Summer Internship Report
IIT Tirupati

Facebook Posts Sentiment Analyzer

Submitted by

Ms. Supria Basak
106118116
Summer Research Fellow

Under the guidance of

Dr. Sridhar Chinmalokonda
Assistant Professor, IIT Tirupati

RISHA Lab, IIT Tirupati

Acknowledgements

First of all, I am grateful to my college NIT, Trichy for giving me the time and creating such a wonderful opportunity to learn in my summer vacations.

I am really fortunate that IIT Tirupati gave me this amazing opportunity to intern in their institute even though they hardly take any interns from other colleges.

I would like to express my special thanks of gratitude to my Guide Dr. Sridhar Chimalakonda sir for his guidance and constant supervision as well as for providing the necessary information regarding the project. His immense knowledge and plentiful experience have encouraged me in all the time of my research.

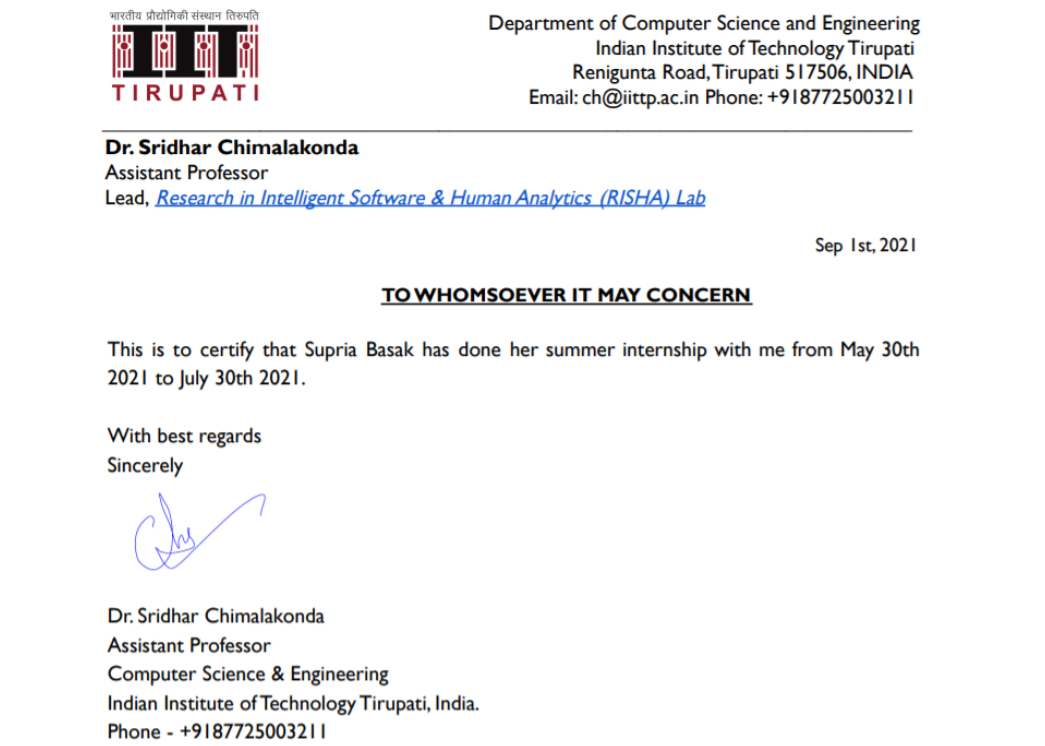
Next I would like to extend my gratitude towards the PhD scholars of RISHA Labs - Akhila Mam and Eshan Sir for their support and encouragement which helped me immensely. Their valuable comments and advises helped me to improve my overall project and also my presentation skills.

Special mention to Rishitha Mam, an immediate senior who has willingly helped me out with her insights and knowledge. Without her contribution in stimulating suggestions and encouragement, it would be impossible for me to complete my study.

Furthermore, I would like to express my deepest appreciation to my Research Partner Sampurn Anand(106118085) who played a crucial role and equally contributed to the project.

