Building a New York Times News Classifier

Stuart Miller
Department of Electrical and
Computer Engineering
Missouri University of Science & Technology
Rolla, Missouri 65409
Email: stuart.miller@mst.edu
Web: http://web.mst.edu/~sm67c

Abstract—**Keeping up with the current state of the the world involves staying knowledgeable about current events sourced from many different journalistic outlets. However, news reporting can often be unordered, haphazard, and even hyperbolic. In order to bring some order to this, a neural-network classifier can provide some insight into the categorizations that news outlets attempt to place on their top stories. By using natural language processing (NLP) methods to format news data for neural network learning, such a system has the ability to examine just the core content of the news publications and extract a true categorization. Through this method, trends can be revealed and conclusions can be drawn about the current state of the field. A news consumer can even gain a more unbiased understanding of the field as a whole.**

I. Introduction

Ecker, et. al. propose that a large portion of current news headlines are downright misleading [1]. While not being abjectly false, the headlines and opening statements can often give a reader a false impression or insert an opinion into the facts being reported thereafter. This is a problem that plagues the current news market; news is often twisted, misinterpreted, or sensationalized. However, at the heart of any news articles, the core information still has to be conveyed. For this reason, it is possible to build a news classifier that looks past and bias presented in headlines, bylines, images, or other accompanying graphics and extract a true classification from the body of the text alone.

In order to achieve this, an investigation into the applicability of text mining and natural language processing (NLP) combined with a neural network classifier is presented. The goal is to provide a tool that can extract true classifications, as well as trends in the current news market.

II. NLP & Network Preprocessing

A. Data Acquisition

Data used to train the network is acquired directly from the New York Times developer API [2]. Queries to the API can be customized in a number of ways, but for this project, data downloads consist of every published article during a one month period. For obvious reasons, video articles, media pieces and photo journals are excluded. All in all, this results in a data set of approximately 4000 articles per month. This also provides an easy train/text/validate split, with successive months taking the place of each division.

In addition to providing the news article contents, the New York Times API also provides a wealth of associated meta data with each article. For classification purposes, each articles comes with several ”subject” tags. These were used to create the expected output classifications of the network.

B. PreProcessing

Articles were filtered according to standard natural language processing techniques. Words were first trimmed of any extraneous symbols and punctuation, then normalized to lowercase. Subsequently, each was fed through a Porter stemmer. Formalized by Martin Porter in 1980 [3], the Porter stemming algorithm normalizes prefixes and suffixes to common English words. (For example: ”estimated”, ”estimator”, and ”estimation” would all be normalized to ”estimat”.) Common stopwords such as ”a, ”the”, ”that”, etc. were also removed. Through such a procedure, each article was reduced to a list of stemmed terms. Each documents list of words was aggregated by term and put into a data structure containing all terms and their frequency counts, commonly known as the ”bag of words” model.

C. Term Frequency-Inverse Document Frequency

All that remains to prepare the data for input to a neural network is quantify that bag of words, a process done here using the term frequency-inverse document frequency (TF-IDF) algorithm. Proposed in 1972 as part of work done at the Cambridge Computing Laboratory, the TF-IDF algorithm [4] provides a quick and easy way to rank a document's relevance to a particular term. Initially proposed by Jones, the TF-IDF is a valuable concept for ranking within bodies of literature. In short, the TF-IDF ranking for a particular term in a particular document is comprised of the product of the term frequency and the inverse document frequency (Equation 1).

Term frequency is defined as the number of times term t appears in document d over the total number of terms in the document (Equation 2). Inverse document frequency is defined as the log of the inverse of the total number of documents containing term t (Equation 3). Various weightings and smoothings can further be applied to fine-tune a result for

Word Stem	Quantity	Word Stem	TF-IDF Score
new	17294	perform	2.356
year	11324	come	1.500
like	9467	lot	1.318
state	8932	everi	1.274
time	8389	commun	1.249
peopl	7575	though	1.198
trump	7556	sai	1.186
york	6908	talk	1.134
work	6418	sexual	1.067
presid	6031	want	1.050
last	5823	offic	1.045
make	5469	right	1.043
two	5444	well	1.032
compani	5430	question	1.020
first	5359	million	0.957
just	5107	peopl	0.950
includ	5102	end	0.946
unit	5056	big	0.942
get	4929	deal	0.935
mani	4875	court	0.931

TABLE I: Term Frequency vs. TF-IDF Score, January 2018

a a particular data set if so desired. Weightings can be binary, log-based, or scaled by any custom factor.

$$tfidf = tf(t,d) \cdot idf(t,d) \tag{1}$$

$$tf = \frac{t \in d}{\forall t \in d} \tag{2}$$

$$idf = \log\left(\frac{D}{d \in D : t \in d}\right) \tag{3}$$

Although the inner working of Internet search engines are a closely guarded corporate secret, it is theorized that the basis of each is a variation of a TF-IDF ranked search, a notion investigated and exploited by Lin and Chi in [5].

