Sentiment Analysis using Natural Language Processing

Atish Deore[1], Yash Bijore[2], Chaitanya Gudhate[3], S.K. Patil[4], M.D.Katole[5]

[1,2,3,4,5]Dept. of E & TC Engg.Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune

deoreatish9@gmail.com

yashbijore@gmail.com

chaitanyagudghate@gmail.com

skpatil_skncoe@sinhgad.edu

katole.mukesh@gmail.com

Abstract

Nowadays text synthesis is most vital part of web technology. NLP are often a subfield of linguistics, computing, information engineering, and AI concerned with the interactions between computers and human (natural) languages, especially the way to program computers to process and analyze large amounts of tongue data. Natural Language Toolkit also known as NLTK is to be used in which there are pre-existing provisions and libraries which are required for this particular application. The programming language to be used is Python. The system will be designed such that it will result into the sentiment of the given sentence. In Layman's terms, it would indicate if the database is positive, neutral or negative. This technology is already implemented in Google search recommendations, YouTube search recommendations, etc.

Keywords-*NLP, NLTK, python, tokenization, normalization.*

I. INTRODUCTION

Everything we express (either verbally or in written) carries huge amounts of data . The topic we elect, our tone, our selection of words, everything adds some sort of information which will be interpreted and value extracted from it.[1] In theory, we will understand and even predict human behavior using that information.

But there is a problem: one person may generate hundreds or thousands of words during a declaration, each sentence with its corresponding complexity.[2] If you would like to scale and analyze several hundreds, thousands or many people or declarations during a given geography, then things is unmanageable.

Data generated from conversations, declarations or maybe tweets are samples of unstructured data.[6] Unstructured data doesn't fit neatly into the normal row and column structure of relational databases, and represent the overwhelming majority of knowledge available within the actual world. It is messy and hard to manipulate. Nevertheless, because of the advances in disciplines like machine learning an enormous revolution goes on regarding this subject.[8] Nowadays it is not about trying to interpret a text supported its keywords, but about understanding the meaning behind those words (the cognitive way). This way it's possible to detect figures of speech like irony, or maybe perform sentiment analysis.

Natural language processing could also be a subfield of linguistics, computing , information engineering, and AI concerned with the interactions between computers and human (natural) languages, especially the thanks to program computers to process and analyze large amounts of natural language data.[3] NLP is a part of computer science and artificial intelligence concerned with interactions between computers and human (natural) languages. It is used to apply machine learning algorithms to text and speech.

II. LITERATURE SURVEY

Sr. No.	Reference	Purpose	Merits	Demerits
[1]	Fundamentals of Sentiment Analysis and Its Applications (A. Kamal, M. Abulaish, and Jahiruddin,)	This paper talks about the challenges appearing with handling Big Data with techniques such as Natural Language Processing, data mining, text mining in the fields of Management, Marketing and many more. .	Basic concepts of sentiment analysis explained	Only Generalized methodology explained
[2]	Survey on Sentiment Analysis: A Comparative Study (H. Tanaya, S. Sagnika, and L. Sahoo)	This paper was a reference to structured data, unstructured data and different techniques to deal with it. .	More detailed model of sentiment analysis is given	Techniques not much used
[3]	Sentiment Analysis onTwitter Data Using Machine Learning Algorithms in Python (K. Ganagavalli, A. Mangayarkarasi, T. Nandhinisri, and E. Nandhini)	Ways to deal with textual data such as tweets and different machine learning algorithms available in Python to work on it.	Detailed information about importing datasets, applying ML algorithms is provided	Algorithm not much effective
[4]	Sentiment Analysis of Twitter Data using Python (L. Branz and P. Brockmann)	This paper stated the importance of Integrated Development environments (IDEs) such as Anaconda and PyCharm while writing the code. .	Handling twitter data Using python is explained	Methodology not Explained in detail
[5]	Sentiment Analysis in Python using NLTK (I. V Shravan and C. Networking)	Different libraries available in Python to do Sentiment Analysis especially NLTK and how to use it.	Sentiment Analysis in Python using NLTK Explained in detail	Do not meet to expected algorithms
[6]	Latent Dirichlet allocation J. Mach. Learn (D. M. Blei, A. Y. Ng, and M. I. Jordan)	We describe latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora.	Setting up the corpus i.e. Dataset is very well explained.	Not suitable for All applications
[7]	A holistic lexicon-based approach to opinion mining (X. Ding, B. Liu, and P. S. Yu)	One of the important types of information of the reference is the opinions expressed in the user generated content, e.g., customer reviews of products, forum posts, and blogs.	Opinion mining with Lexicon based approach is mentioned.	Algorithm has to undergo more training.
[8]	Mining the peanut	This reference begins by	Various	Modern

	gallery: Opinion extraction and semantic classification of product reviews (K. Dave, S. Lawrence, and D. M. Pennock)	identifying the unique properties of this problem and develop a method for automatically distinguishing between positive and negative reviews.	algorithms for successful classification of text are mentioned.	algorithms not Mentioned.

