Sentiment Analysis With KNN Algorithm

Hyunwoo Max Cho
Shanghai American School

Abstract:

Today, a lot of information is being poured out on the Internet. However, the ability to analyze and process information has become more important as there is so much information. Sentiment analysis is a technique that allows you to analyze a given text to understand the sentiment of the person who wrote it. [2] This is used in a wide variety of fields, ranging from finding out whether or not people who purchased goods are satisfied, to finding people who are at risk of suicide in advance or summarizing the main contents of the book. In the past, people conducted a sentiment analysis by scoring how positive each word is. However, this has a huge constraint on handling huge amounts of information in real time. Nowadays we combine machine learning with natural language processing (NLP) for this purpose. In this study, the author will use the KNN algorithm to identify how close the words are and analyze the sentiments of the people who wrote the movie review data.

1 Introduction

Design of the Research

Sentiment analysis is one of the most useful technologies that is used in many companies and research institutes of today. For instance, if a customer buys an item on Amazon and leaves a review, we can analyze keywords in that review to examine problems and appealing aspects of the product. Another example is the 2016 U.S. election. Many public opinion polls have predicted Donald Trump will lose the election, however, the experts foresaw that Donald Trump will win the election since they were able to see peoples' dissatisfaction for current government policies using results from the sentiment analysis in social networks such as Twitter or Facebook. Consequently, sentiment analysis is being used as an important tool for swiftly and concisely processing the vast data on the internet nowadays.

However, the problem of the existing sentiment analysis method is that it needs preprocessing process of scoring values to numerous words. In other words, we have to quantify how positive or negative the word Love is compared to the word Sad. Hence, the researchers had to categorize a vast number of words before conducting sentiment analysis which can be a tedious process. In this study, the author has examined if it is possible to conduct sentiment analysis without this preprocessing using the KNN algorithm to link uncategorized words that are related. To examine this, the author has analyzed 50,000 movie reviews dataset from Stanford using the machine learning algorithm to obtain the result. [2]

Design of the Paper

In section two, background knowledge that was used for this research such as machine learning, KNN algorithm, and NLP will be established. Section three will examine the preprocessing

process of the movie review dataset. In section four, the author will conduct the experiment based on our algorithm to measure how accurately it analyzes the sentiments. Lastly, the conclusion part will discuss the conclusion and references.

2. Background Knowledge

2.1. Machine Learning

Machine Learning is a way for a program to execute the given task on its own without predetermined rules, rather than a programmer explaining certain tasks to a computer. In the past, artificial intelligence had to be constructed manually by the experts in the field which required programming a vast amount of rules directly. To develop the supercomputer Deep Blue by IBM, for example, not only programmers, but also top chess players at the time were needed for the manual process of implementing numerous rules and exceptions to win the world chess champion. Hence, such methods require enormous effort and cost since it requires a lot of manpower. Besides, there are problems in a certain area that even people cannot disentangle or require too many rules to be manually implemented. Machine learning is a method that is designed to be used in these cases.

Machine learning began with the conception of the 'learning process' when the human brain processes information, so that the computer can discover new rules on its own without help from humans. Mainly, there are two major stages in machine learning. First is the training phase where the model learns features and patterns from the given data. In this process, the computer gradually modifies its model by looking at countless data and the correct results. The following second stage is the prediction phase where the usefulness of the machine learning model is determined. In this phase, the machine learning model classifies given data by itself based on the previous learning, and the performance of the model is determined by the accuracy of its result.

