12-4-2022

Artificial Intelligence in the Medical Field: Medical Review Sentiment Analysis

Nicholas Podlesak
Northern Illinois University, z1885689@students.niu.edu

NORTHERN ILLINOIS UNIVERSITY

Artificial Intelligence in the Medical Field: Medical Review Sentiment Analysis

A Capstone Submitted to the

University Honors Program

In Partial Fulfillment of the

Requirements of the Baccalaureate Degree

With Honors

Department Of

Computer Science

By

Nicholas Podlesak

DeKalb, Illinois

May 2023

Abstract

In this research project, natural language processing techniques' ability to accurately classify medical text was measured to reinforce the relevance of artificial intelligence in the medical field. Sentiment analyses (analyses to determine whether the text was positive or negative) were performed on the prescription drug reviews in an open-source dataset using four different models: lexical, a neural network, a support vector machine, and a logistic regression model. Each model's effectiveness was gauged by its ability to correctly classify unlabeled drug reviews (i.e., a percentage representing accuracy). The machine learning models were able to accurately classify the text, while the lexical model could not reliably produce an accurate prediction. The significance of the preprocessing technique known as 'stemming' was also analyzed in this project as well. Stemming made a negligible difference in accuracies (<1%).

1 Introduction

The medical field has mass amounts of data lying dormant, needing to be analyzed. When new medicine is prescribed to people all over the world, how will doctors accurately gauge the medicine's effectiveness? Artificial Intelligence can be used to analyze patients' response's to certain drugs, allowing for doctors to note trends and make decisions based off those trends accordingly.

Natural language processing is the branch of artificial intelligence concerned with breaking down human language into something that can be understood by a computer. When working on any NLP project, it is important to have a sufficient amount of data to analyze for better and more accurate results. The dataset used in this project was pulled from UC-Irvine's open-source machine learning repository (UCI Machine Learning Repository, n.d.). It was used in a similar research experiment, in which this research design is based off of, conducted by

BMC Medical Research Methodology (Harrison and Sidey-Gibbons, 2021). The dataset consists entirely of prescription drug reviews scraped from various medical review websites. It was specifically chosen for its contents (medical text), as it aligns with the goals of the project.

In this project, natural language processing techniques were used to perform sentiment analyses on the prescription drug reviews in the dataset. Each review in the data set has a certain sentiment attached it; positive or negative. Supervised machine learning models were trained to try and accurately predict those sentiments solely based off the text in the review alone. A simpler type of analysis was performed as well, which uses simple mathematics with the help of a lexicon to predict the overall sentiment of text. This type of analysis is considered lexicon based.

A lexicon refers to a list, typically created by linguistics experts, of words matched to a sentiment. For example, the word “great” likely is assigned a positive sentiment, and the word “hate” would likely be assigned a negative sentiment. A lexicon can be used to assign sentiments to individual words in text, providing insight on the overall sentiment of the text.

Supervised machine learning models use sophisticated mathematical algorithms to try and map an input to an associated label (in this case, the label is a sentiment of positive or negative). Therefore, such models must be trained with labeled data to first to ‘learn’ the association between the text and its label. Then, the model can be used to predict the label of standalone text. Three supervised machine learning models were used in this project: a neural network, a logistic regression model, and a support vector machine. Accuracy across the three models was analyzed.

2 Methods

2.1 Setup

The dataset was first downloaded from UC-Irvine's open-source repository, then stored locally ready for use. Using the Python library Pandas, the dataset was imported and stored in a data frame; a matrix-like data structure used in many statistical packages. The original dataset consists of over 200,000 prescription drug reviews, so to reduce computational burden and hardware demand, only the four most reviewed drugs in the original dataset were analyzed: etonogestrel, nexplanon, levonorgestrel, and ethinylestradiol. All other rows in the data frame were dropped, leaving a total of 11,999 reviews left to be analyzed. The entirety of the project was conducted using Python and relevant libraries.

2.1 Data preprocessing

It is standard in natural language processing to try and reduce any variations and inconsistencies in text before feeding it to machine learning models. This will prevent the same words that are represented by different characters from being interpreted differently. For example, "HATE", "Hate!", and "hate.", all have very similar meanings (if not the same) yet would be interpreted differently if left as is. The text needs to be standardized to minimize situations as such.

