Kimya Buckner
Ling 406
Spring 2020

Sentiment Analysis Term Project

Introduction

Sentiment analysis has become an essential process across a range of sectors in our society. It is a valuable tool in gathering information on public opinion that is used to help drive effective and informed business decisions (Walther 2019). Sentiment analysis, also known as opinion mining, is the process of computationally identifying and categorizing opinions expressed in a piece of text, to determine whether the writer's attitude towards a particular topic or product is positive, negative, or neutral (Girju 2020). Today, thanks to the prevalence of technology, more opinions are being shared online than ever before. For example, social media sites contain many millions of status updates and comments that can be mined to reave the general sentiment on any number of topics. Additionally, online reviews are especially suited for sentiment analysis, as the writers themselves typically indicate a sentiment of positive, neutral, or negative through the designation star rating. This plethora of available texts to be mined presents many opportunities to perform sentiment analysis.

Sentiment analysis has several unique applications. Historically, sentiment analysis has aided enterprises to gauge public opinion, conduct market research, determine advertisement placement, and monitor brand and product reputation. Sentiment analysis can also be found in sectors like politics and stock trading. With such a wide range of applications, it is important to continue to invest in the research on sentiment analysis. Having an understanding of public opinion is necessary to ensure that our society is a reflection of the larger population's desires and attitude.

Problem Definition

In the context of computational linguistics, sentiment analysis is the use of algorithms to classify a piece of text as positive, neutral, or negative. It combines natural language processing and machine learning techniques to assign sentiment scores to categories within a sentence or phrase. The process typically consists of breaking down a sentiment annotated corporus into analyzable tokens, training the algorithm by assigning a sentiment or polarity score to words, phrases, or pattern tokens, and testing the trained algorithm on new data to measure the accuracy with which it classifies the attitude of input text. Typically, types of input for semantic analysis are reviews (such as movie, product, and restaurant) and social media posts on websites like Facebook and Twitter. The output for semantic analysis would be a category classification for the input data.

Previous work

There have been several notable examples of sentiment analysis over the past few years. One such instance was the use of sentiment analysis by Kraft in 2003 to detect public outrage about the presence of trans fat in Oreo Cookies (Schweidel and Moe 2014). As a part of its normal operations, Kraft would scan social media websites for texts relating to their products. They would then perform sentiment analysis on texts identified as relevant to the company. It was through this process that a negative sentiment towards the trans fat in Oreo cookies was identified. Guided by this data, Kraft proceeded to remove trans fats from its snack products

(Schweidel and Moe 2014). This is a concrete example of sentiment analysis aiding corporations to make decisions that reflect the preferences of its customers. Companies have discovered that through the sentiment analysis of social media, they are provided with a more holistic understanding of their customer base's attitude towards their products. It is for this reason that an entire industry of "social listening" has emerged in recent years. Companies such as SproutSocial, HubSpot, and TweetReach are social listening companies designed to help you, "inform your business strategy with social listening" and provide your company with "analysis and actionable responses" to help meet the needs of your customers. With the plethora of customer information available via social media, and the numerous companies that have made it their mission to perform customer opinion mining, it seems as if sentiment analysis will only become more and more prevalent across different domains of society. For this reason, it is likely that funds and research will continue to be dedicated to sentiment analysis for the foreseeable future.

Approach

This paper will discuss sentiment analysis trained on a set of annotated movie reviews. The corpus of movie reviews was obtained from a 2004 Cornell polarity data set. This set of data contains 2000 movie reviews -- 1000 designated as negative and 1000 designated as positive (Pang, Lee, and Vaithyanathan 2004). The approach used to implement this sentiment analyzer is a Bag-of-Words model. This model consists of creating a lexicon of relevant words found in the documents and scoring those words in some way. This is a relatively simple and straightforward approach that has been successfully used in sentiment analysis many times. However, this model does have several limitations. For example, the choice to use a bag-of-words disregards word order. This results in context being completely ignored. Additionally, the vocabulary in a Bag-of-Words approach must be very carefully designed. The vocabulary must be meticulously composed, otherwise the size of the vocabulary could easily become too large. It is for this reason, when using a bag of words approach, that diligent preprocessing of the data is such an essential requirement.