The true benefit of the TF-IDF algorithm is not only that it provides an easy way to numerically rank textual terms; but that it can bring categorically important terms to the top of the ranking, while decreasing the ranking of common words that are less useful in classification. For example, when examining news articles, the term ”year” is consistently one of the most frequently appearing terms; a logical notion as news articles must frequently cite dates and times of events. Year does not provide a good input value to a neural network as it would not help distinguish one unique subject from another. Terms that are common to certain types of news articles (say ”democrat”, ”sexual”, or ”touchdown”) would receive high TF-IDF scores as they uniquely indicate certain subjects (”politics”, ”sexual misconduct”, and ”football”, respectively).

Data taken from the January, 2018 set of articles is shown in Table I. Here, the quantity is summed over all articles and the TF-IDF scores are averaged over all articles.

III. Network Architecture

A. Inputs

As described above, the network takes term TF-IDF values for each article as inputs. The total number of unique terms for a month's news article totalled around 80,000. Most of these were unique references to names and places, or words that only appeared in a single article. To reduce the size of the input set, the 200 top-scoring words were extracted and used as the input data.

B. Outputs & PostProcessing

In a similar manner, the data set consisted of far too many subject classifications for the network to comprehend. The top 20 most common subjects were chosen as outputs and the rest were discarded. Most of the discarded subjects did not appear frequently enough to be able to sufficiently train the network (less than 50/4000 total articles).

The decision was made to train an individual binary classifier network for each subject. While a single 20-output network would have sufficed, with the limited computing resources available, binary classifier networks trained much faster.

The infrequency of the output class does present a problem though. Even the most prevalent output class only averaged around 500 articles per month out of the total 4000, with most having between 80 and 200 articles in their class. Table II. shows the subject distribution for January 2018. To account for this, data undersampling was utilized. A 50/50 split was used, taking a random equivalently sized sample from the overrepresented class and mixing it in with the underrepresented class.

Finally, a postprocessing step was applied to change the continuous $[0-1]$ value to a boolean classification. The most consistent results were obtained when selecting the median of all outputs as the threshold value. There were consistently a few outlier, but the majority of the data could be properly classifier in such a manner.

Subject	Articles
UnitedStatesPoliticsAndGovernment	525
PoliticsAndGovernment	261
#MeTooMovement	180
SexualHarassment	171
Movies	166
BooksAndLiterature	139
ImmigrationAndEmigration	131
Television	127
WomenAndGirls	125
RealEstateAndHousing(Residential)	122
TravelAndVacations	116
FashionAndApparel	107
UnitedStatesInternationalRelations	106
Art	90
SocialMedia	87
SexCrimes	85
CookingAndCookbooks	83
Theater	82
InternationalTradeAndWorldMarket	81
ComputersAndTheInternet	80