Last but not the least, I am grateful to my parents and Research Lab fellows for their constant encouragement and kind cooperation.

Internship Certificate



भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI

Department of Computer Science and Engineering
Indian Institute of Technology Tirupati
Renigunta Road, Tirupati 517506, INDIA
Email: ch@iittp.ac.in Phone: +9187725003211

Dr. Sridhar Chimalakonda
Assistant Professor
Lead, *Research in Intelligent Software & Human Analytics (RISHA) Lab*

Sep 1st, 2021

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Supria Basak has done her summer internship with me from May 30th 2021 to July 30th 2021.

With best regards
Sincerely

Dr. Sridhar Chimalakonda
Assistant Professor
Computer Science & Engineering
Indian Institute of Technology Tirupati, India.
Phone - +9187725003211

Dr. Sridhar Chimalakonda
(Project Guide)

Date: 1st September, 2021

Abstract

The objective of my research internship was to present a paper in an A* conference and then get it published in any reputed journals. After rigorous discussion, I, my research partner Sampurn Anand and Our supervisor Dr. Sridhar Chimalakonda decided to work on a tool that will assess our mental well-being by discovering depression in social media. Even though our road-map initially revolved around assessing each user's emotional state, due to strict guidelines imposed by the Facebook authority, we decided to execute our research on public pages. In the two months, I experimented thoroughly with various NLP methods and algorithms to find the suitable one for our research base and worked with other research scholars to get familiar with different research backgrounds related to computer science. The project being an ongoing one requires **confidentiality**. So, keeping that in mind this intern report has been written.

Contents

Chapter 1

Industry

RISHA(Research in Intelligent Software and Human Analytics) is a research lab at IIT Tirupati that strives to expand the boundaries of Software Engineering, focusing on cutting-edge research, building tools, and leveraging the power of computing to help society (Educational Technologies and Human-Computer Interaction). Their research is essentially about deriving and integrating empirical and theoretical knowledge.

Their prime motto is to aspire high, set benchmarks, and give their best towards creating a remarkable impact in research, development, and society.

Software is predominant today in all walks of life. On the other hand, the software is quite effort-intensive, increasingly complex, and buggy! RISHA lab focuses on cutting-edge research and develops tools in multiple areas of Software Engineering (SE) such as Bug Detection, Code Comprehension, Semantic Code Search, API Deprecation, AI for SE, SE for AI, Energy-Aware Software Engineering, Modernizing legacy code, and so on.

Their other thrust area is Computing for Society (specifically Educational Technologies and Human-Computer Interaction).

They had a decent impact in developing multiple solutions (YTCoder, SurviveCovid-19, Mood of India, and so on).

Software Engineering

They research effective and scalable ways to help developers improve software quality by qualitatively and quantitatively analyzing a diversified range of software artifacts (such as code, bugs, logs, commits, designs, patterns, and so on) in millions of software repositories.

Computing for Society

They leverage advances in computing (such as Software Engineering, Artificial Intelligence, Augmented Reality, Visualization, Social Media Analytics, User Interfaces) to address societal challenges currently in the domains of

i education

ii healthcare

iii environment, and

iv cultural heritage.

Current Research Areas

Empirical Software Engineering, Semantic Code Search, Code Smells, Summarization, Modernizing legacy code, Novel Source Code Representations, Software Documentation, Architecture, Quality, APIs, Cross-Project Learning, Knowledge Graphs, Bug Localization, SE for AI, AI for SE, Energy-Aware Software Engineering and Tools.

Educational Technologies, Personalized Learning, Learning Analytics, Gamification, Ontologies, Virtual & Augmented Reality, Virtual Labs, Human-Computer Interaction, User Interfaces, and Usability.

Collaborators / Sponsors

University of Waterloo, Bosch, CodeChef, Richard Lounsbery Foundation, Accenture labs, commonwealth educational media center for Asia(CEMCA), and so on.

What's required to succeed in RISHA Lab?

→ Technical knowledge, strong in some programming languages, version control, good coding practices, and skills to develop open-source software. Most importantly, understanding existing source code, making quick extensions, and using available open-source frameworks to get tasks done.

