Social Media Content Filter

Nanthini Balasubramanian NANTHINI@SEAS.UPENN.EDU
Garvit Gupta GARVIT@SEAS.UPENN.EDU
Jordan Farrer FARRER@WHARTON.UPENN.EDU

Abstract

The Facebook News Feed is filled with shared news articles and posts. Our project proposes an automatic news category filtering feature using machine learning that enables users to filter to only the types of posts and news articles they want. We examine a variety of text categorization algorithms that can label posts into categories such as business, science, entertainment, and health. We foresee that this methodology could be implemented either as (1) an external browser extension that hides Facebook posts based on categorization or (2) an internal Facebook product feature that is more systematic and transparent than the current 'See fewer posts like this' feature. To create a minimally viable product (or MVP), we built a Chrome browser extension that labels posts in the Facebook News Feed.

1. Introduction

1.1. Motivation

The Facebook News Feed is a machine learning algorithm that ranks posts from friends and pages in some order based on the predicted probability about what users want to see. At the moment, there is not much user configuration of the content, except for indicating you would like to see less posts by certain friends or pages. This project demonstrates how either a third-party or Facebook could apply text categorization algorithms to the creation of a feature to filter news content based only on the news storys posts.

We imagine a feature in which users select the type of news content they would like to see in their Feed. For example, a user might not want to see any entertainment news and could de-select that category. A third-party could create a browser extension that removes the HTML component for the selected types of news article. Likewise, Facebook could implement this as a feature within News Feed.

University of Pennsylvania, CIS 419/519 Course Project.

We seek to address the problem of limited filtering capabilities with a machine learning centric product. The solution includes a filter for users (see Section 4) that uses the outputs of our algorithm to label existing posts.

1.2. Data

1.2.1. Initial Data

To get an understanding of the problem and the application of the classifiers, we used UCI Machine Learning Repository dataset that contains 425,000 news headlines from 2014 collected from a news aggregator. In order to validate the effectiveness of our categorization methods, we built a validation set of New York Times headlines labeled based on the section in the newspaper from January 2017 to October 2017 via the New York Times API. Our classifiers (discussed in Section 2) performed quite well against this out-of-sample dataset so we moved forth with our project with Facebook data.

1.2.2. Actual Data

We used Facebook's Graph API to scrape 77,311 live posts created across six major categories: politics, entertainment, business, sports, health, and science. We selected pages of high perceived credibility that could closely represent other pages in the category. For example, we scraped the Facebook posts made by Barak Obama, Donald Trump etc. as part of the politics category. For the business category, we scraped data from JP Morgan Chase, General Electronics and Shell, among others.

2. Approach

2.1. Feature Extraction

After collecting and scraping the data from UCI, the New York Times, and Facebook, we needed to create a meaningful representation of the textual data. In order to create meaningful features, we eliminated English stop-words (e.g. "the", "it", "a", "this" etc.), converted all letters into lowercase, and removed non-printable characters, emoticons, and URLs. We used a TF-IDF (term frequency-inverse document frequency) representation for the features

and attempted different n-gram models (for $n = 1, 2, 3, 4,$ and 5).

We studied the effect of using n-gram features in our model but found that this representation did not improve the performance of any of our model regardless of n.

2.2. TF-IGM

A recent approach used to represent text data is the Term Frequency Inverse Gravity Moment representation (Chin et al., 2016). The TF-IGM representation uses the term distribution across different classes of text and has been shown experimentally to outperform TF-IDF on the common datasets used in text classification. These findings encouraged us to implement the TF-IGM representation, in addition to TF-IDF, and observe the results on the models for the same classification models.

Due to the computational cost of TF-IGM, we evaluated this approach on a smaller portion of our dataset described in Section 1.2.2. TF-IGM representation weighs a term based on the importance of the term within a given class. This is calculated by computing the frequencies of each term in every class and sorting it in ascending order $[f_{k1}, f_{k2}, f_{k3}...]$ where f_{ki} is the frequency of the term t_k in class i.

The IGM component is defined as follows:

$$\text{IGM}(t_k) = \frac{f_{k1}}{\sum f_{kr} * r} \tag{1}$$

where r is the rank of the frequency. Then, TF-IGM of the term t_k is obtained using

$$\text{TF-IGM}(t_k) = tf \times igm(t_k) \tag{2}$$

3. Results

We utilized three classification algorithms: Logistic Regression, SVMs and Nave Bayes. In addition to implementing the TF-IDF representation on the Facebook data we also tested TF-IGM (Inverse gravity moment).