Using the Contractions library in Python, all contractions in each review were expanded. Words like "Shouldn't" and "Don't" were expanded to "Should not" and "Do not". All words were then converted to lowercase; "Should" to "should", "Do" to "do". Afterwards, a regular expression was used to match and replace all non-alphabetic characters with a single space. Next, all 'stopwords' – words like "I" or "the" that provide no insight to the overall sentiment of the review were removed from the text. Finally, using the Porter-stemming algorithm, all words in each review were reduced to their common stem. For example, "watching" and "watched" would

be reduced to “watch”. The stemmed reviews were stored in a new column in the data frame so that they could be separately analyzed by the models to measure the effectiveness of using this technique.

Each review in the dataset came with an associated star rating, from 1 to 10, provided by the reviewer. Another column was added to the data frame consisting of a binary evaluation of the star rating. Reviews with star ratings over 5.0 were considered positive and were assigned a 1 in the new column. All other reviews were considered negative and were assigned a 0. These binary ratings represent the label for a given review (1 for positive, 0 for negative), used to help train the machine learning models.

2.2 Sentiment Analysis: Lexicon Based

Using the Python library VADER, each review was assigned a ‘negative’, a ‘neutral’, and a ‘positive’ polarity score. These scores are essentially just percentages, and were calculated by the proportion of negative, neutral, and positive words in the review. For example, suppose the cleaned review for a random drug in the dataset be “great.” This review consists of one word, and that word carries a positive sentiment. Its positive polarity score would be 1.0, because it contains 100% positive sentiments. The negative and neutral polarity scores would both be 0.0. These three scores are then normalized to a value between -1.0 and 1.0 inclusive, denoted as the compound score. The more positive the compound score, the more positive the overall sentiment of the review is. As such, reviews with positive compound scores were considered positive reviews, and assigned a 1. Reviews with negative compound scores were considered negative and assigned a 0. These predicted sentiment scores of either 1 or 0 were compared to the actual binary labels associated with the reviews to calculate the percentage of accurate predictions. This

same process was completed with the stemmed reviews as well so that any notable difference in accuracy could be analyzed.

2.3 Sentiment Analysis: Supervised Machine Learning

2.31 Support Vector Machine

The data frame containing the reviews was split into two separate sets: the training set, consisting of 75% of the total reviews in the original data frame, and the testing set, consisting of 25% of the total reviews in the original data frame.

Although the general data preprocessing was sufficient to perform the lexicon-based sentiment analysis, more specialized preprocessing is required before any of the machine learning models can be trained. Like all mathematical models, machine learning models can only take raw numbers as input for any sort of computation to be possible. The reviews need to be vectorized – transformed from one large string of words to a vector of numbers, where each individual element in the vector represents a word in the review. To do so, Sci-kit Learn's TF-IDF vectorizer algorithm was implemented. TF-IDF (term frequency – inverse document frequency) is a statistical algorithm used to calculate the overall relevance of a given word in a given corpus.

After the data was vectorized and split into training and testing sets, it was ready to be analyzed using the model. The support vector machine was implemented using Sci-kit Learn's linear support vector classifier. The model was trained using the training data set, then tested using the unlabeled, test data set. This process was repeated with the set of stemmed reviews so that any notable differences in accuracy could be analyzed.

2.32 Logistic Regression Model

Like the additional preprocessing procedure for the support vector machine, the data underwent a 75-25 train-test split, then was vectorized using Sci-kit Learn's TF-IDF vectorizer. Next, Sci-kit Learn's logistic regression model was imported and built, then trained with the labeled train data set. After training, it was used to predict the sentiments of the unlabeled test data. This process was repeated with the set of stemmed reviews so that any notable differences in accuracy could be analyzed.