→ Stellar problem solving, programming, and technical writing skills.

→ The grip on Literature! The ability to understand Literature, existing research papers, techniques, tools, relate, and most importantly, when to use which techniques/tools and their trade-offs.

→ Ability to quickly learn and apply methods, tools, and techniques from areas such as Machine Learning, Natural Language Processing, and so on!

→ Writing skills! One can get this only through scientific reading and writing! (reading one paper per day till one reaches 100 papers!)

Chapter 2

Training Schedule

2.1 First Week

2.1.1 Goals

1. To get familiar with the way of writing Research Papers.
2. To get familiar with the Hot-Topics available for research in a particular domain and choose a particular topic.
3. Write Literature Reviews of different Research Papers.

2.1.2 Takeaways

1. Learnt LaTex and how to write Research Paper in a proper format.
2. Decided on the topic after exploring a wide range of available topics in the field of ML and NLP.
3. Started analyzing different available works on the chosen topic and wrote their literature reviews.

2.2 Second Week

2.2.1 Goals

1. To learn how to code using Python and Implement different Deep Learning and Natural Language Processing Algorithms.
2. To decide on a particular flow of Algorithms.
3. Explore more about A* Conferences, Journals, Funding, etc.

2.2.2 Takeaways

1. Revised all the important concepts of Python and learnt how to implement various algorithms such as SMC, Word Mover's Distance, etc.
2. Made a flowchart of algorithm to be used by hit and trial.
3. Got to know about various Conferences and Journals around the globe.
4. Got familiar with the research culture in real life.

Chapter 3

Introduction

With the advancement of social networking sites' popularity, Facebook has reached a milestone of 2.89 billion monthly active users being the biggest social network worldwide. In recent times, most of our mental health is connected to Social Media, Facebook to be specific.[4]

In recent times, rapid growth has been witnessed in the analysis of social media for studying. Effects on mental health due to Facebook posts are tremendous[7]; an intriguing post can provoke our thoughts, can make us think; some happy or positive posts can light up our day. In contrast, negative news and negative interaction can ruin our day. Facebook has taken the place of and holds almost equal importance as real-life communication; since it covers a broader spectrum, choosing our desired kind of interaction is important. [6]

To solve the stated issue, we have implemented a Facebook sentiment analyzer to analyze the textual emotion of the posts posted on pages and reactions. We have collected over 1Million posts from the top 100+ verified Facebook pages using facepager as our data-set. Using the model stated, we introduced a social media emotion meter index that may serve to characterize levels of emotions of a post, and in addition, one can filter out the

negative posts. To cross-check our results, we surveyed 1000 people of different ages, asking about their emotions towards some posts, and the results successfully matched our results.

The task of text emotion analysis mainly includes Sentiment classification, emotion information extraction, and emotion information retrieval and induction [5]. There are two types of machine learning techniques generally used for sentiment analysis, one is unsupervised, and the other is supervised [5]. Unsupervised learning does not consist of a category, and they do not provide the correct targets at all and therefore conduct clustering. Supervised learning is based on a labeled data-set, and thus the labels are provided to the model during the process. These labeled data-sets are trained to produce reasonable outputs when encountered during decision-making.

To help us understand sentiment analysis better, this research paper is based on supervised machine learning.

The proposed method differs from existing methods in terms of the sentiment scale. Instead of using negative, positive, and neutral emotions, we have used the linear scale with a decimal value score. We have implemented a unique multi-modal approach using Simple Matching Coefficient(SMC), Word Mover's distance, Gaussian Mixture Models (GMMs), f-divergence & Word2Vec for feature extraction. Later on, we used Shannon Entropy for feature weighting and SVMs and Decision trees for classification and bagging. Sentiment computation has been done using SentiWordNet Tool.

Chapter 4

Work Done

4.1 Algorithms Used & Methodology

Figure 4.1: Proposed Methodology.