III. NATURAL LANGUAGE PROCESSING

The essence of Natural Language Processing lies in making computers understands the natural language. That's not an easy task though. Computers can understand the structured form of data like spreadsheets and the tables in the database, but human languages, texts, and voices form an unstructured category of data, and it gets difficult for the computer to understand it, and there arises the need for Natural Language Processing. There's a lot of natural language data out there in various forms and it would get very easy if computers can understand and process that data. We can train the models in accordance with expected output in different ways. Humans have been writing for thousands of years, there are a lot of literature pieces available, and it would be great if we make computers understand that. But the task is never going to be easy. There are various challenges floating out there like understanding the correct meaning of the sentence, correct Named-Entity Recognition(NER), correct prediction of various parts of speech, coreference resolution(the most challenging thing in my opinion). Computers can't truly understand the human language. If we feed enough data and train a model properly, it can distinguish and try categorizing various parts of speech(noun, verb, adjective, supporter, etc…) based on previously fed data and experiences. If it encounters a new word it tried making the nearest guess which can be embarrassingly wrong few times.

It's very difficult for a computer to extract the exact meaning from a sentence. For example – The boy radiated fire like vibes. The boy had a very motivating personality or he actually radiated fire? As you see over here, parsing English with a computer is going to be complicated.

Types of NLP are: Percy Liang, a Stanford CS professor and NLP expert, breaks down the various approaches to NLP / NLU into four distinct categories: 1) Distributional 2) Frame-based 3) Model-theoretical 4) Interactive learning

.

IV. PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- Python 2 and Python 3. Both are quite different

A. FEATURES: The features of Python are: A. Easy

When we say the word 'easy', we mean it in different contexts.

a. Easy to Code

Python is very easy to code. Compared to other popular languages like Java and C++, it is easier to code in Python. Anyone can learn Python syntax in just a few hours. Though sure, mastering Python requires learning about all its advanced concepts and packages and modules. That takes time. Thus, it is programmer-friendly.

b. Expressive

First, let's learn about expressiveness. Suppose we have two languages A and B, and all programs that can be made in A can be made in B using local transformations. However, there are some programs that can be made in B, but not in A, using local transformations. Then, B is said to be more expressive than A. Python provides us with a myriad of constructs that help us focus on the solution rather than on the syntax. This is one of the outstanding python features that tell you why you should learn Python.

c. Free and Open-Source
Firstly, Python is freely available. You can download it from the Python Website. Secondly, it is open-source. This means that its source code is available to the public. You can download it, change it, use it, and distribute it. This is called FLOSS(Free/Libre and Open Source Software). As the Python community, we're all headed toward one goal- an ever-bettering Python.

V. METHODOLOGY

A large amount of data that is generated today is unstructured, which requires processing to generate insights. Some examples of unstructured data are news articles, posts on social media, and search history. The process of analyzing natural language and making sense out of it falls under the field of Natural Language Processing (NLP). Sentiment analysis is a common NLP task, which involves classifying texts or parts of texts into a pre-defined sentiment. You will use the Natural Language Toolkit (NLTK), a commonly used NLP library in Python, to analyze textual data.
Tokenization: Tokenization is the process of converting text into tokens before transforming it into vectors. It is also easier to filter out unnecessary tokens. For example, a document into paragraphs or sentences into words.
Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.
Removing stop words: Stop words are the most commonly occurring words which are not relevant in the context of the data and do not contribute any deeper meaning to the phrase. In this case contain no sentiment.
Normalization:
Words which look different due to casing or written another way but are the same in meaning need to be process correctly. Normalization processes ensure that these words are treated equally. For example, changing numbers to their word equivalents or converting the casing of all the text.
'100' → 'one hundred'
'Apple' → 'apple'
The following normalization changes are made:
1. Casing the Characters
Converting character to the same case so the same words are recognised as the same. In this case we converted to lowercase.
2. Negation handling
Apostrophes connecting words are used everywhere, especially in public reviews. To maintain uniform structure it is recommended they should be converted into standard lexicons. The text will then follow the rules of context free grammar and helps avoids any word-sense disambiguation.