3.1. Logistic Regression

Logistic regression is a popular model for categorical problems and performs particularly well for large-scale text problems. With the TF-IDF representation of the Facebook dataset we tuned the regularization parameter C and using 10-fold cross-validation. Given our search grid selected an optimal value of $C = 5$. Figure 1 shows there is a decreasing performance improvement as C increases. Using the TF-IDF representation, logistic regression yielded a cross-validated accuracy of 83.94%.

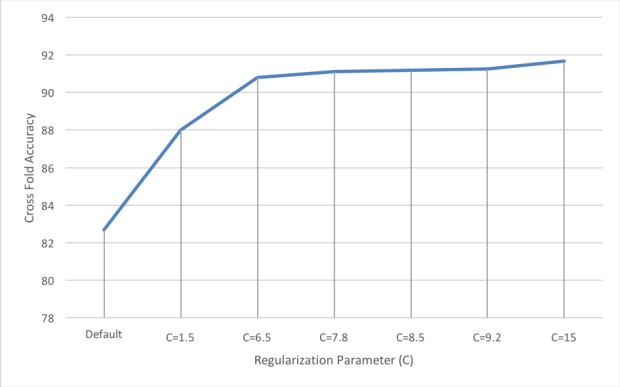


Figure 1. Logistic regression performance vs regularization parameter C

3.2. Multinomial Naïve Bayes

Naïve Bayes is a family of classifiers that rely on applying Bayes Theorem by assuming conditional independence among features. Multinomial Naïve Bayes is used when the data may be believed to have come from a multinomial distribution as is the case for multi-class classification problems.

With the TF-IDF representation of the Facebook dataset we tuned the regularization parameter α and using 10-fold cross-validation. Given our search grid selected an optimal value of $\alpha = 0.25$. We observed a decreasing performance as α increases past 0.25. This is logical as the role of α is to perform Laplacian smoothing on the probability, which means that it accounts for missing features in the dataset. Using the TF-IDF representation, multinomial Naïve Bayes yielded a cross-validated accuracy of 83.72%, which is extremely similar to the accuracy of logistic regression.

3.3. SVM with Cosine Similarity Kernel

We used SVM with cosine similarity kernel to determine how correlated the documents were and hence separate them into classes. As SVM with the cosine similarity kernel (as implemented in `sklearn`) is very slow with large datasets, we split the Facebook headlines datasets into smaller samples, then performed 10-fold cross validation as with logistic regression and Multinomial Naïve Bayes. We then averaged the performance metrics across each of the samples. We assume that SVM would have performed better if it the entire dataset was used.

Using the TF-IDF representation, multinomial Naïve Bayes yielded a cross-validated accuracy of 74.76%, which is much lower than the other two models but is likely a function of the smaller sample size.

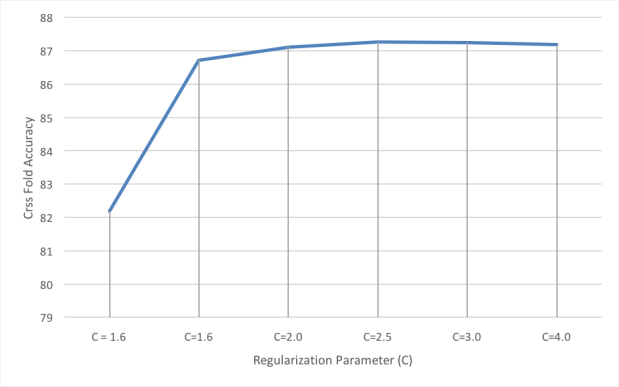


Figure 2. SVM with cosine similarity kernel performance vs regularization parameter C.

3.4. Performance with TF-IGM

The three models described above were implemented again, using TF-IGM representation discussed in Section 2.2. In order to use this more computational intensive representation, we had to limit our data. For our analysis, we used 65% of the 77 thousand posts we acquired through Facebook's Graph API. We obtained very similar performances for both logistic regression and Multinomial Naïve Bayes. Table 2 shows the results for these models using TF-IGM.

3.5. Comparison

We found that the Multinomial Naïve Bayes and Logistic Regression models perform quite well.. SVM was skipped because of extremely heavy computational requirements for the kernel, specifically with our 77k documents. The new TF-IGM representation performed extremely well (Considering much lesser number of training samples) resulting in the following 10-fold cross-validation accuracies. Thus, we explore other performance metrics of the logistic regression and Multinomial Naïve Bayes models on the TF-IDF representation.

Table 1. Summary of performance metrics for the three types of models investigated using TF-IDF representation.

Model	Metrics	
	Accuracy	**F1 Score**
SVM with Cosine Similarity	0.7476	0.7243
Multinomial Naïve Bayes	0.8372	0.7963
Logistic Regression	0.8394	0.8076

Table 2. Summary of performance metrics for the three types of models investigated using TF-IGM representation.