Our proposed methodology shown in the Figure: 4.1 portrays various concepts of Machine Learning, Natural Language Processing and Artificial Intelligence. Here, an aspect-level text

classification has been performed. In the proposed system, a decimal value is obtained as a sentiment value of a particular post. As it can be put on a linear scale of sentiments and then interpreted, it is termed as a probabilistic model. This decimal value can describe the sentiments related to the post in a better way, making our proposed model faster and more accurate.

The step-by-step procedure to perform sentiment analysis on the Facebook posts is listed below:

1. Gathering of data.
   (a) Data is gathered using Facepager.
2. Pre-processing of data
   (a) Tokenization, Normalization and Lemmatization
   (b) Different dictionaries are used. (POS dictionaries and so on.)
3. Feature Extraction
   (a) First step- SMC and Word Mover's Distance.
   (b) Second step - Gaussian Mixture Models and f - divergence
   (c) Third Step - Word2Vec
4. Feature Fusion
5. Feature Weighting
   (a) Shannon Entropy
6. Classification
   (a) Neural Networks
   (b) Gini Index
7. Sentiment Computation

(a) SentiWordNet Tool (From Word-Net database)

8. **Classified Facebook Posts with their sentiment values.**

The above steps are inter-linked. Output form each step is used as an input in the next step. Implementation of the methods is done using different Python Libraries and the data-set obtained form the Facepager Tool.

4.1.1 Data Mining

We have used the Facepager tool [2] to import the data from Facebook. Facepager fetches publicly available data from Facebook, Twitter, and other websites based on APIs and web scraping. Scraped data is stored in an SQLite database and can be exported to CSV. Overall, 1.04M data was extracted from 104 pages using the tool. The dataset consists of message, from, created_time, updated_time, reactions, shares. Our target field contains the posts' captions for textual emotion analysis and the reactions for emoticon sentiment analysis. In the proposed model, the dataset is pre-processed and used further for classification.

4.1.2 Data Pre-processing and Preparation

From the data-set obtained in the previous step, the field named "message" will be used for sentiment analysis. To make this field ready for Sentiment Analysis, following steps are followed which can be performed in any sequence:

- Tokenization - Sentences of the posts are broken down into words and each word is called a Token.
- Normalization - Each word is normalized into standard forms (i.e., all lower cases) using Euclidean Vector Length

Similarity. If there is repetition of letters in a word then it is replaced with the meaningful word. (i.e., sooooo - so) Also, the acronyms are converted to their normal forms. (i.e., EU - European Union)

- Lemmatization - Switches any kind of word to its base root mode. Words in their inflected forms are converted to normal forms. (i.e., Laughing - Laugh)
- Stop-Word Dictionary - For Parts of Speech, each word is tagged either as Stop-Words (ST) or a Non-Stop-Words (NST).
- Emoticon Dictionary - All the emoticons used in the posts are converted to normal text equivalent.
- Every URL is removed from the posts and whenever there is use of "@", the text following it, is replaced by the word "USER". (i.e., @Amir_khan - USER)

Till now, for every post's message field, we have a 2D array of tokens which contain the values: {Words , POS Tags}. Here, all the words are in their normal forms. This 2D array is fed as an input to the next step.

4.1.3 Feature Extraction

This is the most important step of the proposed methodology. In this step, proper scores will be assigned to the tokens. The assigned score will be used to find the relevance of the token in that post. The token which is appearing the most should have higher scores. Proposed model is a bi-modal structure. So, there are two methods in particular which will be used. In the method 1, Simple matching coefficient and Word Mover's Distance is used.

Method 1

Simple Matching Coefficient

This algorithm is used to increase the accuracy of the proposed system and to shorten the given data sets. Tokens of one post is compared with other tokens of the same post and a Coefficient value is assigned to each token as weights. Since posts are already normalized, repetition of a word in a single post is avoided without changing the results.

