Facebook Post and Comment Generator for News Articles

Fernando Estrella, Erica Granor, Vitor Mouzinho

December 2019

How to Run the Code:

Install all packages mentioned below. Open the all_together.py file and run it. Follow instructions in the console.

Requirements:

SpaCy

Pickle

Nltk

NewsApi

BeautifulSoup

Installation:

pip install spacy

python -m spacy download en_core_web_sm

pip install beautifulsoup4

pip install lxml

Contents of GitHub Repository:

https://github.com/fernandoaestrella/nlp-final-project/

Introduction

Social media is a widely used platform for learning about the news. Many news organizations post short descriptions, mostly written by humans, of news articles to engage viewers. We have created a program that will summarize news articles into brief “posts”, which each include a summary, an AI-generated short comment, and link to the article. Additionally,

we have generated a response comment for each article based on whether the article content is good or bad.

Methods

This project was split into several components: corpus identification and generation, sentiment analysis, a summary generator, and comment generator. All group members worked collaboratively and reviewed each other's contributions, but ownership of the following pieces were as follows: ArticleSummarization, summary generator, collecting articles from News API, parsing news articles, extracting information from news articles, PostGenerator and words following predicted sentiment, also putting everyone's code together, to make the project run by Vitor; sentiment analysis implementation, classifier model selection, entity extraction, GitHub/overall project management and integration by Fernando; corpus identification, sentiment prediction base code, writing, extraction of words of given sentiment from article, secondary comment generation by Erica.

To obtain news articles, we chose the Google News API (newsapi.org), because it provides a comprehensive collection of current news articles from different sources. It provides many useful elements, like the full content of the article, topic, date and source. Other sources considered include the New York Times API, which did not provide full content of articles, and the BBC, which did not provide coverage from different sources. Additionally, the New York Times provided archival news, whereas for social media it is important to have an up-to-date news feed.

For the sentiment analysis, we used SciKitLearn to train and test a subset of the same classifiers for Assignment 5: Naïve Bayes and Decision Tree, for feature sets of raw counts of all words in the news articles corpus, binary counts of all words in the news articles corpus, and raw and binary counts filtered for words in the news articles with SentiWordNet positivity or negativity scores of >= 0.5.

We tried to train on a set of tagged news articles from a project described here: https://www.aclweb.org/anthology/Y16-2013.pdf. We decided to do this because we thought it would give higher accuracy when trying to predict the sentiment of a news article since the

models would be trained with actual news articles instead of with movie reviews. The articles in the ACE 2005 dataset are labeled with one of six possible emotions in this project. The labels for each of the documents can be found here: https://github.com/MingleiLI/ACE2005_emotion_corpus/blob/master/EmotionLabel. We grabbed the source files from here: https://github.com/Aureliu/BIU-RPI-Event-Extraction-Project/tree/master/ACE_EVENT/corpus/orig. We used only the documents tagged "joy" and "sad" to form a training and test corpus of 159 documents. We separated the corpus in 2 parts: 75% of the corpus would be used for training and 25% would be used for testing. We achieved the following accuracy results:

Classifier model	Accuracy Score 1	Accuracy Score 2	Accuracy Score 3	Average
all words raw counts bayes	0.7	0.65	0.725	0.6916666667
all words raw counts tree	0.45	0.6	0.7	0.5833333333
all words binary bayes	0.65	0.775	0.7	0.7083333333
all words binary tree	0.525	0.65	0.55	0.575
SentiWordNet words raw counts bayes	0.65	0.6	0.525	0.5916666667
SentiWordNet words raw counts tree	0.6	0.425	0.575	0.5333333333
SentiWordNet words binary bayes	0.725	0.6	0.7	0.675
SentiWordNet words binary tree	0.625	0.5	0.625	0.5833333333

The all words binary bayes model had the highest accuracy, at 70.83333333%. However, we ended up using the second highest option because it actually gave a mix of predictions

(positives and negatives) based on the input topics. Given more time, we could more carefully consider other models on which to run the sentiment analysis.

The predicted sentiment of the article was used to inform the comments generated by our comment generator. Depending on whether the article is determined positive or negative according to our machine learning algorithm, that positivity or negativity is translated into a short comment. Entity extraction from SpaCy was used to identify key terms in the article that would describe briefly what the article is about. These SpaCy entities were “PERSON” (person) “ORG” (organization), and “GPE” (geo-political entity). For example, if the string, “The U.K. had recent discussions with Trump” were an article and it were tagged negative, the comment would read, “This is a terrible article about Trump”. If the string were “The. U.K. voted on tariffs this morning,” and this were marked positive, the comment would read, “Check out this great article about the U.K.” “Great” denotes positive sentiment and “terrible” denotes negative sentiment. People, followed by organization, followed by geopolitical entity are prioritized in the comment, because they increase in specificity. The comment generator produces one comment per Facebook post.

The summary generator was developed by scraping the website of the url where the article came from. Vitor used various different scraper APIs until he found one that the website doesn’t recognize as a bot; from there we used Vitor’s Article summarization class to summarize the article based on frequencies. Then with the summarization of the article and sentiment analysis of the article Vitor chose one specific sentence that had the most negative or positive sentimental words and choose that sentence to be the post. It takes a text file of the news article and the URL, and produces a short summary and link, suitable for posting immediately to Facebook.

Discussion and Future Work

- We found a small corpus of news articles with tagged sentiments that gave us good enough results. Given more time, we could manually tag more articles or look for more tagged corpora.

- Determine whether the methods used to do a sentiment analysis on movie reviews are directly applicable to news articles. Are there any common elements of news articles (e.g. common vocabulary and connotations, customs/rules) that affect sentiment in a news context but would not be picked up by SentiWordNet?
- Determine how to increase the accuracy of the models, perhaps by processing the input article text to make sure it doesn't contain some weird characters that the classifier is not expecting.
- We considered using the Markov Chain algorithm to generate the descriptive comment for the post and even had running code for it but the results, while sometimes fun, many times lacked sense and would not be a very appropriate output for the output of the program. We also considered adding words that had the predicted sentiment in the generated comment but that would have further complicated the comment generation because sentences are not only generated with words of a certain sentiment but with neutral words as well. Maybe inputting both sets to the dictionary needed for the Markov Chain and making sure words from both sets have an equal chance at being chosen would have given slightly more reliable results.
- Given more time, we could generate more custom/relevant comments, such as customizing the sentiment to the magnitude of the positivity or negativity of the article (e.g. "This article is moderately promising/good/great/fantastic") and be more descriptive with the entities.