Assessing and Implementing Automated News Classification

Francisco Romero
Department of Electrical Engineering
Stanford University
Stanford, California
faromero@stanford.edu

Zahra Koochak
Department of Electrical Engineering
Stanford University
Stanford, California
zahraa@stanford.edu

Abstract—**Newsfeed websites sort articles by subject to make it easier for readers to search for articles in their preferred category. When uploading a new article, authors are usually tasked with selecting the most pertinent category so the new addition can then be grouped with similar articles. We are interested in further developing the framework to automatize the classification of news articles using machine learning and Natural Language Processing (NLP) techniques. We explore three classification methods: Support Vector Machine (SVM), Naïve Bayes, and Softmax Regression, and evaluate each classifier's ability to select the appropriate category given an article's title and a brief article description. Our results show Softmax Regression to be the best classifier among the three we evaluated.**

Index Terms—**news, articles, NLP, SVM, Naïve Bayes, Softmax, classification, tf-idf**

1. Introduction

When visiting a newsfeed website, we are often interested in reading articles in a specific category. Based on their content, articles are sorted by subject, which allows readers to effortlessly find articles in their preferred category. To determine the article's category, most newsfeed websites ask the author to select the best-fit category for their article. Selecting an article's category is not only based on the author's opinion, but can also be tedious when several articles are simultaneously being added to a newsfeed website. Since the vocabulary and terminology used by an article's author is indicative of the target audience and, more generally, of the article's category, we believe this process can be effectively automated.

For our project, we are interested in assessing three classification methods to determine the feasibility of automatically classifying news articles. We selected to use the article's title and a 1-2 sentence article description as the input to our classifier. We then evaluate the capability of our classifier using a minimal amount of information about the article's subject. Finally, we used three supervised learning classifier to output a predicted article category: Naïve Bayes, Support Vector Machine (SVM), and Softmax Regression. Our data spans over seven categories: Sports, US, Science and Technology, Business, World, Entertainment, and Health. Based on the lexical features of each article, it was the job of each classifier to select the most appropriate category for the article.

2. Relevant Work

Previous work has focused on developing algorithms and software to automate the process of accurate text classification. Young and Jeong implemented a new feature scaling method that uses the Naïve Bayes classifier. The feature scaling method was tested on a news group dataset and outperformed other popular ranking schemes, such as Information Gain while noting Naïve Bayes as being a suitable classifier for news articles [10]. Wang *et al.* developed an optimal text categorization algorithm that is based on the SVM algorithm used in this paper [11]. Using a news article corpus similar to ours, they found their algorithm to outperform other classifiers such as the decision-tree algorithm and the K-nearest neighbor algorithm. Hakim *et al.* evaluated the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm's ability to be used in text classification for news articles in Bahasa Indonesia [13]. However, their approach did not focus on any machine learning techniques, only on the TF-IDF algorithm. Looking to our future work, one of the first frameworks for Neural Networks was developed by Ruiz and Srinivasan [12]. Using about 2,350 documents, they showed the ability of Neural Networks to accurately categorize text. Do and Ng explored text classification using a modified Softmax Regression algorithm [17].

Table 1: Lexical Feature Extraction

Top 10 Words for Each Category						
Sports	US	Science & Technology	Business	World	Entertainment	Health
NFL	NEW	NEW	US	PRESIDENT	NEW	STUDY
FIRST	US	APPLE	BILLION	US	THEATER	NEW
GAME	STATE	GOOGLE	NEW	KILLED	SHOW	HEALTH
WIN	TEXAS	SPACE	PRICES	FORCES	STAR	CANCER
OVER	POLICE	FACEBOOK	OIL	NUCLEAR	IDOL	US
NEW	STATES	US	BANK	GOVERNMENT	WEDDING	MAY
PLAYERS	TUESDAY	ONLINE	SALES	POLICE	AMERICAN	DRUG
SEASON	OVER	INTERNET	MAY	PEOPLE	UP	RISK
COACH	WEDNESDAY	SONY	STOCKS	BIN	FIRST	HEART
METS	COURT	IPAD	UP	LIBYA	FILM	PEOPLE

The classifier outperformed one-against-all SVM and multi-class SVM.