When their is occurrence a token more than once, SMC Values are calculated for that particular token using the following formula:

$$SMC\_Value = \frac{number\ of\ matching\ attributes}{number\ of\ attributes}$$

$$SMC\_Value = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

Where:
f_{11} is the total number of attributes where token A and token B both have a value of 1.
f_{01} is the total number of attributes where the attribute of token A is 0 and the attribute of token B is 1.
f_{10} is the total number of attributes where the attribute of token A is 1 and the attribute of token B is 0.
f_{00} is the total number of attributes where token A and token B both have a value of 0.

Word Mover's Distance

In machine learning, Word Mover's Distance(WMD) [3] is a promising tool where word embeddings and meaning of the words are incorporated in computing the distance between two documents. It allows to submit a query and return the most relevant documents. The pre-trained word embeddings are given,

and the dissimilarities between documents can be measured with semantic meaning by computing the minimum amount of distance that the embedded words of one document need to travel for reaching the embedded words of another.

This particular algorithm is used here to reduce the overall time complexity of the Proposed System by reducing the number of total comparisons in the next steps.

$$d = [d_1, d_2, ..., d_n]^T$$

Where:

$$d_i = \frac{c_i}{\sum_j^n c_j}$$

c_i = { Token i appears c_i times in the given document }

Gaussian mixture model

The Gaussian mixture model is the probabilistic model that will assume all the data points that will be generated from a mixture of a finite number of Gaussian distributions(with unknown parameters). It's used for representing Normally Distributed sub-populations within an overall population.

As an example, while modeling human height data, height is normally modeled as a normal distribution for each gender with a mean of approximately 5'6" for males and 5'0" for females. When we have only the height data and not the gender assignments for each data point, the distribution of all heights would follow the sum of two scaled -different variance and shifted - different mean normal distributions. This is an example of a Gaussian mixture model. The advantage of the Gaussian mixture models is that - Do not require which sub-population a data point belongs to and allow the model to automatically learn the sub-populations. It can be thought of as a generalization of k-means clustering to incorporate information about the co-variance structure of the data and the centers of the

latent Gaussian's. Gaussian mixture object implements the EM (Expectation-Maximization)algorithm to fit the mixture of Gaussian models. For multivariate models, it can draw confidence ellipsoids and later on compute the Bayesian information criterion, which will assess the number of clusters in the data. It's the fastest algorithm to learn mixture models because it will only maximize the likelihood and not bias the means towards zero. But on the contrary, when one will have insufficiently many points per mixture, it becomes difficult to estimate co-variance matrices. A Gaussian mixture model is generally parameterized by two types of values- the mixture component weights and the component means and variances/co-variances. GMMs are also used for feature extraction from speech data.

Silhouette Analysis

Silhouette analysis is used to study the separation distance between the resulting clusters. The silhouette plot will measure how close each point in one cluster is to points in the neighboring clusters. This provides a way to assess parameters like the number of clusters visually. The Silhouette Coefficient is calculated using the mean intra-cluster distance - a and the mean nearest-cluster distance- b for each sample. The silhouette score for each cluster is above average silhouette scores.

F-divergence

According to probability theory, an f-divergence is a function $D_f(P \parallel Q)$ which measures the difference between two probability distributions P and Q and helps the intuition to think of the divergence as an average, weighted by the function f, of the odds ratio given by P and Q.

Let P and Q be two probability distributions over a space Ω so that P is absolutely continuous with respect to Q. Later on, for a convex function f such that f(1) = 0, the f-divergence of P

from Q is defined as

$$D_f(P \parallel Q) \equiv \int_\Omega f\left(\frac{dP}{dQ}\right) dQ.$$

If P and Q are both absolutely continuous with respect to a reference distribution μ on Ω then their probability densities p and q satisfy dP = p dμ and dQ = q dμ. In this case the f-divergence can be written as

$$D_f(P \parallel Q) = \int_\Omega f\left(\frac{p(x)}{q(x)}\right) q(x)\, d\mu(x).$$

Word2Vec

Word2Vec is a natural language processing technique that is a family of model architectures and optimizations that can be used to learn word embeddings from large data-sets. Word2Vec uses a neural network model to learn word associations from a large text corpus. After training, this model can detect synonymous words or suggest other words for a partial sentence. So basically creates vectors that are distributed numerical representations of word features. Thus is used to extract the notion of relatedness across words or products such as semantic relatedness, synonym detection, selection preferences, concept categorization, and analogy. Word2Vec can make highly accurate guesses about a word's meaning based on past appearances without any human intervention.