The interior of machine learning consists of artificial neurons that resemble human neurons. Just like neurons in the human brain produce the appropriate electric signal after processing the signal, artificial neurons in machine learning produces output by applying a certain function to a given input. There are several parameters in these functions and the output of the machine learning model varies depending on the value of these numbers. Therefore, to make the machine learning model that performs well, it is important to find the right value for these parameters from training. [1]

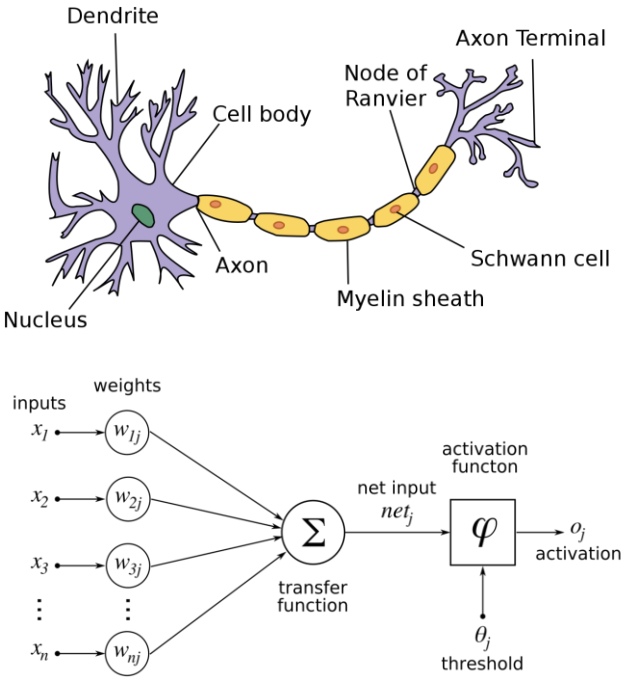

Figure 1: Comparison between a neuron and an artificial neuron

2.2. KNN Algorithm

Perhaps the saying that best describes the KNN algorithm is "Birds of a feather flock together." As this saying goes, the KNN algorithm is a method for classifying the data by looking at a few closest neighbors when new data is given.

For example, let's say we want to find out if '?' in the figure below is a triangle or a circle. First, the KNN algorithm examines k number of the nearest neighbors and which category it belongs. There are two figures in Figure 2; the inner circle is when k=1 and the outer circle is when k=4. To prevent a tie, the value of k is usually set to an odd number.

After setting up the range of the circle, the algorithm measures the number of circles and triangles within this circle and determines the value of '?' by the majority class. In this case, we have to be careful since the result may vary depending on the value of k. For instance, in the figure below, '?' is classified as a circle if k=1, however, it is classified as a triangle if k=4. The appropriate k-value depends on the characteristics of the given data, therefore, it requires fine tuning.

Although the KNN algorithm is a simple algorithm, its' application is expanded into many fields such as image processing, face and character recognition in the video, personal preference prediction of movie, music, or product, medical, and pattern recognition of genetic data. [3]

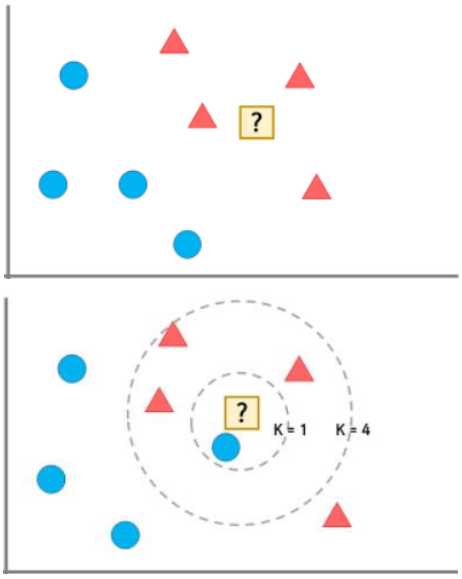

Figure 2: Classifying with KNN algorithm example

2.3. Natural language processing

Natural language is the language that we use in our daily lives. Natural language processing is the task of analyzing the meaning of these natural languages so that computers can handle them.

Natural language processing is a field used in places such as voice recognition, content summary, translation, sentiment analysis, text classification tasks (spam classification, news article category classification), question answering systems, and chatbots.

