Sentiment Analysis of Drug Reviews

Chenyuan Liu
chl063@ucsd.edu

Xiaoxian Zhang
xiz090@ucsd.edu

Abstract

Product reviews and user feedback contain huge amounts of information that reflect the features, advantages and disadvantages of products through user experience or sentiment polarity. This information can be leveraged through data mining approaches such as sentiment analysis. In this article, we focus on the pharmaceutical field, using sentiment analysis to analyze user sentiment polarity to drive valuable insights. These analysis not only help users make decisions, but also help healthcare experts make improvements to specific drugs.

In this work we perform sentiment analysis to predict the sentiments of user reviews on specific drugs.

1 Dataset

We used a data set (Gräßer et al., 2018) in the UCI Machine learning repository. The data set was scraped from Drugs.com for retrieval of user reviews and ratings on drug experience. Drugs.com is, according to the provider, the largest and most widely visited pharmaceutical information website providing information for both consumers and healthcare professionals. It provides 215,063 user reviews on specific drugs along with related condition and a 10-star user rating reflecting overall user satisfaction.

Furthermore, we derived three-level polarity labels for overall patient satisfaction using thresholds as specified in Table 1. The data set was further split into training and test partitions according to a stratified random sampling scheme with the proportion of 75% and 25%, respectively. The total number of individual drugs in the data amounts to 6345 and the average number of reviews per drug is 58.86. The average length of each review is 458.73 characters and review length categorized by sentiment type follows nearly very similar distribution as shown in Figure 1.

Rating	Sentiment	proportion
rating < 4	-1	22%
$4 \leq$ rating < 7	0	12%
rating ≥ 7	1	66%

Table 1: Labels

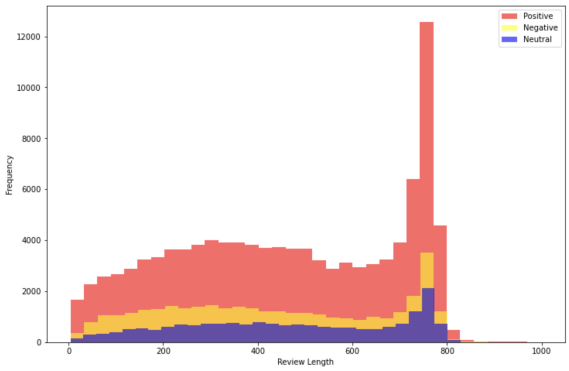

Figure 1: Review length for different sentiment types

We also generated word clouds for positive and negative sentiments to explore whether keywords in these two categories have big difference. However, it was interesting to discover that many keywords in both word clouds overlap. The biggest concern for medications is side effect. We also noticed that there are negative words such as bad, pain, mood swing that only occur in negative reviews that can help us classify.

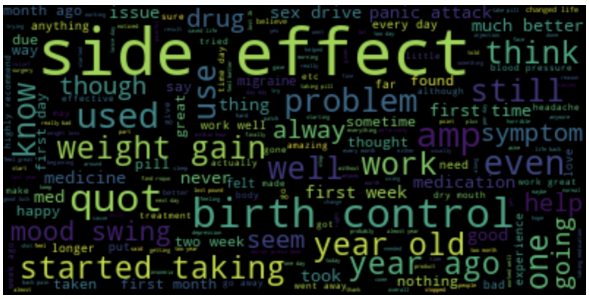

Figure 2: Word Cloud of Positive Sentiment

Figure 3: Word Cloud of Negative Sentiment

2 Literature Study

Sentiment analysis is aimed to determine whether a set of text expresses a positive or negative attitude or opinion. The task of sentiment analysis is to obtain the polarity from the observed text. Sentiment analysis can be categorized into three categories: detection of the polarity of words, sentences and documents. In our task we are dealing with the polarity of sentences.

For sentiment analysis, most researchers have worked on general domains (such as news, products, movies, and restaurants), but not extensively on health and medical domains. Text in this domain contains many medical terminologies, so we are not able to apply those popular pre-trained models in general domain.

Felix (Gräßer et al., 2018) studied the same data set to analyze the sentiments concerning overall satisfaction, side effects and effectiveness of user reviews on drugs. This work showed that transfer learning approaches can be used to exploit similarities across domains and is a promising approach for cross-domain sentiment analysis.

Yun Niu (Niu et al., 2005) applied supervised learning method to analyze the possibilities to medical text. The predicted possibilities are divided into four categories: positive outcome, negative outcome, no outcome and neutral outcome. No outcome is defined as no RCTs comparing combined pharmacotherapy and psychotherapy with either treatment alone. They discovered that the combination of linguistic features and domain knowledge leads to the highest accuracy.