4.1.4 Feature Fusion

After feature extraction, we move to feature fusion. From the previous step of Feature Extraction, we obtain tokens arranged in clusters and classified as words, POS tags, English words and stop words. In feature fusion, all the scores of the algorithms

used such as Word Mover's Distance for all the tokens and keywords are combined. Keywords will naturally have higher scores. Towards the end of this step, we get tokens arranged in clusters and classified as sentiment scores.

4.1.5 Feature Weighting

This step is done to lower the recall value because the number of features will be reduced. The accuracy will increase because only relevant scores are filtered out. Shannon entropy of each cluster is calculated. Clusters with low entropy values are passed on to the next step (this is achieved by setting a threshold value).

4.1.6 Classification

Classification is done using neural networks and the Gini index.

4.1.7 Sentiment Computation

Sentiment computation is done using SentiWordNet. Finally, we get our classified Facebook posts with their sentiment values.

4.2 Tools Used

4.2.1 SentiWordNet

SentiWordNet[1] is an opinion lexicon that is derived from the WordNet database. Each term is related to numerical scores, and it says about the positive and negative sentiment information. It is an important tool for sentiment analysis tasks.

4.2.2 Facepager Tool

Facepager[2] is an application for automated data retrieval on the web which is made for scrapping Facebook, Instagram,

YouTube, Twitter's data. Facepager can only scrape public pages for Facebook as Facebook has strict guidelines to maintain the privacy of each users. Facepager API abides by all the guidelines and is a safe way to extract data that can be used for research or business purpose. Facepager was initially free to use but from September, 2021 it has imposed some new rules on the usage of Facepager.

Chapter 5

Personal Contribution

As I'm aspiring to go for Highers, instead of trying for a company intern, I opted for research internship. During the internship period, I have learnt a lot and got the chance to redefine my past knowledge about algorithms and different aspects of software engineering and artificial intelligence.

My personal contribution for the Project & Internship-

→ For me the first phase was the learning phase. Did online course on python for data science, AI Development offered by IBM from Coursera.

→ Second Phase was deciding the project theme. I chose NLP because I have plans to do further study related to it

→ Making the road-map

→ Attended the workshop & seminar conducted in IITT over 2 months period for better understanding

→ Wrote literature review of 40+ related papers

→ Initially tried our idea using twitter posts

→ Made the list of 110 vital Facebook pages to do data scraping

→ Getting 7 lacs Facebook posts data

→ Preparing the survey questions

→ Deciding on the algorithms to be used

→ Working on the algorithm

→ Required Data extraction of Facebook posts

→ Testing the algorithms with smaller data to assess the best algorithms

→ Working on tools for text analysis

→ Working on the emotion analysis part

→ Presented our work in monthly clubhouse meeting

→ Working on Extension for Facebook

Contribution in writing Intern Report:

Wrote Abstract, Industry, Introduction and Work Done. Helped in writing and assembling other sections

Chapter 6

Results and Analysis

This is an on-going project. Till now, we have experimented with many algorithms for different steps of Feature Fusion and Feature Extraction. Accuracy obtained with the present combination of algorithms was approximately 90-94% with different sizes of data-sets. This is substantially higher than that of other research works on the same topic (though it was for other social media platforms).

Once, the choice of algorithms is finalized, we will be converting it to a Chrome Extension. The final Chrome Extension will read every post currently being displayed on the browser Tab. It will perform Sentiment Analysis on every post. Based on the result obtained from the sentiment analysis, it will display a scale on top of every post. It will be a coloured scale as shown in Figure: 6.1.

Figure 6.1: Scales

Further, any user using this extension, can filter out posts

with negative sentiment values. One can re-arrange all the posts in their feed based on the sentiment values. These features will make sure that the user gets only those posts which keep their mood joyful.