Model	Metrics	
	Accuracy	**F1 Score**
SVM with Cosine Similarity	0.7929	0.7842
Multinomial Naïve Bayes	0.8173	0.7538
Logistic Regression	0.8095	0.7791

In calculating precision, recall, and the F1 score for each model, we use weighted averaging because of the imbalance in the number of classes in our Facebook posts dataset. Next we created ROC curves using a 80/20 train/test split that enabled us to see the performance for each category of posts. We note that the curves appear about the same for and Multinomial Naïve Bayes with regards to the relative performance of each category (i.e. we see that sports performs the best and that health performs the worst).

Finally, to explore how the model are separating the categories, we created word clouds showing the terms with the highest TF-IDF value for each of the categories. Included in Figure 4 are the word clouds for science and politics. Unsurprisingly, the word with the highest TF-IDF value for science is "new" and for politics it is "president". These are merely exploratory and confirmation visuals, but they help demonstrate how logistic regression and Multinomial Naïve Bayes are using the TF-IDF representation.

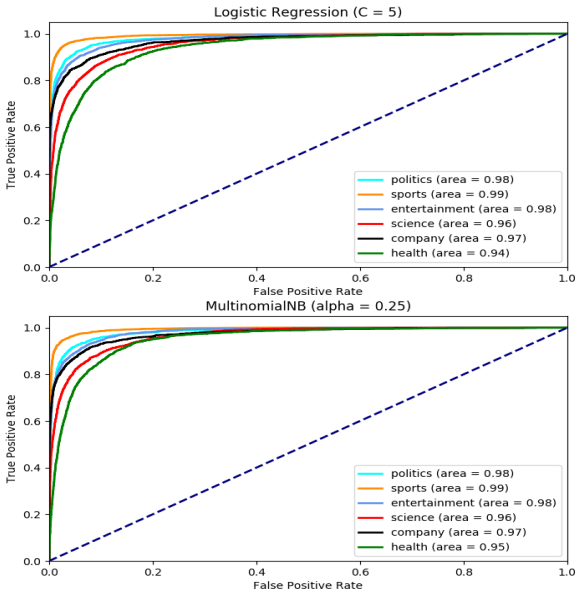


Figure 3. ROC Curves for Naïve Bayes and Logistic Regression Models.

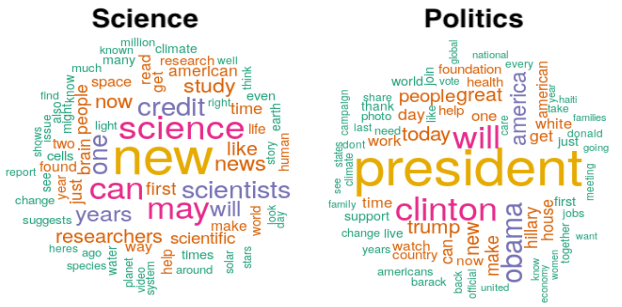


Figure 4. Word Cloud for two of the six categories of posts

4. Product - Chrome Plugin

We developed a Google Chrome browser extension that labels Facebook posts into one of the categories in real time.

4.1. Mechanics

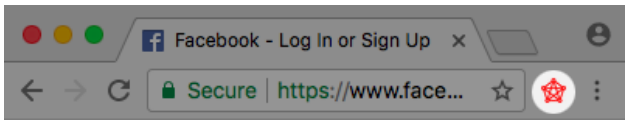


Figure 5. The extension is displayed in the browser's toolbar.

Our extension runs a local classification server, which uses Python's pickle library to read the vectorizer and the model trained on 77k instances. When a new Facebook post is encountered, the plugin extracts the headline text from the post and sends it to the server, which cleans and vectorizes the text to the format our model was trained on. The model predicts the instance and reverts with class label almost instantly.

4.2. Application

The user encounters a seamless experience. The posts are categorized in real-time with the category being reflected on the top-right corner for each of them.

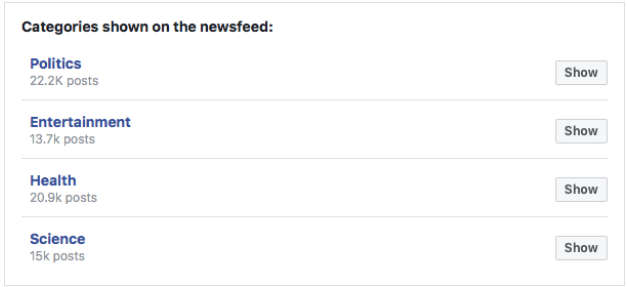


Figure 6. Mockup of the proposed option to allow user to choose the categories.