Artificial intelligence is emerging as an important keyword in the IT field as deep learning has recently received attention. Natural language processing is the most important field of research in artificial intelligence in that it makes machines understand human language, but there are still many mountains to be conquered.

Usually, NLP involves the following three processes.

1. **Text Preprocessing:**

Text preprocessing includes preprocessing processes such as capitalizing/uncapitalizing, removing special characters, removing emojis,

and text normalization processes such as word tokenization, stop words removal, and Stemming/Lemmatization.

2. **Feature Vectorization:**
Feature vectorization is the process to extract features from preprocessed text and converting the features to vectors. Typically, BOW(Bag of words) or Word2Vec method is used.

3. **Machine learning modeling:**
This is the process of constructing a machine learning model for the created dataset and training/predicting the model.

3 Data Processing

Text preprocessing is a very important task in natural language processing. Text preprocessing is the process of preprocessing text for a specific use. If you don't prepare the ingredients properly, you can't cook properly. Likewise, without proper preprocessing of text, natural language processing techniques will not work properly. In this section, we adapted various techniques for preprocessing text.

3.1. Data Load

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
from tensorflow.keras.datasets import imdb
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GRU, Embedding
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.models import load_model

train_df = pd.read_csv("train_data.txt")
test_df = pd.read_csv("test_data.txt")
```

First, we imported the libraries needed for the program and loaded the data that will be used for training. The 50,000 movie review comments dataset from Stanford was used for the training. [5] csv stands for comma separated values, which means a file format in which values are divided by commas.

3.2. Data Tokenize

Next, we need to split sentences into words. This is process is called Tokenization. In this experiment, we tokenized the sentence based on white space. In addition, the cleaning process which removes special characters such as period and comma, was performed at the same time.

3.3. Cleaning and Normalization

```
def cleaning_text(text):
        soup = BeautifulSoup(text, "html.parser")
    text = soup.get_text()
    return text
```

The cleaning and normalization process is required for the machine learning model to analyze the text properly. In this case, cleaning means the process of eliminating noises that are not needed for the sentiment analysis. As you can see from the above code, for example, we can

remove if the text contains html tags. Next, normalization is the process to match words that have the same meaning but was expressed differently. For instance, Good and good are the same word but the computer might classify them as different words because of the capitalization. Therefore, we have to normalize the data by converting all characters with lowercase characters.

Also, we removed unnecessary words. First, words that are rarely used are removed. For example, if a word is used only 3 times in total for 50,000 movie reviews, we can assume that it would hardly help the language processing. Next, we remove very short words(1~2 characters). Those words, such as a, I, he, do not have much meaning most of the time and serve a grammatical role, therefore, we can remove such words.

3.4. Stemming and Lemmatization

```
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

s=PorterStemmer()
print([s.stem(w) for w in words])

n=WordNetLemmatizer()
print([n.lemmatize(w) for w in words])
```

Stemming or Lemmatization refers to the process of changing the word to its original form. For example, formalize → formal and allowance → allow. The purpose of this process is to match words that have the same meaning but are expressed differently in the grammatical context. They may appear as different words at first glance, however, generalizing these words into a single word can improve the performance of the machine learning model by reducing the number of words in the document.

3.5. Stopwords

```
stop=set(stopwords.words('english'))
#removing the stopwords
def remove_stopwords(text):
    tokens = tokenizer.tokenize(text)
    tokens = [token.strip() for token in tokens]
            filtered_tokens = [token for token in
tokens if token.lower() not in stopwords]
    return ' '.join(filtered_tokens)
```

To extract meaningful word tokens from the data, we need to remove word tokens that do not have significant meaning. In this case, words that do not have significant meaning refers to the words that are used frequently but are not quite helpful when analyzing the text. Words such as I, my, me, over, for example, often appear in the sentences, but they rarely contribute to the actual analysis of the meaning. These words are called stopwords, and NLTK has over 100 predefined English stopwords in their package. The above code uses the predefined stopwords list to remove these words from the data.

3.6. Regular Expression

