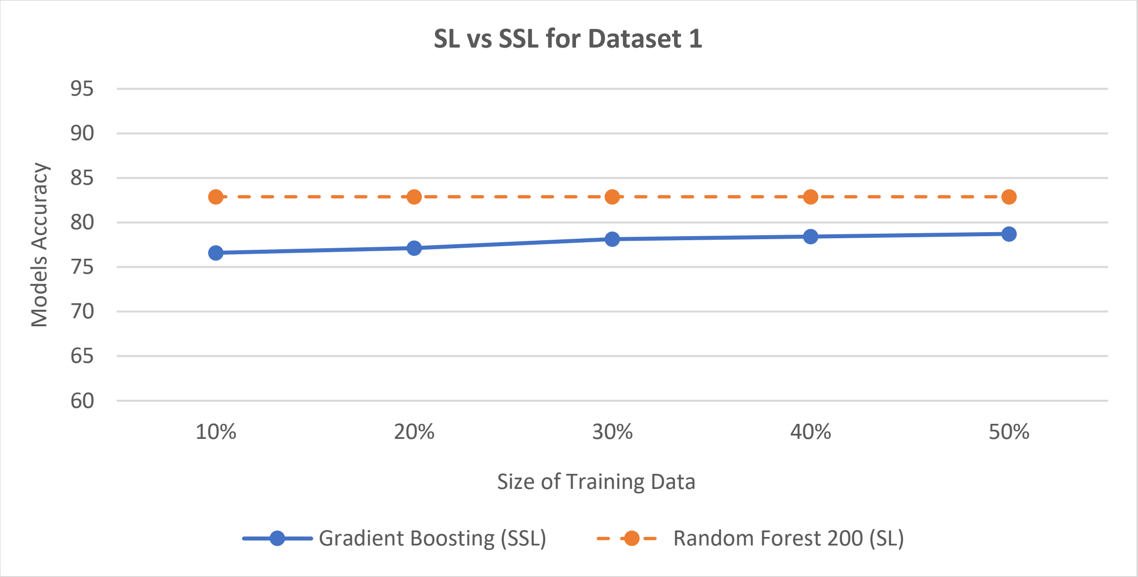

Figure 2. SL vs SSL for Dataset 1.

It is concluded that both approaches are useful to perform objectivity analysis on sports articles, however, better results are obtained using Supervised Learning for the Dataset 1.

Experiment 2 – SL vs SSL for Dataset 2: To create an optimal machine with supervised learning from the Dataset 2, it was initially tested with the same algorithms used in the first scenario of experiment 1, however, totally different results were obtained. The only algorithm that gave good results was SVM, varying per the kernel used. The other algorithms generated models with very low accuracy, biased models (they only predict a single class well) or simply models that do not converge, that is to say, they take very long learning times or never finish learning.

Tests were also carried out with both the simple Dataset 2 and the stemmed. With this variant, the results were very similar, obtaining slightly better accuracies when the stemmed dataset 2 is used for learning. Table 4 shows the average results obtained by the different SVM models generated for Dataset 2.

Table 4. Average precision for models of experiment 2.

Model	*Accuracy*	*Error*	*# Objective*	*# Subjective*	*% Objective*	*% Subjective*
Base Line	**63.50**	**36.50**	**635**	**365**	**100.00**	**100.00**
SVM Linear	68.94	31.06	502	187	79.06	51.34
SVM Radial	**72.52**	**27.48**	**503**	**222**	**79.18**	**60.93**
SVM Polynomial	70.44	29.56	507	198	79.81	54.14
SVM Sigmoid	37.70	62.30	333	44	52.47	12.00

As can be seen in the previous table, the model that generated the best results both in terms of global precision and the prediction of subjectively written articles is the one created with the SVM algorithm with the Radial kernel.

Just as in the previous experiment, we proceed to evaluate the Semi-Supervised Learning approach for Dataset 2. In this experiment the Gradient Boosting was used again as an algorithm and the data for training and tests was divided using the same percentages: training {10%, 20%, 30%, 40%, 50%} and tests {90%, 80%, 70%, 60%, 50}. Each of the 5 sub-experiments of Co-Training was repeated 5 times and the results of the overall precision of the model were averaged, which can be seen in the following table.

Table 5. Gradient Boosting Accuracy with Co-Training for Dataset 2.

	Size of Training Data				
Gradient Boosting with CT	**10%**	**20%**	**30%**	**40%**	**50%**
Global Accuracy	63.00	67.29	68.21	73.46	74.40

The results of applying Co-Training for Dataset 2 were totally different from those obtained in experiment 1, since the Gradient Boosting with CT trained with 40% and 50% of tagged data improved the overall accuracy obtained by the SVM algorithm with kernel Radial trained with SL. Furthermore, in this experiment, the accuracy of the model was not constant as the number of labeled records increased, but rather the model's accuracy improved.

Figure 3 shows the comparison between the average global precision of the 2 training approaches (SL vs. SSL). It is concluded that for Dataset 2 created from the most significant unigrams (selected by TF-IDF) of each sports article, it is more effective to use Co-Traning to predict subjectively written articles than to use a traditional SL approach.