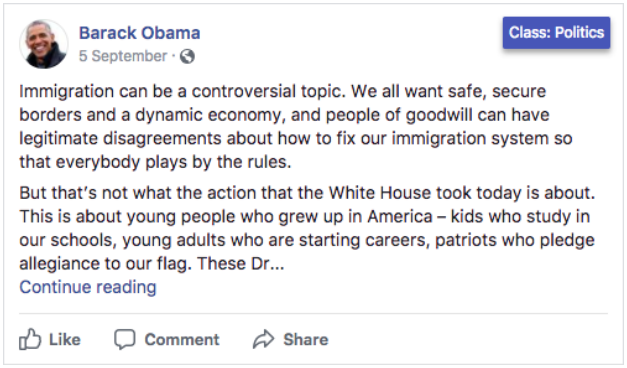


Figure 7. The posts are automatically labelled with the predicted category in real-time.

5. Discussion

5.1. Recommendation

Given that both Multinomial Naïve Bayes and logistic regression require approximately the same amount of time to train, we would recommend using the logistic regression model as it results in slightly better performance.

5.2. Limitations

We had to manually find pages of certain categories on Facebook that would fit into each of the six categories used. In the future, we would like to use a more systematic approach, where we don't have to restrict the number of categories manually. Since our dataset is large and too sparse for some of the models to learn quickly, we would like to use techniques to handle large-scale data for better representation of sparse data. TF-IGM representation took an extremely long time to train even for a small amount of training samples and it would be highly inefficient to run of large datasets.

5.3. Next Steps

We propose a Facebook-wide feature where users are able to select the categories of posts they would and would not like to see. With the six current categories, the feature could be represented as in Figure 6.

While TF-IGM is an interesting representation, a more efficient processing scheme needs to be developed for this to be used in practice.

Attempt to use a neural network as a classification algorithm.

Posts can fall under multiple categories instead of just a single category. A multilabel classification technique must be employed for handling such situations.

References

Chin, Kevin, Zhang, Zuping, Long, Jun, and Zhang, Hao. Turning from tf-idf to tf-igm for term weighting in text classification. 66, 09 2016.

Dilrukshi, I., Zoysa, K. De, and Caldera, A. Twitter news classification using svm. In *2013 8th International Conference on Computer Science Education*, pp. 287–291, April 2013. doi: 10.1109/ICCSE.2013.6553926.

Facebook. Facebook graph api. https://developers.facebook.com/docs/graph-api/.

Kroha, P. and Baeza-Yates, R. A case study: News classification based on term frequency. In *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*, pp. 428–432, Aug 2005. doi: 10.1109/DEXA.2005.6.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

New York Times Developer Network. New york times article search api. https://developer.nytimes.com/article_search_v2.json.

Oremus, Will. Who controls your facebook feed. *Slate*, Jan 2016. URL http://www.slate.com/articles/technology/cover_story/2016/01/how_facebook_s_news_feed_algorithm_works.html.

Social Media Content Filter

Problem

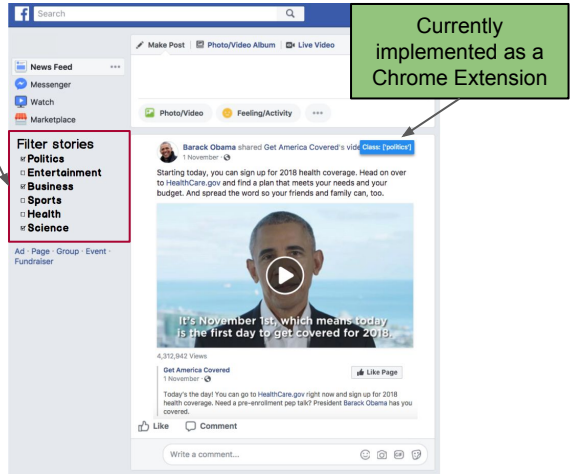
Facebook currently has limited content filter options

Only elect for fewer posts "like this"

Solution

New product feature allowing filter by high-level content type

Filter to the content you want



Currently implemented as a Chrome Extension

Text Classification Solution

Process

- Gathered page post headlines Facebook's Graph API
- Used TF-IDF and TF-IGM (term-frequency inverse gravity model) representations
- Tuned regularization parameters
- Built Chrome extension to demonstrate use
- Recommendation: **Logistic Regression for use in product**

Results

Model	Cross-Validated Accuracy
SVM with Cosine Similarity	0.7476
Naive Bayes	0.8372
Logistic Regression	0.8394