TABLE II: Subject Count, January 2018

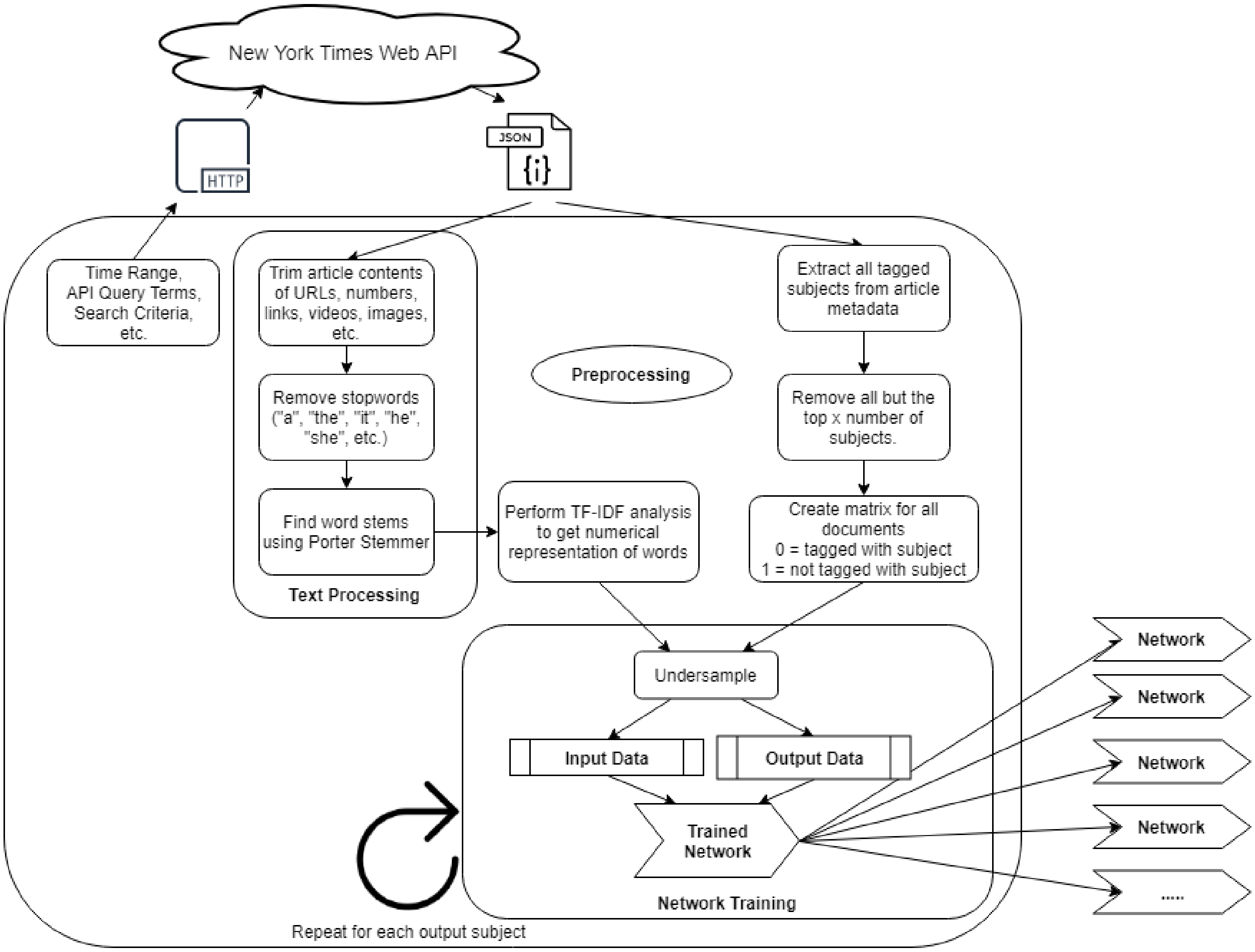
New York Times Web API
HTTP
JSON
{i}
Time Range, API Query Terms, Search Criteria, etc.
Trim article contents of URLs, numbers, links, videos, images, etc.
Remove stopwords ("a", "the", "it", "he", "she", etc.)
Find word stems using Porter Stemmer
Text Processing
Preprocessing
Perform TF-IDF analysis to get numerical representation of words
Extract all tagged subjects from article metadata
Remove all but the top x number of subjects.
Create matrix for all documents
0 = tagged with subject
1 = not tagged with subject
Undersample
Input Data
Output Data
Trained Network
Network Training
Repeat for each output subject
Network
Network
Network
Network
.....

Fig. 1: Network architecture

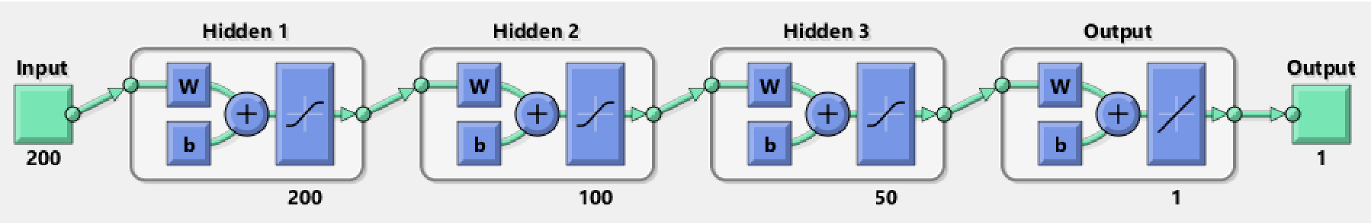

Fig. 2: Network Architecture

C. System Architecture

Overall, the system as described is shown in Figure 1.

D. Network Configuration & Training

In order to train the network, scaled conjugate gradient backpropagation[6] was used. Even after being reduced to a binary classifier, Levenberg-Marquardt backpropagation[7] proved too computationally expensive. Additionally, network weight initialization used the Nguyen-Widrow method[8] as there was no reason to choose any specialized method here.