Xia Peng(Xia et al., 2009) developed a system to classify new posts on a forum according to their topic and polarity. The forum is called "Patient Opinion", which is an online review service for users of the British National Health Service. Then the users can add reviews on food, staff, treatments, and so on. Both topic and polarity identification were achieved through a rather simple machine learning approach using BOW.

In another research on drug review sentiment analysis, (Na and Kyaing, 2015) adopted pure linguistic approach of computing the sentiment orientation (polarity) of a clause from the prior sentiment scores assigned to words, taking into consideration the grammatical relations and semantic annotation (such as disorder terms) of words in the clause. Experiment results with 2,700 clauses show the effectiveness of the proposed approach, and it performed significantly better than the baseline approaches using a machine learning approach.

(Yadav et al., 2018) did medical sentiment analysis using social media in several popular domains such as depression, anxiety, asthma, and allergy. The focus is given on the identification of multiple forms of medical sentiments which can be inferred from users' medical condition, treatment, and medication. Thereafter, a deep Convolutional Neural Network (CNN) based medical sentiment analysis system is developed for the purpose of evaluation.

3 Approaches

The dataset contains several columns and our interests mostly lie on the user reviews, and since we are going to use different text mining approaches, it is necessary to adjust the data into a format that is suitable to fit in different models. Here we used a sequence of approaches to handle the review data and generated a new data set with irrelevant information dropped and processed review information.

3.1 Missing values

Since the data set was retrieved from the website and it is possible that there are flaws and missing values. Since missing values are presented as NaN in the data set we decided to drop all the missing values to clean the data set.

3.2 Text Preprocessing

To begin with, all reviews were preprocessed according to a standard scheme: punctuation, special characters and stopwords were removed and alphabetic characters were transferred to lowercase. After that, lemmatization was applied to convert words to the root format.

3.3 Text Vectorization

Subsequently, the preprocessed documents were tokenized to obtain the overall vocabulary. Feature

representations of each review were generated using different approaches: count vectorizer, TF-IDF vectorizer, Word2Vec.

3.4 Model Training

Using the extracted feature representations, Logistic Regression, Naïve Bayes Classifier and linear SVM Classifier were employed for the building of sentiment classification models. Besides, LSTM and BERT were used as a combination of text representation and classifier.

3.5 Model Evaluation

Our goal in this task is to build a model that can analyze users feelings based on their drug reviews. Thus we used accuracy score and marco F1-score on the test set to evaluate the performance of our model.

Accuracy score represents the percentage of model's predictions that are correct. The score tells us how well the model predicts, however, accuracy score cannot represent a fair metric when a data set is unbalanced because it fails to evaluate the minority group. As shown in Table 1, sentiment labels are considerably unbalanced in this case.

So we also used macro-F1 score as another evaluation metric to alleviate disproportionate distribution. The macro-F1 score is computed by taking the arithmetic mean of all the per-class F1 scores. This method treats all classes equally regardless of their support values.

To avoid overfitting problem, model hyperparameters were tuned using a 5-fold Cross Validation Grid Search on the training data, using accuracy as scoring metric.

4 Experiments

4.1 Count Vectorizer

This text representation is a bag-of-words(BoW) method, which describes the occurrence of words within a document. It just uses counts and disregards the grammatical details and word orders. We firstly created a vocabulary indices of words from the entire set of documents. Then we constructed the numerical feature vector for each document that represents how frequent each word appears in different documents. The feature vector representing each document had 45896 dimensions, which would make training process extremely slow. However, the vectors were sparse in nature as the words in each document will represent only a small subset of words that are present in the entire corpus, so we transformed feature vectors to sparse matrix to increase computing efficiency.

Features	Models	Acc. / F1
Unigrams	LR	0.81/0.68
Unigrams	NB	0.73/0.59
Unigrams	SVM	0.78/0.57
Unigrams+Bigrams	LR	0.91/0.85
Unigrams+Bigrams	NB	0.84/0.72
Unigrams+Bigrams	SVM	0.89/0.82
Unigrams+Bigrams+Trigrams	LR	0.91/0.86
Unigrams+Bigrams+Trigrams	NB	0.87/0.80
Unigrams+Bigrams+Trigrams	SVM	0.91/0.85