Our work deviates from the aforementioned studies in that we used only the title and a short description of each article for our lexical feature extraction and we focused on evaluating all three classifiers rather than trying to optimize the performance of a particular one. In addition, we are using a variant of the Softmax Regression algorithm presented by Do and Ng to perform our text classification.

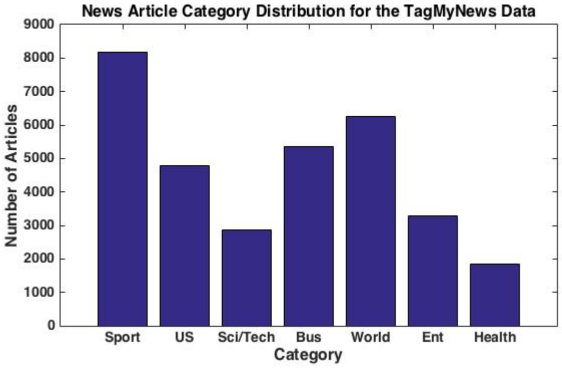

Figure 1: Distribution of the TagMyNews Data

3. Dataset and Pre-Processing

To perform our classification evaluation, we used the TagMyNews Dataset [1]. The corpus includes 32,602 training examples of news articles. Each training example has a structure including a title, a description, a news article link, an ID, the date of publication, the news article source, and a subject category. Of interest to us were the article's title, the brief article description, and the pre-labeled category. Figure 1 shows the distribution of news articles for each category. The majority of the training examples were from the *Sports* category, while we had the fewest training examples from the *Health* category. However, as we show in Section 5, the number of training examples for each category did not necessarily reflect the classifier's ability to determine the category of an article from the testing set.

Pre-processing the news article data involved three steps. First, we separated each article's title, description, and pre-labeled category into a separate text file, since the corpus is formatted into a single file. Second, we removed all punctuation from the title and description. Third, we capitalized all letters in the title and description. The latter two steps are necessary for performing a lexical feature extraction using the vocabulary of the title and the article description.

4. Methodology

To test the three classifiers, we divided our data into training news articles and testing news articles. 70% of our data (22,819 articles) were designated as the training articles and the remaining 30% (9783 articles) were designated as the testing articles.

Since an author writes an article with an intended category or subject in mind, we believe the vocabulary can be used as our features for our classifiers. Thus, our objective for our feature extraction was to obtain the *f*-most salient words for each category, and count how many times each word appeared in a given article. We tested feature sizes of *f* = 40, 50, 70, 150 and 200 to obtain the best accuracy possible for each classifier. To obtain the *f*-most salient words for each category, we used the TF-IDF algorithm, which we explain in the next section. The extracted features were then passed to each of the three classifiers.

4.1 Extracting Salient Words with TF-IDF

The TF-IDF weighting scheme will assign each term, *t*, a given weight in a document as follows:

$$w_{t,d} = tf_{t,d}\ x\ \log(N/df_t) \qquad (1)$$

where *N* is the number of documents. The weight is assigned by the product of $tf_{t,d}$, the term frequency, and $\log(N/df_t)$, the inverse document frequency. For each category, we computed the TF-IDF of each term and obtained the *f*-most salient words from the sorted list of TF-IDF rankings. While the goal of using TF-IDF was to extract the most meaningful and indicative words for each category, we needed to further filter the results of the algorithm to exclude words such as 'the', 'or', 'him', which carry no significance. Thus, we implemented a "stop-word" list to remove these meaningless words from our feature set based on [18]. Table 1 shows a list of the top 10 words extracted for each category using TF-IDF.

