Daniel Foster
Machine Learning
26/10/2017
Using a Support Vector Machine for Classifying Facebook Post Reactions

Introduction

The present study investigates the use of a Support Vector Machine (SVM) for classifying Facebook comments according to the reactions to the post to which they reply. Data was scraped from the Facebook page PBS NewsHour, a page for the news program of the same name shown by the American TV channel Public Broadcasting Systems (PBS). The tool used to gather the data is called Facebook Page Post Scraper (Woolf 2016) which allows for the downloading of all statuses from a Facebook page and their corresponding comments along with metadata that includes user reactions to the post. In following a traditional Machine Learning example from Pang et. al (2002) a Support Vector Machine was chosen in order to compare the results of using social media data to the classic example of positive and negative movie reviews. The SVM was implemented in Python using sci-kit learn's LinearSVC. An accuracy of 82% was achieved when classifying so-called 'angry' reactions against 'love' reactions using unigrams as features, very comparable to Pang, et. al's result of 83% using unigram data. Pang, et. al.'s result was then actually surpassed with classification of 'sad' against 'angry' posts, the two reactions with the most comment data, with 86.5% accuracy.

Background

In this section the methods and data being used, a Support Vector Machine and Facebook posts and comments with reaction metadata respectively, will be discussed. The reasons for their choice and some of the background of each topic will be explained.

Support Vector Machines

The use of a Support Vector Machine to classify the posts is based on the result achieved by the paper being followed as an example in this study, "Thumbs up? Sentiment Classification using Machine Learning Techniques" by Pang, et. al. (2002). According to Cristianini and Taylor (2000) "Support Vector Machines (SVM) are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory." Intuitively an SVM tries to optimize the gap between possible classification lines between groups of points. The SVM looks for the largest or widest possible gap between the points of the classes. Pang, et. al. (2002) use an SVM in comparison with two other traditional machine learning methods for classification: Naïve Bayes and Maximum Entropy. Naïve Bayes uses the maximum likelihood of an event to classify the next event, with the naiveté being that the classifier does not look at previous events for information. Maximum Entropy classifiers, on the other hand, do not assume conditional independence of events. They find that SVM performs the best out of these three traditional models, with 82.9% accuracy using unigrams as features and 77.1% with bigrams. These are the two feature sets being compared in the present study.

Social Media Data

The major difference with the classification at hand is the type of data being used. It will be interesting to see whether social media data can mimic the results seen with movie reviews, and if the sentiments being tagged by Facebook's reactions show themselves as useful for sentiment classification. In a blog post by Max Woolf titled "Classifying the Emotions of Facebook Posts Using Reactions Data" which spurred the idea for this study, it is suggested that Facebook reaction data, the classification of the reaction of posts as 'like', 'love', 'haha', 'wow', 'sad', and 'angry', could be very useful for sentiment analysis. It is hoped that this exercise can help to explore the validity of this claim. First though, data will have to be gathered and processed for proper use.

With internet or social media data, creating a useful corpus is a complex task and requires special attention to effectively close the gap between raw and uniform data. Data from the web is noisy, mixes modes, and can be unbalanced and unrepresentative (Leech 2007, Hoffmann 2007). Common challenges to overcome in this regard include the effect of timing, unbalanced language use, special symbols and encoding problems, as well as automated or repeated material. Since the characteristics of data on different platforms are decided upon solely by the business providing the service in question, further challenges can include getting helpful metadata, following a conversation, and finding good methods for accessing data. Zappavigna (2012), for example, encounters some of these challenges in exploring her HERMES corpus of 100 million words of English Twitter data. She shows that in certain cases, timing, for example, can be a boon or a hinderance, especially with Twitter data, skewing the balance of the corpus (2012: 18). In the same vein, most of these challenges can be mitigated or ruled out when the strengths of social media data such as quantity, searchability, and speed and ease of gathering data are used to their full advantage. Finding the right tool can also make a big difference. For the purposes of the current study, the Facebook Page Post Scraper (Woolf 2016) ideally was able to link comments to posts by their identification number and show the reaction data for a given post. The scraper provides a python script able to easily download first the posts of a given page, once the user provides their Facebook API key and specifies the name of the page. It then exports the data to a .csv for further use. The user can then run a similar script to download the comments which correspond to all the posts downloaded and get another .csv file.