Table 2: Count Vectorizer

Features	Models	Acc. / F1
Unigrams	LR	0.80/0.67
Unigrams	NB	0.69/0.36
Unigrams	SVM	0.76/0.49
Unigrams+Bigrams	LR	0.83/0.68
Unigrams+Bigrams	NB	0.66/0.27
Unigrams+Bigrams	SVM	0.73/0.44
Unigrams+Bigrams+Trigrams	LR	0.84/0.69
Unigrams+Bigrams+Trigrams	NB	0.66/0.27
Unigrams+Bigrams+Trigrams	SVM	0.69/0.36

Table 3: TF-IDF Vectorizer

In the experiments, we tuned minimum token frequency threshold and n-gram number of adjacent tokens. Both, single tokens, e.g. words, (unigrams) as well as two or more adjacent tokens (bigrams, trigrams), e.g. 2- or 3-word expressions, were used to derive features. It turned out that adding up to trigrams worked the best, as shown in Table 2.

We trained on 3 different machine learning models that are popular in NLP domain: Logistirc Regression, Multinomial Naïve Bayes Classifier and linear Support Vector Machine (SVM) Classifier. We fine-tuned penalty type and regularization strength for all these models. In addition, different learning rates were tried in SVM classifier for Stochastic Gradient Descent (SGD) learning. As shown in Table 2, for each feature representation, Logistic Regression always performs best, and SVM comes the second. The result is reasonable as Naïve Bayes Classifier is relatively simple as it only uses Bayes Theorem to compute the posterior probability for each class. SVM is more complex as it develops hyperplane equation which separates points into classes. While SVM tries to find the

"largest" margin (distance between the line and the support vectors) that separates the classes, Logistic Regression is not used to find the best margin. Instead, it can have different decision boundaries with different weights that are near the optimal point.

4.2 TF-IDF Vectorizer

TF-IDF vectorizer combines Term Frequency and Inverse Document Frequency to generate numerical feature vectors. Words that occur more frequently in one document and less frequently in other documents are given more importance. The process of hyperparameter tuning is similar to count vectorizer. The result in Table 3 illustatesd that the Logistic Regression model that were trained using unigrams, bigrams and trigrams had the best performance.

Compared with models trained using count vectorizer, models trained on TF-IDF vectorizer gave relatively lower accuracy and considerably worse F1-score. The abnormal F1-scores of Naïve Bayes Classifier and linear SVM Classifier were because these two classifiers predicted no 0 (neutral) labels in certain cases, setting the precision and F1-score of that category to zero.

4.3 Word2Vec

We firstly built a vocabulary ordered by term frequency by descending order for the corpus, and trained the corpus using Word2Vec. We utilized two different algorithms to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram. In the CBOW architecture, the model predicts the current word from a window of surrounding context words. In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. For the hyperparameter tuning phase, we optimized vector dimensionality size, window size and minimum word frequency threshold.

Logistic Regression achieved accuracy of 0.76 and F1-score of 0.50; Performance of SVM was slightly lower, with accuracy of 0.74 and F1-score of 0.47. Naïve Bayes Classifier is not applicable as it only accepts non-negative numerical values.

4.4 LSTM

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) capable of learning order dependence in sequence prediction problems. Unlike standard feedforward neural networks, LSTM has feedback connections. An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. Each block contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates – that provide continuous analogues of write, read and reset operations for the cells.

We did additional text preprocessing steps: encoding the labels, tokenizing and converting text to numerical vectors, padding or truncating vectors to a fixed length. Then we built our model. First, we initialized the sequence model. Then, we added a word embedding layer, a LSTM layer and a Dense layer, which is a fully connected neural network. Dropout layers were added between layers and also on the LSTM layer to avoid overfitting. We used Categorical Cross Entropy Loss and Adam optimizer to train the model.

We also tried to use 2-layer stacked LSTM model and bi-directional LSTM model. The bi-directional one uses bi-directional recurrent neural nets to present each training sequence forwards and backwards to two separate recurrent nets, both of which are connected to the same output layer. Training time for these models increased exponentially but there was little performance improvement.

4.5 BERT

Besides using various supervised machine learning model, we also applied BERT, an unsupervised model to finish the task. BERT (Devlin et al., 2018) model is a model derived from transformer. The transformer used a mechanism called attention, it not only consider the original word but also other words associated with it when translating. This can make the result more accurate by having more feature information when translating words. BERT is a bi-directional encoder representation based on transformer, it takes into account the preceding and following words to get its meaning in the context when processing a word. In order to achieve the mentioned method, BERT uses masking, the model will randomly select 15% of the words in the corpus, and then 80% of them will use the mask to replace the original words, 10% of them will be randomly replaced with another word, and the remaining 10% will keep the original words unchanged,

The model is then asked to correctly predict the selected word. The reason for these processing is that mask will only appear during training. If the selected words are changed to mask, this will cause the trained model to target mask, which is inconsistent with the actual situation. There will be some probability to randomly replace mask, and this probability is 10%*15%=1.5%, which will basically not harm the text understanding ability of the model, and the model needs to maintain the distribution of each input token in order to predict correctly Context representation, so as to achieve the purpose of getting the preceding and following words feature information.