In order to mimic a convolutional neural network, a ”funnel” structure was used for the hidden layers. Starting with 200 to match the inputs, each of four hidden layers grew gradually smaller to reduce to a single neuron in the output layer. The layer structure is shown in Figure 2.

Throughout the design process, several network architectures were tried. The best results were found after switching to a single binary classifier instead of a multi-class, multi-output network. The binary classifiers tran much faster produce better accuracy. However, the single most influential part of optimizing the network was modifying the preprocessing steps. Modifying the training parameters had very little effect, compared to refining the TF-IDF, sampling, or input size parameters.

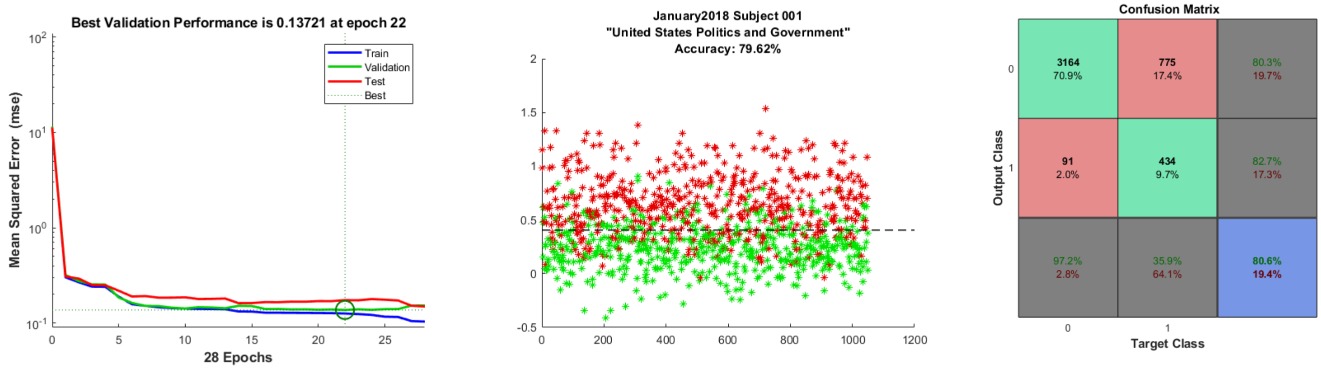

Fig. 3: January 2018, Subject 1
”United States Politics and Government”

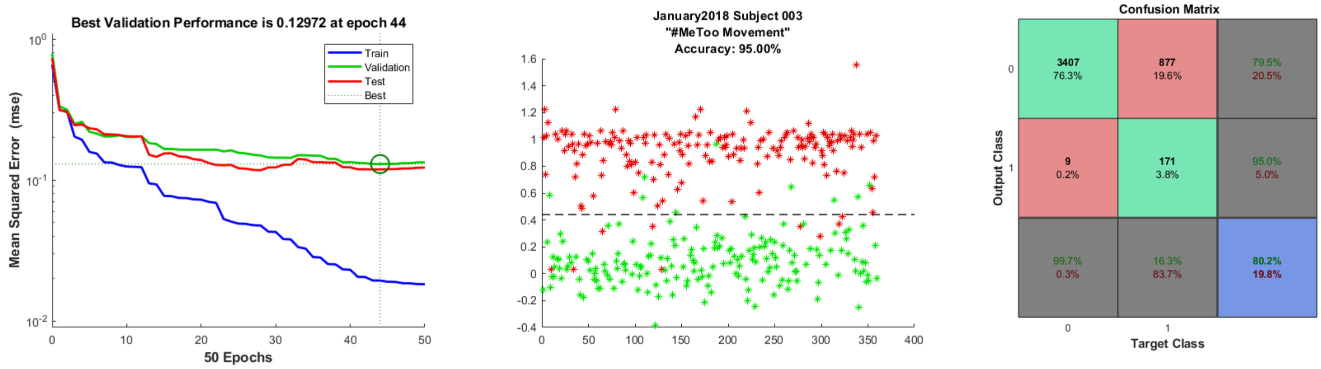

Fig. 4: January 2018, Subject 3
”#MeToo Movement”

IV. Results

A. Training & Classification Accuracy

Figures 3 and 4 show the results of training. The networks generally display pretty good performance. Some have a tendency to overfit, but still produce mostly accurate results. Notice the dotted reference line on the scatter plots denoting the threshold. Here, the colors represent each data point's actual class, while its y axis value represents its calculated class; either above or below the threshold. While there are a few outliers, most of the data points are classified correctly.

