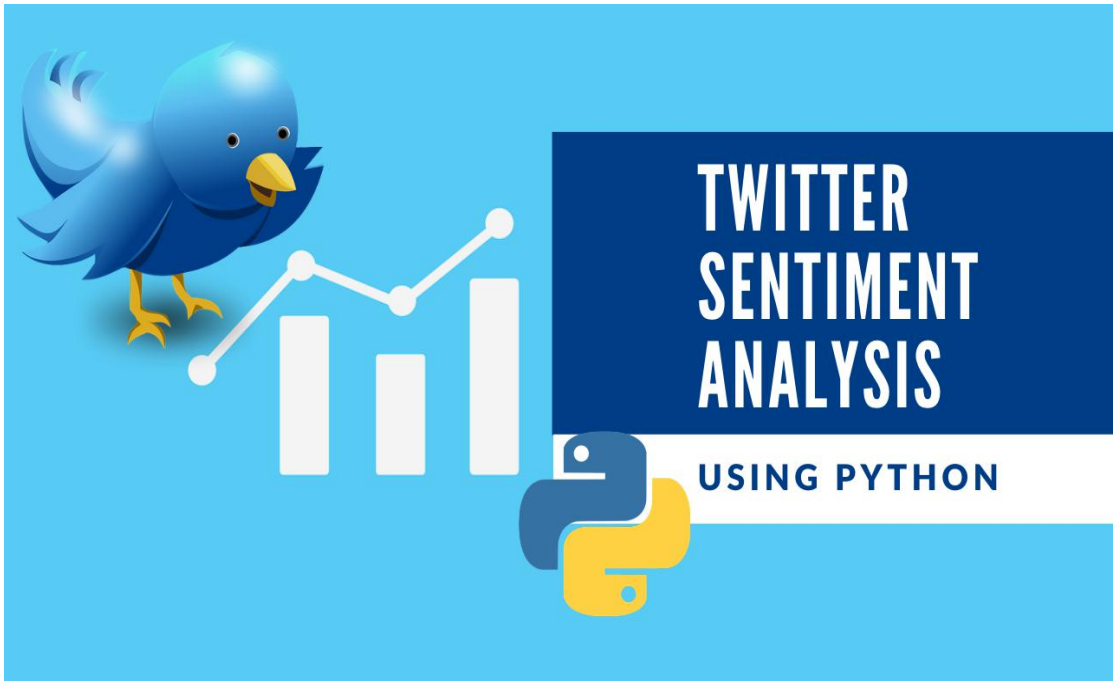
Project Report

Sentimental analysis of Tweets

IIT Kanpur - Summer Internship 2020
Applied Machine Learning and Data Science 2020
Course Code - 002



Project Members

1. Kaustubh Dwivedi (kaustubhdwivedi1729@gmail.com)
2. Tanmay Misra (tanmaymisra619@gmail.com)

Introduction

Context

This project has been done as a part of our Summer Internship at Indian Institute of Technology, Kanpur supervised by Dr Laxmidhar Behera, we had three months to fulfill the requirements in order to succeed the module. Every three weeks, a meeting was organized to show and report our progress and fix the next objectives.

Motivations

Being extremely interested in everything having a relation with the Machine Learning, the in-dependant project was a great occasion to give me the time to learn and confirm my interest for this field. The fact that we can make estimations, predictions and give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities. We can use Machine Learning in Finance, Medicine, almost everywhere. That's why I decided to conduct my project around the Machine Learning.

Idea

This project was motivated by my desire to investigate the sentiment analysis field of machine learning since it allows to approach natural language processing which is a very hot topic actually.We have learned NLP under supervision of Dr. Pawan Goyal and applied it's concepts with tweets and try to figure out which is positive or negative or neutral.

The Project

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on. Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,...) but it is also why it is very interesting to working on. In this project I choose to try to classify tweets from Twitter into “positive” or “negative” sentiment by building a model based on probabilities. Twitter is a microblogging website where people can share their feelings quickly and spontaneously by sending a tweets limited by 140 characters. You can directly address a tweet to someone by adding the target sign “@” or participate to a topic by adding an hastag “#” to your tweet. Because of the usage of Twitter, it is a perfect source of data to determine the current overall opinion about anything

What is Sentimental Analysis ?

Sentiment analysis is a machine learning technique that detects polarity (e.g. a positive or negative opinion) within text, whether a whole document, paragraph, sentence, or clause.

Understanding people's emotions is essential for businesses since customers are able to express their thoughts and feelings more openly than ever before. By automatically analyzing customer feedback, from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs.

For example, using sentiment analysis to automatically analyze 4,000+ reviews about your product could help you discover if customers are happy about your pricing plans and customer service.

Types of Sentiment Analysis

Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), and even on intentions (e.g. interested v. not interested).

