Real-Time document level subjectivity classification using machine learning techniques

Tod Kahler

Computer Science and Software Engineering,
University of Washington, Bothell, Washington,
kahlewil@uw.edu

Abstract. Often when someone wants to learn more about a topic they turn to the internet for instant access to the store of the world's information. Some of this information is factual and unbiased, but much of it is heavily opinion oriented. This leads to an overwhelming spread of misinformation as users read opinions as facts. If users could be informed of the subjectivity of a document before they even read an article, the spread of misinformation could potentially be mitigated. In this work, a document's syntactic and semantic features were initially reduced using SVM-RFE for real-time purposes, which was followed by the comparison of three machine learning classifiers: Naive Bayes, Support Vector Machine and Random Forest, in an attempt to classify an article as subjective or objective. The models were trained on a set of 1000 sports articles each labeled subjective or objective. With a reduction of 73% in features a recall of 89% and precision of 90.9% was achieved on a fourfold cross-validation with a support vector machine classifier.

1 Introduction

Many would agree that we are currently in the age of information. With the rise of the internet came instant access to the world's store of knowledge. However, anyone can add to this store of information will little restriction. This leads us to not only being in the age of information, but the age of misinformation.

Thousands of articles, blog posts, social media posts, etc. are let loose on the internet every day, some of which are factual and supported by evidence also known as objective, while others are riddled with emotion, belief and opinion also known as subjective. Neither objective nor subjective writings are necessarily negative on their own. Trouble occurs when one is reading a subjective piece that is believed to be objective. When this occurs, opinions are read as facts and there follows misinformation.

A solution to mitigate this spread of misinformation is to inform the reader of the type of article they are reading (objective or subjective). To do this one would need a real-time document level subjective classifier that could possibly be run through a browser extension. In this paper, various machine learning techniques are applied to the document-level subjective classification problem in hopes of creating a "subjective analyzer".

2 Related Works

Sentiment classification in general is a popular area of research, but subjective classification is a much more niche point of interest. Few studies have been done

on subjective classification and even then, most are concerned with sentence level classification rather than document level.

This work is a direct successor to the subjectivity classification done by Hajj and Rizk's who created the corpus and generated the features used in this work for the purpose of obtaining objective articles to make better informed sports. However, they focused on comparing three different feature extraction methods rather than classifiers, achieving a Recall of 85.7 with a reduction of 40% in features using a cortical algorithm for feature reduction and a modified cortical algorithm as a classifier [1].

3 Approach

3.1 Subjective Analyzer

Figure one is the workflow for a subjective analyzer. First, the text of a document would need to be extracted from a web page; the text would then be preprocessed to isolate the desired content. From here the syntactic and semantic features would be extracted and finally the features extracted would be run through a statistical model to label the article subjective or objective. The goal of this work is not to create the complete framework just described. The focus here will be to identify the significant features that will be extracted and to create the classifying model.

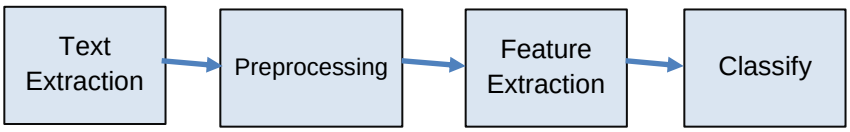

Figure 1 Workflow of real-time subjective analyzer

3.2 Corpus Exploration

The corpus used is composed of a set of 1000 sports articles each labeled either subjective or objective. Labeling was done manually through Amazon's Mechanical Turk. The article authors are composed of both professional journalists and amateur writers from sports blogs. The average length of all of the articles is 697 words with a range of [47, 4283].

3.3 Features Description

The 56 features used were not generated within this study but reused from another work. The detailed feature generation process can be found in [1].

The set of features can be divided into the two following categories.

Syntactic Features All but four features live within this category. Syntactic features are the frequencies of traditional parts of speech, such as pronouns, verb tenses, question marks and quotations

.