The input data is preprocessed in a number of ways. The first step of preprocessing performed is the removal of punctuation from the input data. This is done because most punctuation such as periods, commas, and apostrophes have little to no impact on the sentiment being conveyed by a piece of text. Removing punctuation reduces the number of tokens that will need to be analyzed. Since this implementation is using a Bag-of-Words approach, each token will be an individual word. Thus, the next step in the data preprocessing is converting each review into tokenized words. The final step of preprocessing is to remove all English stop words. Stop words are terms that when removed from a sentence, do not change the meaning of the phrase. By removing these words, the number of lexical items to be processed by the algorithm is minimized, which in turn reduces processing time. Next each word is assigned its part of speech. This sentiment analyzer only considers adjectives and adverbs as relevant. Adjectives and adverbs are typically the words that convey feeling and attitude in a sentence. For this reason, all other parts of speech are ignored. In only considering adjectives and adverbs, one is able to further manage the lexicon size. The lexicon is then sorted by the frequency distribution of its words.

Once preprocessing is done and the relevant vocabulary is created, feature sets are defined. A feature set is created for each review. A feature set consists of a tuple where the first element is a dictionary of words and the second element is the sentiment assigned to the review at annotation. The dictionary of words consists of the 5000 most frequently seen words in the corpus. The dictionary represents a binary vector. Each word is a key and the corresponding value is set to True if the word appears in the review being considered, otherwise the value is set to False. An example feature set for a review would resemble the following layout: ({“good”: True, “great”: True, “bad”: False, ...}, “Positive”) Before settling on this binary vector feature, a term frequency feature set was also considered. In this style of feature set, instead of a key mapping to the words presence or lack of presence in the review, it would map to the number of times that the word appeared in the review. However, because the reviews in this data set are not very long, there wasn't much variability in the frequency of words within a review. For this reason, the binary weighting approach produced better results. The 2,000 feature sets were randomly shuffled and split into two sets -- one for training and one for testing. The models were trained on 1,500 feature sets and tested on the remaining 500.

This sentiment analyzer is implemented using several machine learning models. It uses Naive Bayes, Logistic Regression, and Support Vector Machine Classifiers to classify the data. Naive Bayes tends to perform well in document classification because it assumes that the presence of an individual feature in a document is independent of other features. This works well with a Bag-of-Words approach because it doesn’t consider context. Support Vector Machine (SVM) classification also historically has high accuracy in sentiment analysis. SVM classifiers determine the best decision boundary between vectors that belong to a given group or category and vectors that do not belong to it. As the goal of sentiment analysis is to determine what category (positive or negative) a given vector belongs to, SVM is a logical classification to consider for this task. Finally logistic regression is utilized in this scenario because the feature sets are defined as binary vectors. Logistic regression classifiers work best with binary or dichotomous variables such as the ones in the feature sets used here.

Results

The metrics chosen to evaluate the performance of each model were accuracy, f-score, precision and recall. Since the number of reviews in each category is evenly balanced (1,000 positive reviews and 1,000 negative reviews) the accuracy of each model is a fairly reliable metric. All three of the models had very similar performance levels. Each classifier categorized reviews with an accuracy percentage in the mid to low 80s. Due to the fact that the training and testing feature sets were randomized before use, each run of the program produced some slight variation in the accuracy of the models. This resulted in one model having the highest accuracy on one run of the analyzer, but not being the most accurate classifier in the following run. However after running the program multiple times, the Support Vector Machine classifier tended to result in the highest accuracy most frequently. The second most accurate model was the Logistic Regression classifier. The Naive Bayes classifier had the lowest accuracy most consistently. Figure 1.0 contains the metrics from a run of the program that reflects these accuracies.

	Accuracy	F-Score	Recall	Precision
Naive Bayes	80.4	.803	.804	.805
Logistic Regression	84.6	.845	.845	.847
Support Vector	84.8	.847	.847	.847

Figure 1.0

Naive bayes had an accuracy of 80.4% , a f-score of .803, a recall of .804, and a precision of .805. The Logistic regression classifier had an accuracy of 84.6, a f-score of .845, a recall of .845, and a precision of .847. Finally, the Support Vector Machine classifier had an accuracy of 84.8, a f-score of .847, and a recall of .847. It is not surprising that the Support Vector Machine classifier produced the best results. As previously stated, SVM classifiers determine the best decision boundary between categories. In the case of sentiment analysis the categories would be positive, negative, or neutral. The algorithm behind SVM classifiers aligns almost perfectly with the problem being solved in sentiment analysis. Generally, within a single classifier, there wasn't much difference between the accuracy, f-score, precision and recall. This indicated that the number of false positives was roughly the same as the number of false negatives. This makes sense as the input data has an equal number of positive and negative reviews.

