Sentiment Analysis of Anime Reviews: Comparing Naive Bayes Classifier to Recurrent Neural Networks

By Stephanie Tam

Abstract

Using Valence Aware Dictionary and sEntiment Reasoner (VADER) to analyze sentiment and provide a positive, neutral, or negative rating to a review after processing the text for aspects, the data is given to a Naive Bayes (NB) classifier and a Support Vector Machine (SVM) classifier for training and testing. After running the models, the data

1. Introduction

Anime is an animation style that originated in Japan, but most people use the word to refer to the Japanese animated movies or shows. And like all other movies and shows, anime shows have numerous reviews that are long and time-consuming to read. In most cases, many people just check for the reviewers' overall ratings of the anime; however, that rating is not sufficient to fully determine if the anime is worth watching because each person has his/her own preferences for plot, art, characters, and music. These are categories that comprise the overall review, but they are not always explicitly mentioned in each review.

In sentiment analysis, also known as opinion mining, the opinion of the reviewer towards something, like a product, show, topic, or service is determined via natural language processing (NLP). The opinion is given a sentiment polarity of positive, negative, or neutral; however, this can be quite difficult because written text can include sarcasm, emoticons, and idioms. In addition, sentiment analysis can be performed at three different levels: the aspect level, the sentence level, and the document level (Piryani, Gupta, & Singh, 2017). At the document level, the whole document is assigned a polarity, but moving down to the aspect level means assigning each aspect, or category, a polarity (Piryani et al., 2017). This task is challenging due to the many aspects within a text, especially when the aspects are discussed in different parts of the whole text.

Thus, there is motivation to automatically and more efficiently analyze and classify the reviews so that less time is spent physically reading and processing the hundreds of multi-paragraphed reviews. By assigning a polarity to each aspect and aggregating those polarities into an overall polarity, a better general understanding of the review—and by extension, the anime—is gained in order to determine whether or not the anime is worth watching. In addition, there is motivation to find the best model for correctly assigning sentiment polarity of the review and potentially improve that model's accuracy so that the analysis is more reliable and valid. This will thereby reduce the need to access the site that contains the reviews and verify that the potential anime show is worthwhile.

In this work, the rest of the paper is as follows: Section 2 defines the problem; Section 3 briefly summarizes the related works; Section 4 describes the proposed method; Section 5 presents the dataset and results; Section 6 provides the conclusion; and the last section contains references.

2. Problem Definition

Each anime has its own set of reviews, in which each review may contain opinions about certain aspects. Also, each aspect has various words or phrases associated with it in the text. As in Piryani et al. (2017), let be a set of reviews for each anime and be the set of aspect categories. For each in , it contains terms related to that aspect category, like "OST," or "original soundtrack," are used when discussing the music of the anime show.

The problem is to assign polarity to each aspect category for each review. The average polarity of each aspect is defined as:

$$polarity(C_k) = \frac{\sum_{j=1}^{y_k} p_{a_j}}{y_k}$$

where $p_{a_j} \varepsilon [-1, 1]$. The overall polarity of the review is then computed like the equation above, taking the

summation of the polarities of each aspect category to obtain the overall polarity of the review:

$$polarity(r_n) = \sum_{i=1}^{k} polarity(C_i)$$

And a similar calculation is used to obtain the overall rating for the given anime.

Another problem is determining the information to give to the classifier so that it can learn how to classify the reviews, and therefore, the anime. There are different ways to approach this problem: give the classifier the text and let it learn the sentiment and then rate the review, or give the classifier the review's overall rating and have it learn the sentiment to rate the review itself.

3. Related Work

In recent years, there have been several studies focused on sentiment analysis of movie reviews using machine learning. Since the focus is on anime reviews, the papers are relevant because movie reviews are quite similar to anime reviews.

Govindarajan (2013), uses a hybrid method Naive Bayes-Genetic Algorithm to classify movie reviews after preprocessing the data with stop-word elimination, stemming, and feature extraction. The results show that the hybrid method is more accurate than the NB model or Genetic Algorithm (GA), alone.

Similarly, Piryani et al. (2017) creates a list of initial aspect-related words. Unsupervised linguistic rules are then used to conduct aspect-level sentiment analysis so that an opinion summary can be generated for each aspect and the review as a whole. However, the model that Piryani et al. (2017) used fails to disregard fake opinions that are collected, meaning that the classification is potentially flawed if a significant number of opinion spams exists within the reviews.