Chapter 7

Conclusion

Facebook being the largest and fastest growing site with 2.8 Billion Monthly active users, is a place where we share our opinions, thoughts and life-events, and get affected by seeing the content of it. Our mental well-being is deeply connected with social media usage. So, our initiative to assess mental well-being will be beneficial to prevent many vital issues like depression and anxiety.

This Internship concluded with many multiple take-away. I learnt and applied different NLP & Deep Learning Techniques. The project is being built under constant guidance and supervision of our Supervisor and the Mentors who were appreciative of the work done.

References

[1] Andrea Esuli. Sentiwordnet: A lexical resource for opinion mining., Mar 2019.

[2] Jakob Jünger and Till Keyling. Facepager: An application for automated data retrieval on the web., Nov 2019.

[3] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[4] Christian Montag, Alexander Markowetz, Konrad Blaszkiewicz, Ionut Andone, Bernd Lachmann, Rayna Sariyska, Boris Trendafilov, Mark Eibes, Julia Kolb, Martin Reuter, et al. Facebook usage on smartphones and gray matter volume of the nucleus accumbens. *Behavioural brain research*, 2(8):221–228, November 2017.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques, 2002.

[6] Holly B. Shakya and Nicholas A. Christakis. Association of Facebook Use With Compromised Well-Being: A Longitudinal Study. *American Journal of Epidemiology*, 185(10):203–211, 02 2017.

[7] Amir Hossein Yazdavar, Mohammad Saeid Mahdavinejad, Goonmeet Bajaj, William Romine, Amit Sheth, Amir Hassan Monadjemi, Krishnaprasad Thirunarayan, John M Meddar, Annie Myers, Jyotishman Pathak, et al. Multimodal mental health analysis in social media. *Plos one*, 15(3):e0226248, April 2020.

Plagiarism Report

Using the Plagiarism Tool of Grammarly, the Plagiarism of this whole document was found to be **6%** and the proof of it is shown in Figure: 7.1.

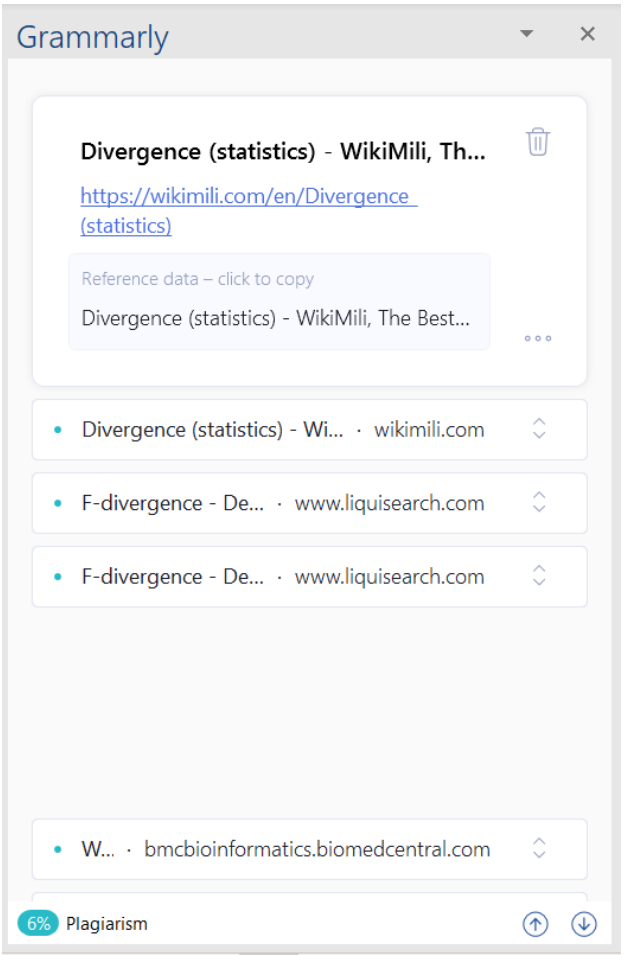


Figure 7.1: Plagiarism percentage of whole document