Here are some of the most popular types of sentiment analysis:

1. Fine-grained Sentiment Analysis

If polarity precision is important to your business, you might consider expanding your polarity categories to include:

Very positive
Positive
Neutral
Negative
Very negative

This is usually referred to as fine-grained sentiment analysis, and could be used to interpret 5-star ratings in a review, for example:

Very Positive = 5 stars
Very Negative = 1 star

2. Emotion detection

This type of sentiment analysis aims at detecting emotions, like happiness, frustration, anger, sadness, and so on. Many emotion detection systems use lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms.

One of the downsides of using lexicons is that people express emotions in different ways. Some words that typically express anger, like bad or kill (e.g. your product is so bad or your customer support is killing me) might also express happiness (e.g. this is bad ass or you are killing it).

3. Aspect-based Sentiment Analysis

Usually, when analyzing sentiments of texts, let’s say product reviews, you’ll want to know which particular aspects or features people are mentioning in a positive, neutral, or negative way. That's where aspect-based sentiment analysis can help, for example in this text: "The battery life of this camera is too short", an aspect-based classifier would be able to determine that the sentence expresses a negative opinion about the feature battery life.

4. Multilingual sentiment analysis

Multilingual sentiment analysis can be difficult. It involves a lot of preprocessing and resources. Most of these resources are available online (e.g. sentiment lexicons), while others need to be created (e.g. translated corpora or noise detection algorithms), but you'll need to know how to code to use them.

Why Perform Sentiment Analysis?

It's estimated that 80% of the world's data is unstructured, in other words it's unorganized. Huge volumes of text data (emails, support tickets, chats, social media conversations, surveys, articles, documents, etc), is created every day but it's hard to analyze, understand, and sort through, not to mention time-consuming and expensive.

Sentiment analysis, however, helps businesses make sense of all this unstructured text by automatically tagging it.

Benefits of sentiment analysis include:

Sorting Data at Scale Can you imagine manually sorting through thousands of tweets, customer support conversations, or surveys? There's just too much data to process manually. Sentiment analysis helps businesses process huge amounts of data in an efficient and cost-effective way.

Real-Time Analysis Sentiment analysis can identify critical issues in real-time, for example is a PR crisis on social media escalating? Is an angry customer about to churn? Sentiment analysis models can help you immediately identify these kinds of situations and gauge brand sentiment, so you can take action right away.

Consistent criteria It's estimated that people only agree around 60-65% of the time when determining the sentiment of a particular text. Tagging text by sentiment is highly subjective, influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data, helping them improve accuracy and gain better insights.

Sentiment Analysis Algorithms – How It Works?

Sentiment analysis uses various Natural Language Processing (NLP) methods and algorithms, which we'll go over in more detail in this section.

The main types of algorithms used include:

Rule-based systems that perform sentiment analysis based on a set of manually crafted rules.
Automatic systems that rely on machine learning techniques to learn from data.
Hybrid systems that combine both rule-based and automatic approaches.
Rule-based Approaches
Usually, a rule-based system uses a set of human-crafted rules to help identify subjectivity, polarity, or the subject of an opinion.

These rules may include various techniques developed in computational linguistics, such as:

Stemming, tokenization, part-of-speech tagging and parsing.
Lexicons (i.e. lists of words and expressions).
Here's a basic example of how a rule-based system works:

Defines two lists of polarized words (e.g. negative words such as bad, worst, ugly, etc and positive words such as good, best, beautiful, etc).
Counts the number of positive and negative words that appear in a given text.
If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral sentiment.
Rule-based systems are very naive since they don't take into account how words are combined in a sequence. Of course, more advanced processing techniques can be used, and new rules added to support new expressions and vocabulary. However, adding new rules may affect previous results, and the whole system can get very complex. Since rule-based systems often require fine-tuning and maintenance, they'll also need regular investments.

Automatic Approaches

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. A sentiment analysis task is usually modeled as a classification problem, whereby a classifier is fed a text and returns a category, e.g. positive, negative, or neutral.

Here's how a machine learning classifier can be implemented:

1.The Training and Prediction Processes
In the training process (a), our model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the test samples used for training. The feature

extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g. positive, negative, or neutral) are fed into the machine learning algorithm to generate a model.

In the prediction process (b), the feature extractor is used to transform unseen text inputs into feature vectors. These feature vectors are then fed into the model, which generates predicted tags (again, positive, negative, or neutral).

2.Feature Extraction from Text

The first step in a machine learning text classifier is to transform the text extraction or text vectorization, and the classical approach has been bag-of-words or bag-of-ngrams with their frequency.

More recently, new feature extraction techniques have been applied based on word embeddings (also known as word vectors). This kind of representations makes it possible for words with similar meaning to have a similar representation, which can improve the performance of classifiers.

3.Classification Algorithms

The classification step usually involves a statistical model like Naïve Bayes, Logistic Regression, Support Vector Machines, or Neural Networks:

Naïve Bayes: a family of probabilistic algorithms that uses Bayes's Theorem to predict the category of a text.

Linear Regression: a very well-known algorithm in statistics used to predict some value (Y) given a set of features (X).