Semantic Features Four features were used to extract semantic information from the articles. Using SENTIWORDNET [citation] each word in an article is classified as subjective or objective. The total counts are divided by the word count of the article to normalize the counts between documents. The first and last sentences were also classified as objective or subjective as these sentences are often used by the article author themselves to introduce or remark on the type of article. The last semantic feature was the text complexity or the difficulty of the text.

3.4 Feature Elimination

There is a balance that needs to be met when doing feature elimination. Remove too many features and the accuracy of your model declines. Include too many irrelevant features and your model begins to overfit to features that have no predictability power. When taking into account our desire for a real-time subjective analyzer, feature elimination becomes of great significance. In order to have a real-time analyzer, not only do you need to classify in real-time, but extract features in real-time. Having to extract all 56 features from an article in real-time would most likely make any real-time classifier application unusable.

In order to eliminate features, a SVM based recursive feature elimination process (SVM-RFE) was implemented to initially rank the features. SVM-RFE uses a wrapper, where the complete set of features is initially trained on a SVM. SVM obtains a weight for each feature. The lowest weighted feature is removed and the process is repeated on the pruned set of features. This recursive procedure is continued until you are left with a single feature, where features removed first are of low rank and features removed last are of high rank.

3.5 Classifiers

Three classifiers were used for identifying and classifying objective articles. Each classifier was programmed in python using the scikit machine learning library [2]. The three classifiers:
MNB Naive Bayes with a multinomial distribution
SVM Support Vector Machine using linear kernel.
RandomForest Random Decision Tree using Gini index for splitting criterion.

4 Experimental Results

4.1 Experimental Setup

Preprocessing consisted of the elimination of two attributes in which a frequency of 0 was recorded for all articles. The features were then scaled to a range of [0, 1] to prevent those features with large frequencies from dominating those with few. Finally, due to the inherent imbalance of the data, where 658 were labeled as objective and 342 labeled as subjective, a random oversampling was performed to balance the weight between the two classes. The features were then ranked using a SVM based recursive feature elimination method (SVM-RFE). The performance of three classifiers (MNB, SVM, RandomForest) were

evaluated using fourfold cross-validation, where recall and f-score were recorded. This evaluation was done on each classifier using every set of k best ranked attributes, i.e. performance for each classifier was recorded using the best ranked feature, then the two highest ranked features, then the three highest ranked and so on.

4.2 Feature Reduction and Classification Results

Overall, all three algorithms outperformed the 50% baseline of random-choice. In Figure 1 you can see the change in f-score for each classifier as the number of features increases. By 30 features we can see that every classifier had hit its max f-score at some point. With figure 1 it is evident that not all of the features had predictability power and many were just noise, which is why we actually see a significant drop after 30 features in both MNB and the Random Forest classifiers.

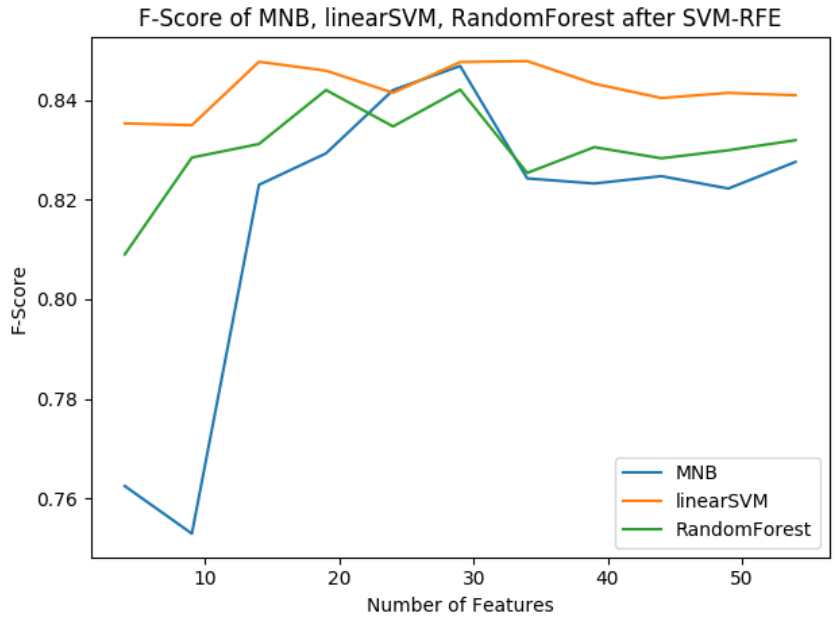

Figure 1. F-Score of MNB, SVM, and RandomForest as number of features increases.