We also investigated a variation of the TF-IDF algorithm called Sublinear TF Scaling [6]:

$$w_{t,d} = wf_{t,d}\ x\ \log(N/df_t) \tag{2}$$

where $wf_{t,d}$ is given by:

$$wf_{t,d} = \begin{cases} 1 + \log(tf_{t,d}), & if\ tf_{t,d} > 0 \\ 0, & otherwise \end{cases} \tag{3}$$

Each term's frequency is now assigned a weight, which may represent a term's significance more accurately than just counting the number of occurrences. However, we found the classic TF-IDF to have better performance on all three models over this modified version of Equation 1. Using Equation 2, the top *f*-words for each category were similar to those of Table 1, but not exactly the same. Since the classic TF-IDF performed better for all three classifiers, we did not use Equation 2 in our final implementation.

4.2 Implementing the Classifiers

We selected to evaluate Naïve Bayes, SVM, and Softmax Regression due to their ability to perform supervised learning on multi-class datasets. In the following sections, we describe each algorithm and how it pertains to our goal of news classification.

4.2.1 Support Vector Machine

The SVM algorithm requires the solution to the following optimization problem [4]:

$$\min_{\gamma,w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \tag{4}$$

$$s.t.\quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

where C is a regularization weighting parameter. The goal of the SVM is to find a linear-separating hyperplane that has the maximal margin in the higher dimensional space that x_i is mapped to [4]. For our project, we used the LIBSVM MATLAB library to implement our Support Vector Classifier (SVC) [2]. The LIBSVM library offers different options that allow a user to set the SVM-type and the kernel-type, as well as values for the different parameters. For our SVM, we tested multiple kernel functions, including a linear, polynomial, and radial basis function (RBF). We found the RBF to have the best performance, and subsequently used it for our implementation of the C-SVC. The RBF kernel is given by $e^{-\gamma|x-x_i|}$, where γ is a weighting parameter and x is a query point [3].

In order to maximize the performance of the SVM, we needed to optimize the parameters γ and C. To do so, we performed a parameter grid search by implementing internal cross validation [3]. We selected exponentially growing ranges for both C and γ ($C = 2^{-5}, 2^{-3}, \dots, 2^{15}$ and $\gamma = 2^{-15}, 2^{-13}, \dots, 2^{5}$). Using 5-fold internal cross validation on *only* the training data, we iterated through the different γ and C options, noting the values that gave the highest accuracy during each fold. We did not include any of the test data in the internal cross validation, since optimizing for a given test set would be incorrect.

LIBSVM implements a one-against-all training model for the SVM when using a multi-class dataset. For each label value, $y = \{1,2,\dots,k\}$, a different SVM model is trained. Thus, we make k-different binary models. We then test each model on the testing data and determine the model from which the highest prediction confidence is returned in order to classify the data.

4.2.2 Naïve Bayes

Naïve Bayes was used as our baseline text classifier because it could be quickly implemented for analysis. Since our news classification framework has been defined for multiple classes, we have developed the appropriate algorithms for this case as:

$$\phi_{j|y=class_j} = \frac{\sum_{i=1}^{m} 1\{x_j^i = 1 \cap y^i = 1\}}{\sum_{i=1}^{m} 1\{y^i = 1\}} \tag{5}$$

$$\phi_{y=n} = \frac{\sum_{i=1}^{m} 1\{y^i = 1\}}{m} \tag{6}$$

From [16], these parameters have a natural interpretation. For example, $\phi_{j|y} = i$ is the fraction of the category i in which word j appears. Having fit all the parameters, we calculate:

$$p(y = j|x) = \frac{p(x|y = j)p(y = i)}{p(x)} \tag{7}$$

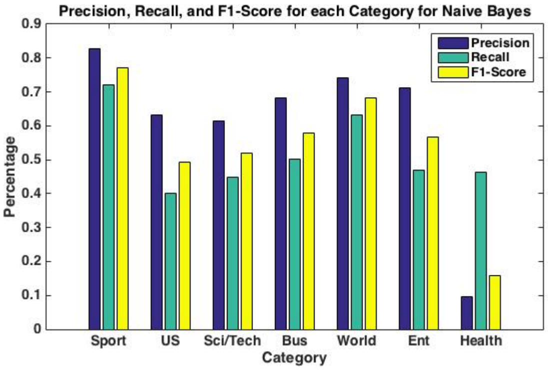

Figure 2: Naïve Bayes Performance Measurements

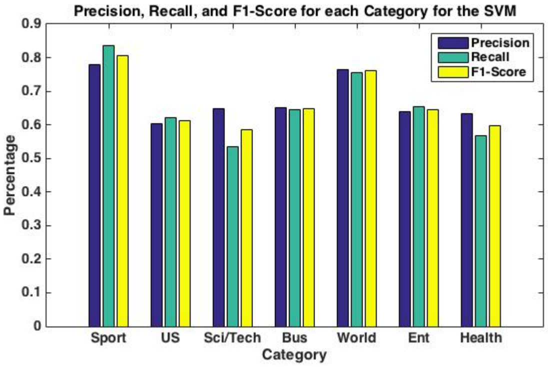

Figure 3: SVM Performance Measurements

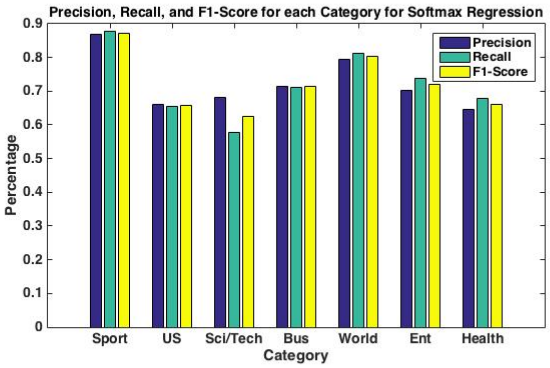

Figure 4: Softmax Performance Measurements

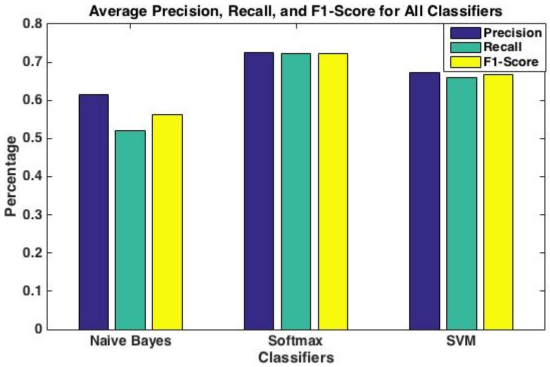

Figure 5: Average Performance Measurements for each Classifier

Since we are applying the classifier over a large vocabulary, we implemented Laplace Smoothing to avoid having $\phi_j's$ end up as zeros.

4.2.3 Softmax Regression

We selected to use Softmax Regression, also known as the Multinomial Logistic Regression, as opposed to 7-binary classifiers because our seven classes are mutually exclusive (i.e. a news article will be a part of at most one category). For this classifier, the class probabilities, $p(y|x)$ are modeled as:

$$p(y|x) = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} = \frac{e^{\theta_i^T x_i}}{\sum_{j=1}^{k} e^{\theta_j^T x_i}} \tag{8}$$

where the θ parameters are learned from the training set by maximizing the conditional log likelihood of the data [16]. In this approach, a total of k-parameters are trained jointly using numerical optimization.