Another proposed method for sentiment analysis of movie reviews comes from Manek, Shenoy, Mohan, and Venugopal (2017). Data preprocessing, particularly tokenization and stemming, is also done so that feature selection can be conducted. A Gini Index then used to extract the opinion-related words, which are used in the classification. NB and SVM are the classifiers used to predict the opinion.

4. Proposed Methodology

There are several classifiers that have been investigated in order to increase accuracy and overall classification. Naïve Bayes seems to be the most common classifier, so it will serve as the baseline in comparison to an SVM classifier. The data obtained from the MyAnimeList site will go through a series of filtering techniques in order to minimize potential issues with meanings, especially since the review text may contain varying sentence structures, sarcasm, idioms, and emoticons.

4.1 Data Preprocessing

As previously mentioned, in order to remove as much noise and as many irrelevant words as possible so that the models can more accurately classify the reviews, the data is preprocessed.

Tokenization splits the text into individual words, each with a value. In addition, each word is tagged with a part of speech (POS), using a POS tagger, which will retain sentence structure and later help in adjusting the values to determine polarity of an aspect. Bigrams also help retain sentence structure as words are paired, allowing for analysis of phrases. And to retain the meaning of the words and phrases, stop words and stemming are used. Stop words remove common, unnecessary words (e.g., "the," "those," "a," etc.) and stemming removes suffixes (e.g., "-ing," "-es," etc.) without muddling the meaning of the words or phrases within the review's text.

4.2 Feature Selection

Feature selection then identifies aspect terms, which are important words within the text that can be associated with aspect categories: plot, character, art, and music. The preprocessing makes attributing words, and therefore, aspect terms, to the appropriate aspect category easier. From there, each aspect term is rated and aggregated into an overall score for each aspect. The equations defined earlier in the paper indicate that the overall score of each aspect is averaged and then summed into an overall score for each review. Then, to determine the overall rating of the anime, each review's score is combined to determine whether the anime has an overall positive, neutral, or negative sentiment towards it, and therefore, is worth my time to watch it or not worth my time.

4.3 Opinion word detection

Each word has a value after being tokenized; however, VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon specifically designed for detecting sentiments/opinions, including emoticons and emojis. VADER provides positive, neutral, negative,

and compound scores. The first three scores are the aggregated ratings of positive, neutral, and negative words within each sentence of the review; the compound score provides the overall rating of each sentence within a score range of -1 to +1, with each being extremely negative and extremely positive, respectively. According to VADER's API, the typical threshold for classifying sentences as positive, neutral, and negative, respectively, are: a compound score of greater than or equal to 0.05, a compound score between 0.05 and -0.05, and a compound score equal to or less than -0.05. This threshold will be kept the same, since it will result in a more reasonable classification of each sentence, and therefore, review.

4.4 Create the model

For the NB model, Python already had preexisting NB models available in the scikit-learn package. Because this project is focused on sentiment analysis, which is related to text classification, the multinomial NB model will be used, with the review's text, the features, and its rating of positive (+1), neutral (0), or negative (-1). The intention with the model was to use the compound scores from each review and the calculation of the rating as the inputs. The NB classifier would then be trained on half of the data and tweaked as necessary to hopefully improve its accuracy. Afterwards, the best accuracy of the NB classifier would be compared to the SVM classifier.

For the SVM model, a standard SVM kernelized model that is built into scikit-learn will be used. The inputs to the model are the individual reviews' scores as X and the anime's overall classification as the Y set. It is understood that the inputs are not exactly the same as the NB classifier, due to the requirements of each model provided by scikit-learn; however, the different inputs should hopefully not interfere with comparing the results between the two models. The intention with the SVM is similar in that the goal was to train it on half of the data—the same data that the NB will be trained on—and work to improve its accuracy in whatever way possible.

5. Experimental Results

5.1 Dataset

The basic data set contains a total of 500 reviews, 200 labeled positive, 100 labeled neutral, and 200 labeled negative. The reviews, taken from MyAnimeList, were copied and pasted into text files. Half of the reviews were supposed to be used as training data, and the rest were supposed to be run during testing.