However, looking at the f-scores alone will not give us a complete evaluation of classifiers. In Table 1 we take a closer look at the classification results.

Tin Table 1 the precision and recall is recorded for each of the three classifiers at every step of 5 features. An intuitive definition of recall and precision in the context of this work is as follows; Recall: Out of all of the objective documents in the data set how many were found; Precision: Out of all of the documents that were predicted to be objective how many were actually objective and not misclassified subjective articles. In terms of the subjective analyzer precision is most likely the most important measure. With a low precision misinformation would occur, as one would be reading a subjective article while being told it is objective, thus would likely end up reading opinions

as if they were facts. A max precision of 82.5% occurs with MNB at 35 features. The max recall of 89% occurs with SVM at 15 features and a corresponding precision of 80.9%. Due to the importance of using the minimal amount of features, the SVM at 15 features is most likely the best candidate for our subjective analyzer as it achieves the max recall at 89% and a precision of 80.9% which is only a difference of 1.6% compared to the max precision achieved.

Table 1. Classification Results (Precision and Recall (%))

# of Features	MNB		SVM		RandomForest	
	Prec	Rec	Prec	Rec	Prec	Rec
5	71.3	81.9	79.6	87.9	74.9	84.1
10	80.1	71.0	80.1	87.2	77.9	86.6
15	81.2	83.5	80.9	89.0	78.1	87.7
20	82.1	83.8	81.3	88.2	79.2	86.9
25	82.1	86.5	80.5	88.2	79.7	88.0
30	82.4	87.1	81.3	88.5	77.2	87.6
35	82.5	82.4	81.6	88.2	79.7	85.4
40	82.1	82.5	81.2	87.7	78.6	87.6
45	82.0	83.0	81.2	87.1	78.8	88.5
50	81.8	82.7	81.5	86.9	78.2	87.1
55	80.6	85	81.7	86.6	79.3	87.1
Min	71.3	71	79.6	86.6	74.9	84.1
Max	82.5	87.1	81.7	89.0	79.7	88.5
Avg	80.7	82.7	81.0	87.8	78.3	87.0

According the SVM-RFE the 15 most important features were as follows: VBP, questionmarks, PRP$, past, WP$, CC, compsupadjadv, CD, Quotes, txtcomplexity, baseform, PDT, VB, WDT, exclamationmarks, pronouns1st. Reducing the feature set to 15 features is a reduction of 73% which should significantly decrease the running time of a subjective analyzer as feature extraction time would be reduced by approximately 73%. Although, each feature would have a unique extraction time so 73% is not a very good approximation. For example, the time to extract the frequency of question marks would be significantly less than the extraction time for PRP$, but this analysis is out of the scope of this work.

5. Conclusion

In this work, the groundwork for a real-time subjective analyzer for classifying subjective and objective articles in real-time was presented. Using the data set and features generated in Hajj and Rizk work [1] a precision of 80.5% and a recall of 89% was achieved at a 73% reduction in features using a SVM-RFE method for feature reduction and a SVM classifier. This is a slight improvement

from Hajj and Rizk's classification framework in which they achieved a recall of 85.7% at a reduction of 70% in features, showing both the effectiveness of SVM-RFE as a feature reduction algorithm as well as SVM as classifier for document level subjectivity classification. For future work one might try to use and expand on the methods used here with a different data set (general news articles) to evaluate the robustness of the method, as only sports articles were used here and the results here may not be as transferable to other topics of articles as we might hope.

6. References

1. Hajj, N., Rizk, Y. & Awad, M. Neural Comput & Applic (2018). https://doi.org/10.1007/s00521-018-3549-3

2. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.