Methodology

In this section we will discuss the methodology being used in the study. First and foremost, data processing and cleaning will be discussed, as it is a major part of using social media data, as mentioned above. It is also very important in general when using a classifier on natural language data, as differences in the appearance of words can cause loss of good data. Further, the feature selection and final data will be discussed.

Data Processing and Cleaning

The Facebook Page Post Scraper was used to download posts from the PBS NewsHour Facebook page. A news source was chosen in hopes of receiving an even spread of reactions for classification. This was based on Woolf (2016) who saw the most even spread of reactions by using the Facebook page for CNN. Woolf also advised to ignore likes, as they are uninformative data that get in the way of examining the other five emotion-based reactions. Posts were classified as a given reaction only if 75% or more of the reactions for a post were one specific reaction. For example, if a post was 20% haha, 20% wow, 20% angry, 20% sad, and 20% love, it would be thrown out. From that point, the scraper was used to download comments corresponding to posts. These comments were imported to python and saved only if the 'status_id' metadata, the identification number of a given post, matched one of the classified posts. Comments were then grouped by the reaction to which they belonged.

Before saving the comments to text files for each reaction, much of the noise in the comments had to be removed. This included hyperlinks, punctuation, and special occurrences like '[[PHOTO]]' appearing as a comment when a user posted a photograph alongside or without any text. The text was then all lower-cased. This normalized the data more or less to solely words. Emojis seemed to be an issue, as sometimes they appear as a unicode string such as '\U000' followed by a code, however if encoding issues are avoided, they can still be processed by python and printed to a text file where they can appear correctly. This was an important advantage, as emojis could be potentially very useful data for sentiment analysis. Once the data was cleaned, each comment was written to one line of a text file along with the sentiment label: LOV, HAH, WOW, SAD, or ANG after a pipe separator.

Daniel Foster
Machine Learning
26/10/2017
Using a Support Vector Machine for Classifying Facebook Post Reactions

Feature Selection and Data

Feature selection was done to mirror the feature selection in Pang, et. al (2002). Two of the main features used in that study were chosen here, unigrams and bigrams. Feature selection is an important part of using an SVM, as it can drastically alter the results. Other possible features would have been other word n-grams, character based features and n-grams of characters, or parts of speech, which were also used by Pang, et. al. but due to time constraints were not used in the present study. Different feature selections can give more information to the model in order to help it make its predictions. The format of messages could also affect which is the most useful feature. For example, considering classifying authors with long texts higher order n-grams may be more useful in the first case, as phrases used by an author may be more salient than individual words.

The code will also use TFIDF, a rating of the importance of words in a corpus which balances the term frequency (TF), and inverse document frequency (IDF) of words. This ranking of word importance will give the model the information it needs to decide if a given word in an angry comment shows with high likelihood that comments containing said word are also angry. The model will also select from all words or bigrams only those with a high TFIDF ranking, cutting out the individual features below a specified TFIDF threshold to make its decisions.

The data used uses five classifications, one for each of the reactions, but in order to compare more directly with the results of Pang, et. al., the top reactions will also be compared in binary classification examples; first love vs. angry, and then sad vs. angry which it was thought could cause some problems for the classifier. Classes will also be balanced in the binary comparison since the amount of posts per reaction might not be the same.

Results

In this section, the characteristics of the data will first be shown. Then, results for different feature selections and using different reaction sets will be discussed. First the results with all five reactions will be shown along with their accuracy ratings, followed by love vs. angry, and sad vs. angry. The top twenty-five TFIDF ranked words will also be shown and discussed in the following section.