2.33 Neural Network

The additional preprocessing conducted before training the neural network differed slightly from the support vector machine and the logistic regression model. An 80-10-10 split was conducted to separate the data into three different sets respectively: the training set, the validation set, and the testing set. The validation set is similar to the testing set; however, it is used during the testing process so that the model can make optimizations while testing for more accurate results. Rather than using the TF-IDF algorithm to vectorize the text, the text was vectorized using the text vectorization layer imported from TensorFlow's neural network API, Keras. The text vectorization layer takes a set of strings as input and transforms it into a set of vectors consisting of word-indices.

Keras was also used to construct and configure the neural network used to classify the reviews. The neural network was configured with five layers, each serving a separate purpose. The first layer in the neural network is the input layer. The input layer takes the preprocessed data object as input and transforms it into a "tf.Tensor" object with a compatible shape to be sent through the following layers in the network. The second layer in the network was the word embeddings layer. This layer essentially tries to learn the connections between the current meaningless integer indices in the vectors to encode them in a similar manner. For example,

suppose the word “hot” is encoded as 1, and the word “warm” is encoded as 2097 initially. This is an instance where two words similar in meaning have vastly differing numerical encodings. The goal of the embedding layer is to transform their initial encodings into values that are similar (typically floating-point values) so that their impact on a mathematical model is similar as well. Ideally, this would produce much more meaningful vectors that capture the relationship between words, resulting in more accurate predictions. The third layer in the model is the flatten layer, which simply transforms the multidimensional input into a single dimension, ready to be passed through the final two layers: the dense layers. These layers are ultimately what calculate the predicted classification for the given input. They compute the weighted average of an input, then pass that average to an activation function. The first dense layer was configured with the ‘relu’ activation function, while the second was configured with the ‘sigmoid’ activation function.

After the neural network was successfully constructed and configured, it was trained using the train and validation data sets, then tested using the test dataset. Like the other models, the neural network was trained and tested with stemmed reviews as well so that any notable difference in accuracy could be analyzed.

3 Results

Model Prediction Accuracy

Model	Accuracy (%)	Stemmed Accuracy (%)	Average Accuracy (%)
Neural Network	85.83	82.09	83.96
Logistic Regression	82.64	82.88	82.76
SVM	82.47	83.27	82.87
Lexical	60.77	60.10	60.44