Since BERT model has a high requirement on the hardware and due to the scale of our dataset and computation capability of our device, we have to use 20% of the whole dataset to train the model. In our work, we implement the BERT model with the pre-trained model as **bert-base-uncased** and 2e-5 as the learning rate. The accuracy reached 0.81.

5 Conclusions

Table 4 showed the result of all of our models, among all combinations of text representations and machine learning models, word vectorizer combined with Logistic Regression model became the most effective method for this task. This combination gave us an accuracy with 0.91 and a macro F1-score with 0.86. Extensive experiments were conducted with different machine learning based classifiers, *e.g.* Naïve Bayes Classifier and linear SVM Classifier, and our conclusion held .

This is an interesting finding that as a basic text representation approach, word vectorizer performed unexpected well among all models we implemented. It is anti-intuitive as the bag-of-words model would lose the relative positional information of sequential data. The Logistic Regression model is also simple and straight forward, whose scale of parameters is way smaller than the SOTA neural network based NLP methods. After careful review, we verified that this result had nothing to do with unexpected implementation error. Thus, we believed that it was an interesting and important example of regularization effect and tradeoff between the potential of model structure and available resources (computational and storage resources, large dataset with high quality, etc).

In our experimental settings, the Naïve Bayes Classifier served as the baseline model and had

Features	**Acc. / F1**
Word Vectorizer	0.91/0.86
TF-IDF Vectorizer	0.84/0.69
Word2Vec	0.76/0.50
LSTM	0.78/0.62
BERT	0.81/0.68

Table 4: Model Comparison

the worst performance in all the conducted experiments as expected. For the SVM model, the vectorized document with high dimensions can be hard to process, as the metric of 'distance' would become invalid in high-dimensional spaces, *e.g.*, the $\mathbb{R}^{700}$ and $\mathbb{R}^{45896}$ used in our Word2Vec and Word Vectorizer models respectively. Excessive training epoches and weak regularization constraints may also result in overfitting. However, in our experiments, it still achieved results comparable to the Logistic Regression model and we believe it is due to the sparse nature of the Word Vectorizes applied.

The traditional BERT or LSTM model for NLP related tasks were expected to reach high performance, however, it turned out that the performance is very poor. We believed that this is due to multiple reasons. Firstly, our training data was limited. Although it was a dataset in vertical field, its scale was way smaller than the datasets of NLP conferences. Our computational power was limited as well. We applied the pre-trained model of BERT to accelerate the training but we still need to cut the maximum length of input tokens in order to fit the model into our memory. The same thing happened again when training the LSTM models and we believed that such truncation together with the information loss would acutely decrease the model's accuracy.

At the same time, the simple models were easy to train, capable of capturing the crucial features quickly and free from the truncation related information loss.

Since we have a large scale of dataset, our hardware cannot hold so much data and model in the memory so we had to use part of the data to train the model. In addition, we also had to limit the length due to the hardware limitation. Despite all compromises mentioned above, BERT still took extremely long time to train.

As for the further improvement of our study, we believe that there are huge potential for BERT model to improve on our dataset, for example, we

can increase the BERT tokens length and since our task focuses on medication scope, we could use in-domain pre-training to change the distribution of data.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health*, DH '18, page 121–125, New York, NY, USA. Association for Computing Machinery.

Jin-Cheon Na and Wai Yan Min Kyaing. 2015. Sentiment analysis of user-generated content on drug review websites. *Journal of Information Science Theory and Practice*, 3(1):6–23.

Yun Niu, Xiaodan Zhu, Jianhua Li, and Graeme Hirst. 2005. Analysis of polarity information in medical text. In *AMIA annual symposium proceedings*, volume 2005, page 570. American Medical Informatics Association.

Lei Xia, Anna Lisa Gentile, James Munro, and Jose P. Iria. 2009. Improving patient opinion mining through multi-step classification. In *TSD*.

Shweta Yadav, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. 2018. Medical sentiment analysis using social media: towards building a patient assisted system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.