Corpus

The following are comments from the corpus with their respective sentiment tags:

- hahahahahahaha | HAH
- still hate him | ANG
- awesome a milestone and celebration for all women | LOV

The PBS NewsHour page returned an unfortunately rather unbalanced profile of reactions. Table 1 shows the number of posts per reaction as well as the number of comments.

Table 1:

Reaction	Posts	Comments
Love	**362**	**624**
Haha	**16**	**93**
Wow	**25**	**79**
Sad	**98**	**1061**
Angry	**185**	**1541**

Haha and wow posts were quite underrepresented, however Love, Sad, and Angry did give a significant amount of data. This probably helped cause the difference in accuracy seen in the full reaction set shown in the section to follow.

Full Reaction Set

Using the SVM to classify the full set of five reactions yielded results much poorer than Pang, et. al. (2002). Table 2 shows the comparison of accuracies and feature sets from this first experiment.

Table 2:

Feature Set	SVM for All Five Facebook Reactions	Pang , et. al.
Unigrams	64.9%	82.9%
Bigrams	32.9%	77.1%

The poor results here could be for a few reasons. Namely, as mentioned, the data for some of the reactions was not enough to help the model distinguish between and correctly classify haha and wow posts. With more time, a larger page with a more even spread of reactions would have been chosen such as CNN. It would be interesting to see how the SVM really handles five classes as compared to two binaries, but I suspect it would still have trouble as the difference between 'haha' and 'wow' may be quite small. We will explore this, though, by comparing sad and angry comments further along, since they have the most data.

Love vs. Angry Reactions

Here the classifier performed much better, and much more comparably to Pang, et. al. when considering the unigram model. The problem at hand was much more related to the positive and negative sentiment seen in Pang, et. al. so this was quite expected. Table 3 shows the results of this experiment:

Table 3:

Feature Set	**SVM for Love vs. Angry Posts**	**Pang , et. al.**
Unigrams	82.0%	82.9%
Bigrams	67.2%	77.1%

Here the bigrams, meanwhile, performed considerably better than with all five reactions, but still not as well as Pang, et. al. The amount of data and the binary nature of the reactions seemed to be the causes of the stark improvement with this problem. In the next section we will see how the SVM performs with more data, but with a less obviously opposed set of reactions.

Sad vs. Angry Reactions

In Table 4, we see the results of classifying sad and angry posts, the two reactions with the most comment data. Somewhat surprisingly the unigram model outperforms both the love vs. angry posts and Pang, et. al.'s baseline.

Table 4:

Feature Set	**SVM for Sad vs. Angry Posts**	**Pang , et. al.**
Unigrams	86.6%	82.9%
Bigrams	34.1%	77.1%

Confusingly, the bigrams perform almost as poorly as with all five reactions involved. The unigram result could be the best simply due to the large amount of data, perhaps the model gets confused here if bigrams start to create overlap between the two emotions.

Discussion and Conclusions

In this section, the top words for each experiment will be discussed, as well as linguistic speculation as to why the models performed as they did. Conclusions about the use of Facebook data as compared to movie reviews will then be drawn.

Full Reaction Set

Here, the lack of data for some of the reactions was seemingly the root of the problem. However, dealing with five classes at once could have also been troublesome for the SVM. In Figure 1, we see the top twenty-five TFIDF words from this experiment.

Figure 1: Top Twenty-Five TFIDF words/bigrams:

> **Unigrams:** trump, s, nt, people, just, thank, good, like, moron, idiot, need, does, sticker, love, know, news, man, president, god, did, pbs, sad, America, right, country
>
> **Bigrams:** fake news, ca nt, does nt, trump s, Puerto rico, did nt, liar chief, climate change, birds feather, thank god, chief staff, god bless, looking forward, white house, birth control, american people, register vote, nt Canada, trump trump, president trump, america great, nt want, gon na, nt care, enemy American

The lists of words here seem informative for the most represented reactions, sad, angry and love, but there is definitely no word that intuitively stands out representing haha or wow here. This shows that the imbalance of the corpus probably caused problems when examining all five reactions at once.

Love vs. Angry Reactions

When considering only angry and love reactions, the informativity of the unigrams and bigrams becomes much more evident. In Figure 2 we see the top twenty-five of each.