3. Removing
Stand alone punctuations, special characters and numerical tokens are removed as they do not contribute to sentiment which leaves only alphabetic characters. This step needs the use of tokenized words as they have been split appropriately for us to remove.
This process finds the base or dictionary form of the word known as the lemma. This is done through the use of vocabulary (dictionary importance of words) and morphological analysis (word structure and grammar relations). This normalization is similar to stemming but takes into account the context of the word.
'are', 'is', 'being' → 'be'

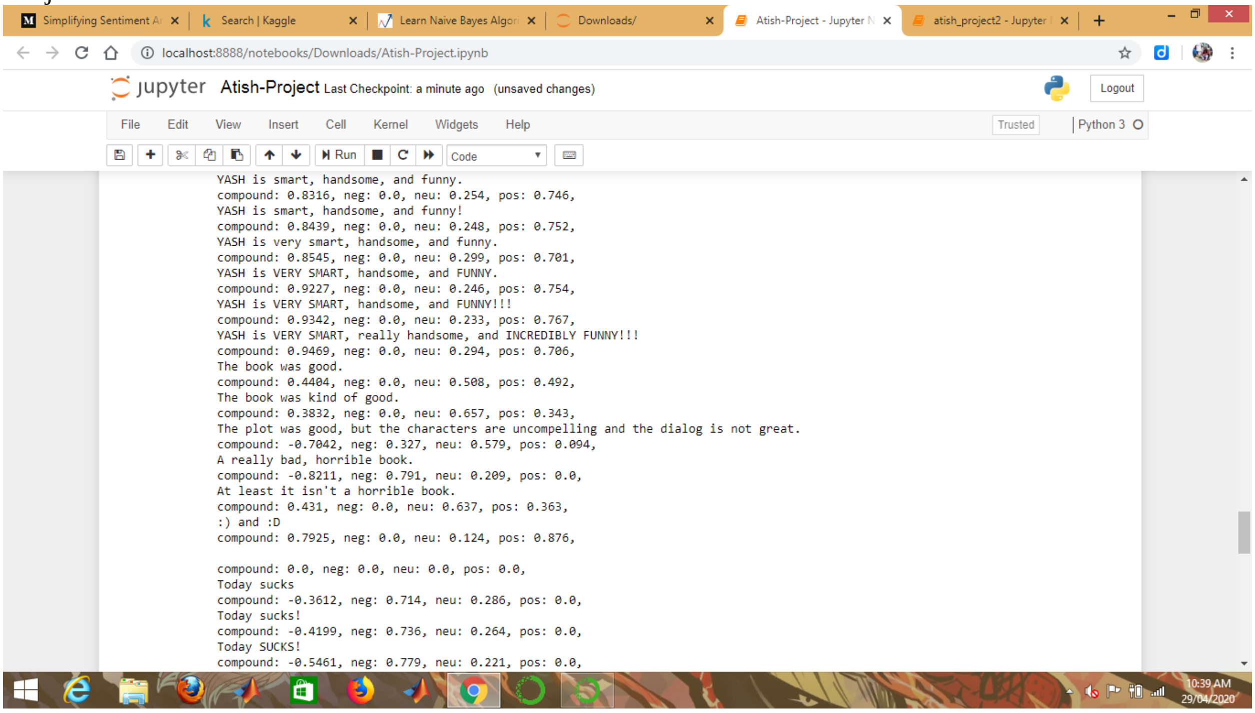
4. Substitution:
This involves removing noise from text in its raw format. For example, the text is scrapped from the web it may contain HTML or XML wrappers or markups. Removal of these can be done through regular expressions.
5. Normalizing the Data:
Words have different forms—for instance, "ran", "runs", and "running" are various forms of the same verb, "run". Depending on the requirement of your analysis, all of these versions may need to be

converted to the same form, “run”. Normalization in NLP is the process of converting a word to its canonical form.