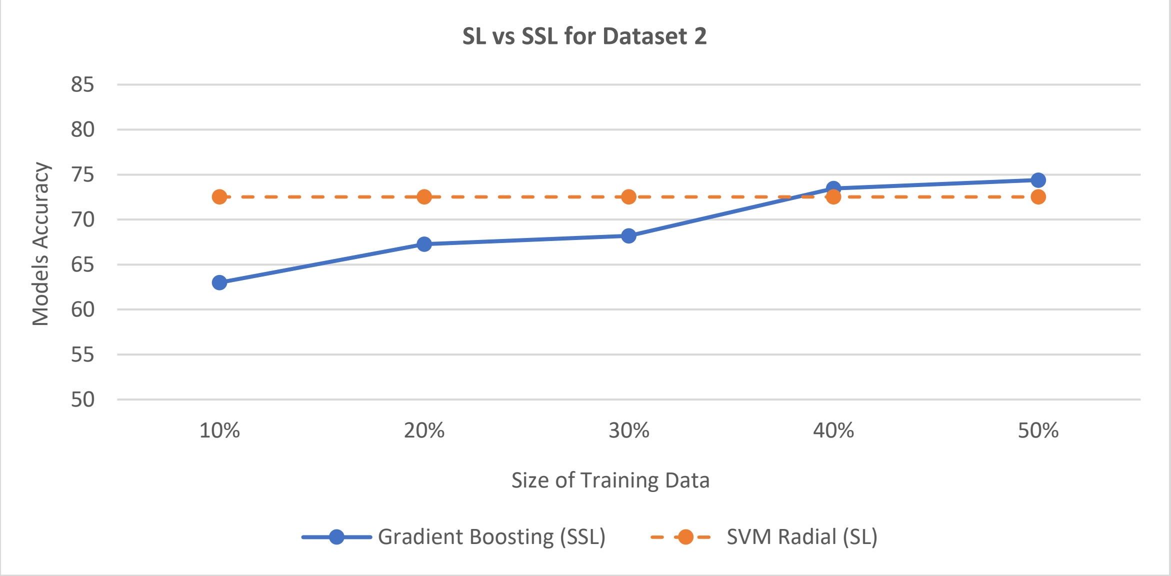

Figure 3. SL vs SSL for Dataset 2.

In terms of computational cost, the Co-Training takes almost twice as long as the SL to train the models. This may be due to the fact that in Co-Training there are 2 models that must be trained several times (until the unlabeled data is exhausted).

VI. CONCLUSIONS

Yes, it is possible to create statistical models of learning to perform objectivity / subjectivity analysis on sporting articles. The results will depend on several factors, such as the quality in the initial labeling of the articles, the algorithms and the type of learning used, and the proportion of the labels to be predicted. However, both in Experiment 1 and in Experiment 2 that we did, it was possible to obtain accuracies greater than 75%, which significantly exceed the baseline.

It is very useful to use the TF-IDF technique to create a matrix with the most important terms of each article, which can be used as input for different ML algorithms. In the particular case of the subjectivity analysis, it helps a lot to improve the accuracy of the models to add to the dataset columns with the number of punctuation marks and the number of stopwords found per article.

The semi-supervised Co-Training technique is used to train algorithms where only a few labeled items are available, but you want to classify a large number of them. In experiment 2 of this work, we obtained with this approach better accuracy of the objectivity / subjectivity of the articles, than a more traditional one such as supervised learning.

Many analyzes focus on phrases or words in a text to infer whether an article is objective; in our case, the extraction of the most relevant words is made from the complete document, although in principle it seems less efficient by only using unigrams, once the top k is armed, the features become similar to an n-gram (n = 10), which means that when measuring document by document with the classifiers, it is understood that the most relevant words that were used were subjective or objective. In a way, this is how the human being measures an article going directly to disqualify a news without having to read it all for the simple fact of detecting with part of it that it is not relevant or that it full of opinions. On the other hand it could be said that framing the article with relevant edges sets the context without having to fill the whole body when measuring objectivity.

VII. FUTURE WORKS

After completing all experiments, we saw many possibilities from the results, even though other developments have had crowdfunding, more personnel, academic support and more machine power and much more experience in the area, our project got very good results. For future works we propose to have (1) polished the algorithms, have more (2) integration between the classifiers and the information retrieval programs, work on (3) pipes or have a distributed database for scalability, optimize the algorithms to work in (4) real time and (5) add more intelligence to the programs that execute the classifiers to let it pick bases on the accuracy the best classifier (e.g pick Ada Boost instead of Random Forest because the accuracy is better on the test examples or speed is better and so on).

Reading related works, we notice that all of this has been done extensively and is very interesting how it is applied to Twitter, this means that this project can be escalated and obtain better results executing all previous ideas.

VIII. ACKNOWLEDGMENTS

We would like to acknowledge the help of Ph.D. Cornelia Caragea from Kansas State University because of all the time and effort she invested on teaching the topics related to machine learning on the summer school 2018 at Universidad de Los Andes.

We would like to acknowledge the help of Ph.D. Oldemar Rodriguez from the University of Costa Rica for all his knowledge shared with us in the fields of Machine Learning and Data Mining.

IX. REFERENCES

[1]. © The R Foundation. (s.f.). R. Cited July 10, 2017, de The R Project for Statistical Computing: https://www.r-project.org/

[2] Nadine Hajj, Y. R. (n.d.). *researchgate*. Retrieved from A Subjectivity Classification Framework for Sports Articles using Improved Cortical Algorithms: https://www.researchgate.net/publication/325248671_A_Subjectivity_Classification_Framework_for_Sports_Articles_using_Improved_Cortical_Algorithms

[3] UCI. (n.d.). *Sports articles for objectivity analysis Data Set*. Retrieved from Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Sports+articles+for+objectivity+analysis#

[4]. Hastie, T., Tibshirani, R., & Friedman, J. (2008). The Elements of Statistical Learning Data Mining, Inference, and Prediction. Stanford: Springer.

[5]. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). An Introduction to Statistical Learning with Applications in R. New York: Springer.

[6]. Fundación Wikimedia, Inc. (July 12, 2018). Wikipedia®. Cited July 12, 2018, Wikipedia - The free encyclopedia: https://en.wikipedia.org/wiki/Gradient_boosting