5. Results and Discussion

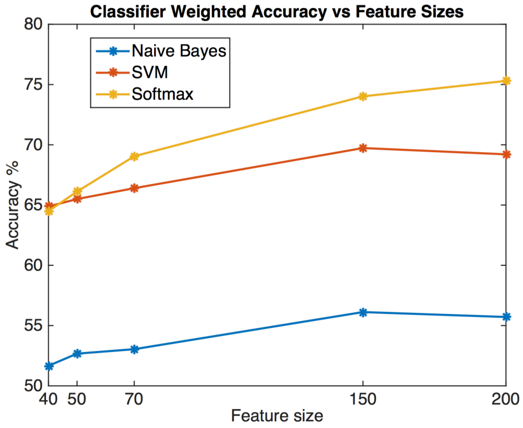

Figure 6: Classifier Accuracy as Feature Size Varies

The weighted accuracy of each classifier is presented in Figure 6. Softmax Regression achieved the best performance out of the three classifiers, with a maximum accuracy of 75.36% at 200 features. Naïve Bayes performed significantly worse than the other two, achieving a maximum accuracy of 56.12% for 150 features, which we attribute to its simplicity and weak scalability. SVM's best performance was not much worse than Softmax Regression, with a maximum accuracy of 69.21% for 150 features. Both

Naïve Bayes and SVM ran into overfitting issues at 200 features (i.e. performance began to decrease). We attribute the performance drop to the minimal overlap in words between the different categories, especially as the feature size increased. However, the optimized SVM performed much better than the non-optimized SVM, which ran into overfitting issues after 70 features and had a maximum accuracy of 61.37%. Since Softmax Regression did not run into overfitting issues for our evaluation's maximum feature size, future work will continue to increase the feature size and further evaluate the performance of the classifier.

To further evaluate each classifier's performance on individual classes, we used the three measurements derived from the confusion matrix: Precision, Recall, and F_1-score. The latter three measurements reflect the importance of retrieval of positive examples in our text classification [14]. Precision is the class agreement of the data labels with the positive labels given by the classifier, while Recall is the effectiveness of a classifier to identify positive labels [14]. The F_1-score is the harmonic mean of precision and recall and is given by:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

Figures 2-4 show each classifier's Precision, Recall, and F_1-Score for each category. The *Sports* category had the highest Precision and Recall, and subsequently the highest F_1-score for all three classifiers. This is due to the esoteric and overly specific terms used in the *Sports* categories lexical features, as seen in Table 1. For Softmax Regression and SVM, the *Science and Technology* category had the lowest F_1-score, while for Naïve Bayes, *Health* had the lowest F_1-score. While the plots of these three metrics for SVM and Softmax Regression strongly resembled each other, Naïve Bayes had Precision higher than Recall for all categories except for *Health*. The balance between Precision and Recall for SVM and Softmax Regression is visible in Figure 5. The average Precision and Recall are almost identical, demonstrating the two classifiers ability to perform better on a per-category-basis, especially for Softmax Regression.

The low F_1-scores of both the *Health* and *Science and Technology* categories are due to the inability of the classifier to distinguish one category from another. In particular, for *Science and Technology*, the majority of the false-negatives came from the *Business* category. This is most likely attributed to technology often being mentioned in a commercial setting. Even more interesting, the number of false-positives for the *Health* category according to Naïve Bayes is very high (evident by the very low Precision in Figure 2). Using Naïve Bayes, the majority of the false-negatives for every category was *Health*, which is the source of the low weighted accuracy.

6. Future Work

Although the Sublinear TF Scaling modification to the TF-IDF algorithm did not outperform the classic TF-IDF, we would like to look into more TF-IDF variants and other methods to improve the feature selection process. Other methods might include Information Gain or Conditional Mutual Information [5]. To explore the optimal feature size, we might want to try a Forward or Backward search procedure with a reduced dataset. We are also interested in testing other classification methods, such as Recurrent Neural Networks, to compare them against the three implemented in this work. In particular, classifying articles into more specific categories, such as *Computing* instead of just *Science and Technology*, may lead to classifier performance differences compared to the results from this work.