Figure 1

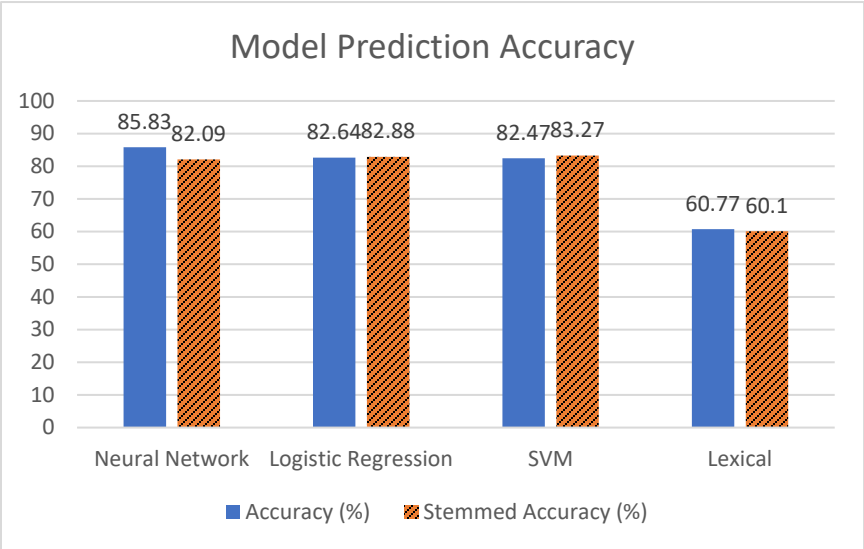

Figure 2

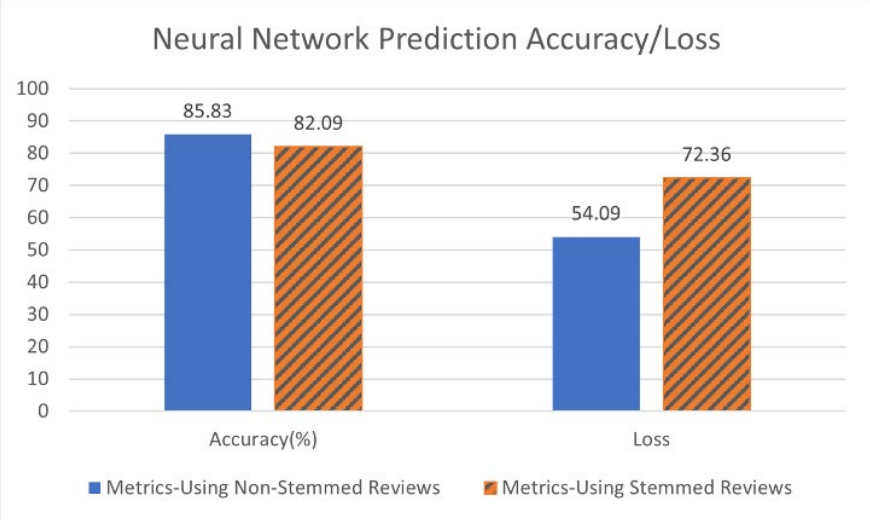

Figure 3

The ‘accuracy’ metric (in all Figures) is simply the percentage of correct classifications that the model predicted. The ‘loss’ metric (in Figure 3) is the summation of the errors that the neural network made while classifying. Figure 2 is included strictly to help visualize the data seen in the Model Prediction Accuracy Table.

The neural network scored the highest, accurately classifying 83.96% of reviews on average, while the lexical analysis scored the lowest, accurately classifying only 60.44% of reviews on average. The logistic regression model and support vector machine accurately classified 82.76% and 82.87% respectively.

The neural network and lexical analysis more accurately classified the non-stemmed reviews, while the logistic regression model and support vector machine more accurately classified the stemmed reviews. The overall average prediction accuracy on the stemmed reviews across all models was 77.08%, and the average prediction accuracy on the non-stemmed reviews was 77.93%. The non-stemmed reviews were more accurately classified by 0.85% on average.

4 Discussion

With accuracy percentages all in the 80's, the three supervised machine learning models can effectively and consistently classify the prescription drug reviews. However, the lexical approach was unreliable, with an average accuracy of only 60.44%. The neural network was the most accurate model, correctly classifying 83.96% of the test reviews on average. Contrary to my original expectations but in line with the results of previous research, stemming didn't improve classification accuracy. In fact, stemming resulted less accurate classifications overall in this project.

It is important to note some of the limitations in this project. The dataset was chosen strictly because it contains medical related text, which was to be packed full of medical terminology. However, that wasn't exactly the case. Reading through the reviews in the set, medical terminology wasn't sparse by any means, but it wasn't overly abundant. The goal of this research project was to prove that artificial intelligence can be used to accurately classify medical text. Although that was accomplished, as prescription drug reviews are considered

medical text, it could be useful to conduct further research using medical text that is much heavier on the medical terminology.

It is also important clarify that the sentiments in the lexicon used in the VADER library were specifically tuned with consideration to a words sentiment when used in social media. Essentially, a word used on a social media platform could carry a different sentiment compared to when it is used in a general setting, and the VADER lexicon is tuned to handle such cases. This could have impacted the results for the lexicon-based sentiment analysis, so it would be insightful to perform a lexicon-based sentiment analysis using a more generalized lexicon.

5 Conclusion

Much like non-medical text, the results indicate that medical text can be accurately classified using natural language processing techniques as well. This project was able to effectively reinforce the relevance of artificial intelligence in the medical field, and further support the claim that the preprocessing technique ‘stemming’ provides little to no additional benefit. It has also opened the door for further exploration of artificial intelligence, more specifically, natural language processing, in healthcare.

References

Harrison, C.J., Sidey-Gibbons, C.J. Machine learning in medicine: a practical introduction to natural language processing. *BMC Medical Research Methodology 21*, 158 (2021). https://doi.org/10.1186/s12874-021-01347-1

UCI Machine Learning Repository: Drug Review Dataset (drugs.com) data set. (n.d.). Retrieved September 1, 2022, from https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29