Figure 2: Top Twenty-Five TFIDF words/bigrams:

> **Unigrams:** trump, thank, idiot, love, nt, need, wonderful, moron, good, people, like, tax, beautiful, president, god, abortion, amazing, pbs, great, money, did, america, stop, know
>
> **Bigrams:** thank god, liar chief, birth control, register vote, trump trump, thank senator, sesame street, president trump, looking forward, god need, need senate, susan Collins, despicable

lunatic, impeach moron, john kraut, did nt, middle class, thanks pbs, sick winning, thank susan, voice reason, tax plan, pedophilia betterment, betterment society

Unigrams show a nice clear split according to linguistic intuitions with idiot, moron, stop, and considering the divisiveness of the context, trump, president, and abortion being clearly angry words. Meanwhile, love, wonderful, good, like, beautiful, amazing, and great are very clearly associated with love posts. Interestingly the bigrams also seem quite informative, but maybe there is imbalance which causes the lower result here. Liar chief, despicable lunatic, and impeach moron, for example all seem quite clearly angry, but there isn't such a clear representation of love posts here.

Sad vs. Angry Posts

The informativity of features between sad and angry actually becomes clearer when considering the top TFIDF words.

Figure 3: Top Twenty-Five TFIDF words/bigrams:

Unigrams: s, trump, nt, people, idiot, just, moron, does, sad, good, god, president, know, did, right, money, America, country, fires, california, tax, make, man, news, wrong

Bigrams: does nt, birds feather, fake news, liar chief, chief staff, puerto rico, climate change, birth control, did nt, register vote, nt canada, trump trump, president trump, american people, god bless, enemy american, looks like, live s, middle class, pregnant wife, nt make, global warming, robert bonham, impeach moron, make stop

The angry words are evident with the same highly ranked words as in the previous section. The sad words, other than 'sad', become somewhat clearer with context, for example fires and california being related to a recent natural disaster. This seems to show that topics can probably drive the reactions quite a bit, and 'sad' words can, in the end, be much more contextual than 'angry' words which are evenly contextual and intuitive (trump vs. moron). The bigrams, however, seem just as informative as the unigrams, so the poor result is difficult to explain. I would imagine that 'god bless' and 'pregnant wife', for example, may be associated with sad posts if the posts in question discuss families losing loved ones. Perhaps, though, many of the other highly ranked bigrams overlap between these reactions.

Conclusions

The present study investigated the use of a Support Vector Machine (SVM) for classifying Facebook comments according to the reactions to the post to which they reply and found mixed but interesting results. In the end, using unigrams as features, and binary comparisons of reactions, the model performed quite well, even outperforming Pang et. al. (2002) with sad vs. angry reactions as classes.

More data seemed to be the most important factor here, so it would be interesting to continue exploring social media data, combining results from multiple pages, and comparing across pages. It seems to have been shown, though, that reaction-based Facebook data is quite compelling and useful in the field of sentiment analysis, just as Woolf (2016) suggested.

Daniel Foster
Machine Learning
26/10/2017
Using a Support Vector Machine for Classifying Facebook Post Reactions

Bibliography

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods.* New York, NY: Cambridge University Press.

Hoffmann, S. (2007). Processing Internet-Derived Text - Creating a Corpus of Usenet Messages. *Literary and Linguistic Computing, 22*(2), 151-65.

Leech, G. (2007). New resources, or just better old ones? The Holy Grail of representativeness. In N. N. M> Hundt (Ed.), *Corpus Linguistics and the Web.* Amsterdam: Rodopi.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning. *EMNLP 2002.* Philadelphia.

Woolf, M. (2016, June 27). *Classifying the Emotions of Facebook Posts Using Reactions Data*. Retrieved from minimaxir.com: http://minimaxir.com/2016/06/interactive-reactions/

Zappavigna, M. (2012). *Discourse of Twitter and Social Media: How We Use Language to Create Affiliation on the Web.* London, New York: Continuum.