Beyond trying to classify news articles into a specific category, we would like to explore the application of our developed framework towards detecting emotions or bias in a news article. Work has been done to explore using SVMs and semi-supervised learning models for political bias by [7]. In addition, [8,9] investigated emotional classification from both the writer's and reader's perspective using SVMs. Effectively, we want to see how well our framework generalizes to other underlying aspects of news articles.

7. Conclusion

Based on the F_1-Scores in Figure 5, our best classifier, Softmax Regression, performed 12.57% worse than the proposed methods of Wang *et al.* for a similar dataset. Nevertheless, we have shown the ability of three different classifiers to automatically classify news articles into their subject category. Naïve Bayes' performance was adversely affected by the high number of false-negatives from the *Health* category. The well-known SVM algorithm, although not the top performer, was also found to be highly suitable for classifying news article into their subject category.

8. REFERENCES

[1] A^3 Lab. TagMyNews Dataset.
http://acube.di.unipi.it/tmn-dataset/

[2] C. Chang and C. Lin, “LIBSVM: a library for support vector machines,” in *ACM Trans. on Intelligent Syst. And Tech.*, vol. 2, no. 3, pp. 1-27, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[3] C. Hsu *et al.*, “A Practical Guide to Support Vector Classification,” 2003

[4] A. Ng. CS 229. Class Lecture, Topic: “Support Vector Machines.” Stanford University, Stanford, CA, October 14, 2015.

[5] A. Khan *et al.*, “A Review of Machine Learning Algorithms for Text-Document Classification,” in *Journal of Advances in Inform. Tech.*, vol. 1, no. 1, pp. 4-20, 2010.

[6] C. Manning *et al.*, “Scoring, Term Weighting and Vector Space Model,” in *Introduction to Information Retrieval*, Cambridge University Press, 2008, ch. 6, sec. 6.4, pp. 126-127.

[7] D. Zhou *et al.*, “Classifying the Political Leaning of News Articles and Users from User Votes,” in *Proc. of the 5th International AAAI Conf. on Weblogs and Social Media*, 2011.

[8] K. Lin *et al.*, “Emotion Classification of Online News Articles from Reader’s Perspective,” in *Proc. of the 2008 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Tech.*, 2008.

[9] D. Bracewell *et al.*, “Determining the Emotion of News Articles,” in *Computational Intelligence*. Springer Berlin Heidelberg, pp. 918-923, 2006.

[10] E. Young and M. Jeong, “Class Dependent Feature Scaling Method using Naïve Bayes Classifier for Text Datamining,” in *Pattern Recognition Letters*, vol. 30, no. 5, pp. 477-485, 2009.

[11] Z. Wang *et al.*, “An Optimal SVM-based Text Classification Algorithm,” in *2005 International Conf. on Machine Learning and Cybernetics*, pp. 1378-1381, 2006.

[12] M. Ruiz and P. Srinivasan, “Automatic Text Categorization using Neural Networks,” in *Proc. of the 8th ASIS SIG/CR Workshop on Classification Research*, pp. 59-72, 1998.

[13] A. Hakim *et al.*, “Automated Document Classification for News Articles in Bahasa Indonesia based on Term Frequency Inverse Document Frequency (TF-IDF) Approach,” in *6th International Conf. on Information Tech. and Elec. Eng*, pp. 1-4, 2014.

[14] M. Sokolova and G Lapalme, “A systematic analysis of performance measures for classification tasks,” in *Information Processing and Management*, vol. 45, no. 4, pp. 427-437, 2009.

[15] A. Ng. CS 229. Class Lecture, Topic: “Supervised Learning.” Stanford University, Stanford, CA, September 30, 2015.

[16] A. Ng. CS 229. Class Lecture, Topic: “Generative Learning Algorithms.” Stanford University, Stanford, CA, October 5, 2015.

[17] C. Do and A. Ng, “Transfer learning for text classification,” in *NIPS*, 2005.

[18] Ranks. Stopwords list. http://www.ranks.nl/stopwords