5.2 Results

Even though VADER could analyze the sentiment of the sentences in the review, the use of the information for training the classifier was extremely difficult. VADER is more intended for Tweets since they are typically shorter and use more English words, unlike an anime review. VADER failed to properly label the anime-specific and non-English words that only those in the anime community would understand as positive or negative. Most of the words were either incorrectly determined as neutral or negative, which significantly affected the overall rating of each sentence, and therefore, the review.

In addition, the provided NB and SVM models within Python's scikit-learn package and their implementations forced an attempt to organize the data into a usable format.

The required inputs for the NB model was much more difficult to navigate than originally thought. The data obtained from VADER's sentiment analysis was much harder to organize into an input that could be used to train the NB classifier, and because there was not sufficient time to modify the classifier or write my own NB classifier, I was unable to train it.

In regards to the SVM, it was easier to organize the data; however, because of the varying number of sentences and whether or not the aspects were mentioned in the review. As a result, the data could not be standardized into a clear, understandable score that could then be used in training the classifier.

5.3 Discussion

VADER made some of the NLP and sentiment analysis easier; however, it was not completely in the intended way for this project. Unfortunately, because the data obtained from VADER were not expected to be in such a difficult format to manipulate, I underestimated the difficulty in attempting to train not only one, but two, classifiers and then compare them.

Given more time, I would have focused on analyzing what VADER did not do, which was detecting the words associated with each aspect and then calculating the polarity for each. Also, as previously mentioned, I would have built my own NB and SVM classifiers so that I could feed it the data from VADER or from my own aspect-detection and sentiment analysis program.

Perhaps, I could have modified VADER in such way that it could handle anime-related terms so that the sentiment analysis would be more accurate for anime reviews and longer, more complex sentence structures.

6. Conclusions

People write reviews in many different ways. Some break their review into parts, discussing the music, the animation, the characters, and the plot. Some, on the other hand, solely write about their feelings, only really discussing the characters and the plot. Sometimes, the reviews are even as short as "One of my favorite animes of all times!" which does little to provide any comment on any of the aspects of any anime. As a result, it is much more difficult to determine the overall sentiment of an anime based on its key features.

In conclusion, there is still much work to be done to be able to use a classification model to determine whether an anime is worth watching or not.

The NLP part ended up being almost exponentially more difficult than I expected. The understanding of linguistics in human speech and applying that in processing the anime reviews ended up limiting my ability to complete the project. It was not the lack of understanding how to implement the learning algorithms, but how to apply the data obtained from NLP to the specified classifiers. As aforementioned, reviews are written in so many ways that classifying a review, and by extension, the anime, requires much more analysis. There are more parts to analyze than just specific aspects and the polarity of each word in the sentence. Unfortunately, there are limitations to this type of analysis due to idioms and sarcasm. If some people struggle to understand sarcasm, it is almost exponentially harder for a machine to be able to do so without lots of training.

As for improvements, there is much room for improvement. One thing I can see myself improving is contacting professors with expertise in this area that could have more directly advised me with this project. Another improvement I can make would be doing even more research on this topic and learning more about NLP so that I know how to better approach this project.

Even though I was unable to achieve the intended goal, I do plan on continuing this project. I personally would like to have my own software that can tell me whether an anime is worth watching, but I would get the most satisfaction from getting a deeper understanding of these learning algorithms and natural language processing. And someday, maybe I would be able to contribute more to these two areas.

References

Govindarajan, M. (2013). Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm. *International Journal of Advanced Computer Research*, *3*(13), 139-145.

Hutto, C.J., & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. Ann Arbor, MI.

Manek, A., Shenoy, P., Mohan, M., & Venugopal, K. (2017). Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. *World Wide Web, 20*(2), 135-154. doi:10.1007/s11280-015-0381-x

Piazza, A., & Davcheva, P. (2016). Sentiment Classification and Visualization of Product Review Data. In Hofmann, M., & Chisholm, A., *Text mining and visualization: Case studies using open-source tools* (pp. 133-150). Boca Raton, FL: CRC Press.

Piryani, R., Gupta, V., & Singh, V. K. (2017). Movie Prism: A novel system for aspect level sentiment profiling of movies. *Journal of Intelligent & Fuzzy Systems*, *32*(5), 3297-3311. doi:10.3233/JIFS-169272