Support Vector Machines: a non-probabilistic model which uses a representation of text examples as points in a multidimensional space. Examples of different categories

(sentiments) are mapped to distinct regions within that space. Then, new texts are assigned a category based on similarities with existing texts and the regions they're mapped to.

Deep Learning: a diverse set of algorithms that attempt to mimic the human brain, by employing artificial neural networks to process data.

4.Hybrid Approaches

Hybrid systems combine the desirable elements of rule-based and automatic techniques into one system. One huge benefit of these systems is that results are often more accurate.

Sentiment Analysis Challenges

Computer scientists have been trying to develop more accurate sentiment classifiers, and overcome limitations in recent years. Let's take a closer look at some of the challenges they face:

Subjectivity and Tone

The detection of subjective and objective texts is just as important as analyzing their tone. In fact, so called objective texts do not contain explicit sentiments. Say, for example, you intend to analyze the sentiment of the following two texts:

The package is nice.
The package is red.

Most people would say that sentiment is positive for the first one and neutral for the second one, right? All predicates (adjectives, verbs, and some nouns) should not be treated the same with respect to how they create sentiment. In the examples above, nice is more subjective than red.

Context and Polarity

All utterances are uttered at some point in time, in some place, by and to some people, you get the point. All utterances are uttered in context. Analyzing sentiment without context gets pretty difficult. However, machines cannot learn about contexts if they are not mentioned explicitly. One of the problems that arise from context is changes in polarity. Look at the following responses to a survey:

Everything of it.

Absolutely nothing!

Imagine the responses above come from answers to the question What did you like about the event? The first response would be positive and the second one would be negative, right? Now, imagine the responses come from answers to the question What did you Dislike about the event? The negative in the question will make sentiment analysis change altogether.

A good deal of preprocessing or postprocessing will be needed if we are to take into account at least part of the context in which texts were produced. However, how to preprocess or postprocess data in order to capture the bits of context that will help analyze sentiment is not straightforward.

Irony and Sarcasm

When it comes to irony and sarcasm, people express their negative sentiments using positive words, which can be difficult for machines to detect without having a thorough understanding of the context of the situation in which a feeling was expressed.

What sentiment would you assign to the responses above? The first response with an exclamation mark could be negative, right? The problem is there is no textual cue that will help a machine learn, or at least question that sentiment since yeah and sure often belong to positive or neutral texts.

How about the second response? In this context, sentiment is positive, but we're sure you can come up with many different contexts in which the same response can express negative sentiment.

Comparisons

How to treat comparisons in sentiment analysis is another challenge worth tackling. Look at the texts below:

This product is second to none.
This is better than older tools.
This is better than nothing.

The first comparison doesn't need any contextual clues to be classified correctly. It's clear that it's positive.

The second and third texts are a little more difficult to classify, though. Would you classify them as neutral, positive, or even negative? Once again, context can make a difference. For example, if the 'older tools' in the second text were considered useless, then the second text is pretty similar to the third text.

Emojis

There are two types of emojis according to Guibon et al.. Western emojis (e.g. :D) are encoded in only one or two characters, whereas Eastern emojis (e.g. ¯ \ _ (ツ) _ / ¯) are a

longer combination of characters of a vertical nature. Emojis play an important role in the sentiment of texts, particularly in tweets.

You'll need to pay special attention to character-level, as well as word-level, when performing sentiment analysis on tweets. A lot of preprocessing might also be needed. For example, you might want to preprocess social media content and transform both Western and Eastern emojis into tokens and whitelist them (i.e. always take them as a feature for classification purposes) in order to help improve sentiment analysis performance.

Here's a quite comprehensive list of emojis and their unicode characters that may come in handy when preprocessing.

Defining Neutral

Defining what we mean by neutral is another challenge to tackle in order to perform accurate sentiment analysis. As in all classification problems, defining your categories -and, in this case, the neutral tag- is one of the most important parts of the problem. What you mean by neutral, positive, or negative does matter when you train sentiment analysis models. Since tagging data requires that tagging criteria be consistent, a good definition of the problem is a must.

How Accurate Is Sentiment Analysis?

Here's what sentiment analysis is: it's a tremendously difficult task even for human beings. That said, sentiment analysis classifiers might not be as precise as other types of classifiers. Remember that inter-annotator agreement is pretty low and

that machines learn from the data they are fed with (see above).

That said, you might be saying, is it worth the effort? The answer is simple: it sure is worth it! Chances are that sentiment analysis predictions will be wrong from time to time, but by using sentiment analysis you will get the opportunity to get it right about 70-80% of the times you submit your texts for classification.

If you or your company have not used sentiment analysis before, then you'll see some improvement really quickly. For typical use cases, such as ticket routing, brand monitoring, and VoC analysis (see below), this means you will save a lot of time and money -which you are likely to be investing in in-house manual work nowadays,- save your teams some frustration, and increase your (or your company's) productivity.

Conclusion

Nowadays, sentiment analysis or opinion mining is a hot topic in machine learning. We are still far to detect the sentiments of s corpus of texts very accurately because of the complexity in the English language and even more if we consider other languages such as Chinese.