We will use regular expressions in Python to search for and remove these items:
□ Hyperlinks - All hyperlinks in text are converted to the URL shortener. Therefore, keeping them in the text processing would not add any value to the analysis.
□ Twitter handles in replies - These Twitter usernames are preceded by a @ symbol, which does not convey any meaning.
□ Punctuation and special characters - While these often provide context to textual data, this context is often difficult to process. For simplicity, you will remove all punctuation and special characters from tweets.
6. Preparing Data for the Model:-
Sentiment analysis is a process of identifying an attitude of the author on a topic that is being written about. You will create a training data set to train a model. It is a supervised machine learning process, which requires you to associate each dataset with a “sentiment” for training. In this tutorial, your model will use the “positive” and “negative” sentiments.
7. Converting Tokens to a Dictionary :-
First, we will prepare the data to be fed into the model. You will use the Naive Bayes classifier in NLTK to perform the modeling exercise. Notice that the model requires not just a list of words in a text, but a Python dictionary with words as keys and True as values.

VI. RESULTS

Thus, the analysis of the text we get as a result of the above has the values between 0 and 1. Here we get the polarity scores such as positive, negative, neutral. Compound term denotes the clauses and conjunctions that are used in the said sentence.



```
YASH is smart, handsome, and funny.
compound: 0.8316, neg: 0.0, neu: 0.254, pos: 0.746,
YASH is smart, handsome, and funny!
compound: 0.8439, neg: 0.0, neu: 0.248, pos: 0.752,
YASH is very smart, handsome, and funny.
compound: 0.8545, neg: 0.0, neu: 0.299, pos: 0.701,
YASH is VERY SMART, handsome, and FUNNY.
compound: 0.9227, neg: 0.0, neu: 0.246, pos: 0.754,
YASH is VERY SMART, handsome, and FUNNY!!!
compound: 0.9342, neg: 0.0, neu: 0.233, pos: 0.767,
YASH is VERY SMART, really handsome, and INCREDIBLY FUNNY!!!
compound: 0.9469, neg: 0.0, neu: 0.294, pos: 0.706,
The book was good.
compound: 0.4404, neg: 0.0, neu: 0.508, pos: 0.492,
The book was kind of good.
compound: 0.3832, neg: 0.0, neu: 0.657, pos: 0.343,
The plot was good, but the characters are uncompelling and the dialog is not great.
compound: -0.7042, neg: 0.327, neu: 0.579, pos: 0.094,
A really bad, horrible book.
compound: -0.8211, neg: 0.791, neu: 0.209, pos: 0.0,
At least it isn't a horrible book.
compound: 0.431, neg: 0.0, neu: 0.637, pos: 0.363,
:) and :D
compound: 0.7925, neg: 0.0, neu: 0.124, pos: 0.876,

compound: 0.0, neg: 0.0, neu: 0.0, pos: 0.0,
Today sucks
compound: -0.3612, neg: 0.714, neu: 0.286, pos: 0.0,
Today sucks!
compound: -0.4199, neg: 0.736, neu: 0.264, pos: 0.0,
Today SUCKS!
compound: -0.5461, neg: 0.779, neu: 0.221, pos: 0.0,
```

VII. CONCLUSION

Subjectivity and sentiment analysis is an emerging field in NLP with very interesting applications. A lot can be learned from the amount of unstructured/structured information on the web which can aid in subjectivity and sentiment analysis.
Simple sentiment of the given text is the expected output from this project which is efficiently done by the use of Python, NLTK and Naïve-Bayes Algorithm. Annotations, abbreviations and sarcasm are the challenges faced in this sentiment analysis.

REFERENCES

[1] L. Branz and P. Brockmann, "Sentiment Analysis of Twitter Data using Python" pp. 238–241, 2018.

[2] K. Ganagavalli, A. Mangayarkarasi, T. Nandhinisri, and E. Nandhini, "Sentiment analysis of twitter data using machine learning algorithm," J. Comput. Theor. Nanosci., vol. 15, no. 5, pp. 1644–1648, 2018.

[3] A. Kamal, M. Abulaish, and Jahiruddin, "Fundamentals of Sentiment Analysis and Its Applications" Stud. Comput. Intell., vol. 639, no. August 2016, pp. 399–423, 2016.

[4] I. V Shravan and C. Networking, "Sentiment Analysis in Python using NLTK," no. January, 2017.

[5] H. Tanaya, S. Sagnika, and L. Sahoo, "Survey on Sentiment Analysis: A Comparative Study," Int. J. Comput. Appl., vol.159, no. 6, pp. 4–7, 2017.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. 4–5, pp. 993–1022, 2003.

[7] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," *WSDM'08 - Proc. 2008 Int. Conf. Web Search Data Min.*, pp. 231–239, 2008.

[8] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," *Proc. 12th Int. Conf. World Wide Web, WWW 2003*, pp. 519–528, 2003.