This is further demonstrated in the confusion matrices which show overall accuracy for the test set; these two subjects both resulted in about 80% accuracy overall. This makes sense as these two subjects are both very common and well-represented in the dataset.

Overall, none of the test data set's every scored about 75-80% accuracy. Several of the less populated classes would frequently produce as low as 60-70%. Why is this? Looking more closely at the initial training data can reveal some flaws in the setup. The New York Times never gave any one article more than 3 classifications. This means that some broad articles that may have had text that toughed on other subjects were trained not to return their additional classifications. Furthermore, the New York Times authors seemed to favor more niche subject classifications. For example, take the articles entitled ”At de Blasio Inaugural, Speeches by Two Who Might Replace Him” [9]. The articles discussed political issues inherent to those considering a run for the New York City mayoral race. The given subjects were ”Elections, Comptrollers”, “Elections, Public Advocate”, and “Inaugurations”. None of these subjects made the list of the top 20 (or even the top 50) and as such the expected classification for the article was nothing (i.e. classify all subjects as false). However, the network returned a true classification for ”United States Politics and Government”, a classification that clearly fits. In cases like these, the training data for the network suffered.

B. Applicability

As an additional test of the model's capabilities, news articles from several other outlets were tested as well. Classifications were run against the trained networks from the January 2018 dataset. Results are shown in Table III below. Given that these examples features articles that are definitively in particular subjects, the network performed quite well. With regards to less clear articles it behaved in mostly the same regards as less clear New York Times articles. This served as further validation for the model.

Article	Outlet	URL	United States Politics and Government	Sexual Harassment	Movies
Congress returns this week: Here's where the Democratic investigations stand	CNN	https://www.cnn.com/2019/04/28/politics/house-democratic-investigations-tax-returns/index.html	1	0	0
Stephen Moore's Fed chances may rest with GOP women senators	CNN	https://www.cnn.com/2019/04/28/politics/stephen-moore-senate-chances/index.html	1	1	0
'Avengers: Endgame' shatters records with $1.2 billion opening	CNN	https://www.cnn.com/2019/04/28/media/avengers-endgame-box-office-record/index.html	0	0	1
Trump flips on Fox News analyst after he calls president's actions 'criminal'	Yahoo	https://news.yahoo.com/trump-flips-on-fox-news-analyst-after-he-calls-presidents-actions-criminal-143939461.html	1	0	0
Trump bashes news 'fakers' as journos gather for D.C. gala	Politico	https://www.politico.com/story/2019/04/27/trump-wisconsin-rally-1291372	1	0	0

TABLE III: Classified Articles from External (non-NYT) Sources

V. Future Work

There remains considerable refinement still possible for this model. Perhaps the most glaring flaw is the lack of comprehensive input data. Either by manually classifying articles, or by seeking out a more accurate sources of subject-categorized news articles than The New York Times, accuracy could likely be improved.

Another problem is that each binary classifier trained with exactly the same parameters. This tended to produce some subjects that overfit and some that underfit. Some additional research may reveal that more common subjects tend to behave in a particular manner and could benefit from customized network parameters that are different than less common subjects. By varying parameters throughout the dataset in such a manner, it may be possible to further improve the model's accuracy.

VI. Conclusions

Overall, this project served as a successful endeavor into classifying news articles. In particular, the ability of the model to predict accurate classifications that were not in the input data, such as the case of the di Blasio article mentioned earlier, attest to the ability of the model. The trends inherent in the common language used in news reporting help create a cohesive model, using both natural language processing and neural network techniques.

References

[1] U. K. H. Ecker, S. Lewandowsky, E. P. Chang, and R. Pillai, "The effects of subtle misinformation in news headlines," *Journal of Experimental Psychology*, vol. 20, pp. 323–335, 2014.

[2] "The new york times developer network," 2019. [Online]. Available: https://developer.nytimes.com

[3] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[4] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[5] T. F. Lin and Y. P. Chi, "Application of webpage optimization for clustering system on search engine v google study," in *2014 International Symposium on Computer, Consumer and Control*, June 2014, pp. 698–701.

[6] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525 – 533, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608005800565

[7] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. [Online]. Available: https://doi.org/10.1137/0111030

[8] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *1990 IJCNN International Joint Conference on Neural Networks*, June 1990, pp. 21–26 vol.3.

[9] J. D. Goodman, "At de blasio inaugural, speeches by two who might replace him," January 2018. [Online]. Available: https://www.nytimes.com/2018/01/01/nyregion/at-de-blasio-inaugural-speeches-by-two-who-might-replace-him.html