```
def remove_special_symobls(text):
    text=re.sub(r'[^a-zA-z0-9\s]','',text)
    return text
```

A regular expression is a method to find a string that has a specific pattern. By using this method, we can find and remove certain patterns that are not filtered during the above processes. For example, mention (@id) or hashtag (#hashtag) in Twitter messages fit into this category. This

process can vary depends on the type of data being analyzed which requires understanding the characteristics of the data. This was only used to remove special characters from the movie review data.

4 Experiments

For the experiment, the author verified the performance of created machine learning model using the following method. First, we loaded all of the 50,000 movie review data and preprocessed the data using the methods explained in Section 3. Then, 70% of the data was used as training data and 30% was used as test data for the experiment. The reason why we need to divide training data and test data for the experiment is because a phenomenon called overfitting may occur if we use all the data for training. Overfitting means that the machine learning model was over-fitted that it found an extreme pattern which only works on the given data. This is undesirable since the model will not work in a general case. [4]

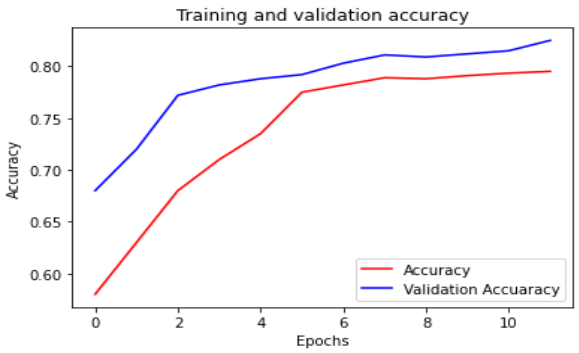

Figure 3: Training and validation accuracy graph

We chose the KNN algorithm for the machine learning model that was used for the experiment and the appropriate k value was chosen through trial and error. As a result, the model was able to classify whether the review is positive or negative with 82.5% accuracy. The above figure is the learning curve of the machine learning model that was used for the experiment and the below figure is examples of the result.

```
predict("The best movie of the year! The main actor is a great actor.")
'Positive'
predict("The background music didn't match the mood of the movie.")
'Negative'
```

5 Conclusion

In this study, the author has developed a machine learning model that can predict if the person thinks positively or negatively about the movie based on the movie review data. To this end, the given data were first preprocessed to turn it into a dataset suitable for training. Then, the machine learning model using the KNN algorithm was trained and the model was able to predict peoples' sentiment with 82.5% accuracy.
Since the above program uses the KNN algorithm, it does not need to know how negative or positive the word is, unlike the previous methods. This is because the KNN algorithm classifies reviews that show similar patterns based on their proximity. In this respect, the result of this study has a significant meaning.
For further research, the model can perform better if it can interpret the meaning based on the actual context, rather than simply comparing the similarity of the words. This requires a model called RNN which can be an interesting challenge.

References

[1] Expert System Team. "What is Machine Learning? A Definition." Expert.ai. Last modified May 6, 2020. https://www.expert.ai/blog/machine-learning-definition/#:~:text=Machine%20learning%20is%20an%20application,use%20it%20learn%20for%20themselves.

[2] MonkeyLearn. "Everything There Is to Know about Sentiment Analysis." MonkeyLearn. https://monkeylearn.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20(or%20opinion%20mining,feedback%2C%20and%20understand%20customer%20needs.

[3] Stanford CS231n notes, "A Complete Guide to K-Nearest-Neighbors with Applications in Python and R", https://cs231n.github.io/classification/#nn

[4] Hawkins, Douglas M. "The Problem of Overfitting." National Library of Medicine. Last modified February 2004. https://pubmed.ncbi.nlm.nih.gov/14741005/.

[5] Stanford “Large Movie Review Dataset”, https://ai.stanford.edu/~amaas/data/sentiment/