This sentiment analysis was also performed with various sizing for the feature sets. The baseline sizing for feature sets was 5000 words from the lexicon. The effect of feature set sizing on performance was then analyzed by using an incremental approach. This was achieved by adding 1000 new word features and measuring the performance on each of the machine learning models. This was repeated to a maximum of 10,000 word features per vector/featureset. The performance of the different classifiers didn’t seem to change much as the sizing of the feature sets increased. This is not surprising as the vocabulary was ordered based on word frequency. This indicates that having a feature set of the 5,000 most frequently used adjectives and adverbs is appropriate for the scope of this set of data. This also indicates that the features that I chose to add incrementally were not very informative to the model.

Discussion and Conclusions

This project has given me a more holistic understanding of the process necessary to complete a well done sentiment analysis. It has also helped me to further understand the importance of preprocessing before training a machine learning model. It seems as if the preprocessing of data is the make or break factor behind the performance of a sentiment classifier. Additionally, I have a much better understanding of what models are most appropriate to use in the context of document classification and sentiment analysis.

This model has the potential to be improved in a number of ways. As previously stated, the bag-of-words model used here does not consider context. This disregard of context could cause some documents to be incorrectly categorized. For instance, in the Bag-of-Words model the word “bad” will likely be associated with negativity. However “bad” could actually indicate a positive connotation in certain contexts. For example, “not bad” typically means that something was satisfactory and in some instances good. In order to be able to identify these nuances, approaches like n-gram models (where n is greater than one) could be attempted in lieu of a bag-of-words. Additionally, the sentiment analysis performed here removed punctuation in preprocessing, however in some instances punctuation can be an indicator of linguistic features. Consider the sentence, “This ‘wonderful’ car turned out to be a piece of junk” (Girju 2020). In this phrase, the quotation marks indicate that while at some point the car was referred to as wonderful, that is not the current opinion of the writer/speaker. Without considering this punctuation, this phrase might be categorized as positive. For this reason, selectively maintaining certain punctuation also has the potential to improve the accuracy of the baseline sentiment analysis performed here. Another way to improve this baseline model would be to implement dependency parsing. This would help to further inform the classifier about where the sentiment in a text is coming from and at whom it is directed (Girju 2020). If I were to implement semantic analysis again in the future, I would certainly consider making some of the aforementioned modifications.

References

Amaresan, S. (n.d.). 10 of the Best Social Listening Tools to Monitor Mentions of Your Brand. Retrieved from https://blog.hubspot.com/service/social-listening-tools

Girju, R. (2020). *Ling 406: Intro to Computational Linguistics*. *LING 406: Intro to Computational Linguistics*. Urbana-Champaign .

Munir, S. (2019, March 27). Basic Binary Sentiment Analysis using NLTK. Retrieved from https://towardsdatascience.com/basic-binary-sentiment-analysis-using-nltk-c94ba17ae386

Pang, B., Lee, L., & Vaithyanathan, S. (2004). Movie Review Data. Retrieved from http://www.cs.cornell.edu/people/pabo/movie-review-data/.

Schweidel, D. A., & Moe, W. W. (2014). Listening in on Social Media: A Joint Model of Sentiment and Venue Format Choice. *Journal of Marketing Research*, *51*(4), 387–402.

Sentiment Analysis Explained. (n.d.). Retrieved from https://www.lexalytics.com/technology/sentiment-analysis

Social Media Listening. (n.d.). Retrieved from https://sproutsocial.com/features/social-media-listening/

Text Classification Using Support Vector Machines (SVM). (2018, October 4). Retrieved from https://monkeylearn.com/text-classification-support-vector-machines-svm/

Walther, C. (2019, January 2). Sentiment Analysis in Marketing: What Are You Waiting For? Retrieved from https://www.cmswire.com/digital-marketing/sentiment-analysis-in-marketing-what-are-you-waiting-for/