Comparing and evaluating tools for Sentiment Analysis

Franco M. Borrelli[1,2][0000-0002-1185-4461] and Cecilia Challiol[1,3][0000-0001-5140-0264]

[1] LIFIA, Facultad de Informática, UNLP, La Plata, Buenos Aires, Argentina
[2] UNLP Master's Scholarship, Argentina
[3] CONICET, Argentina
{fborrelli,ceciliac}@lifia.info.unlp.edu.ar

Abstract. Sentiment analysis is a process of identifying and extracting personal information from textual data. It has become essential for businesses and organizations to understand customers' opinions, emotions, and attitudes toward their products, services, or brands. While creating a custom sentiment analysis model can provide tailored results for specific datasets, it can also be time-consuming, resource-intensive, and require a high level of expertise in machine learning. Some tools offer a faster and more accessible alternative to users without a background in machine learning to create a custom model. However, researchers and practitioners usually do not know how to choose the best tool for each domain. This paper compares and evaluates some sentiment analysis tools' differences, considering how they were built and how suitable they are for analyzing sentiments on some specific topics. In particular, this paper focuses on four popular sentiment analysis tools for Python: TextBlob, Vader, Flair, and HuggingFace Transformers.

Keywords: Sentiment Analysis, TextBlob, Vader, Flair, HuggingFace Transformers, Ruled-based approach, Machine Learning.

1 Introduction

Sentiment analysis [1] is a natural language processing (NLP) technique that automatically identifies and extracts personal information from a text. This technique is used to analyze large amounts of text data, such as customer reviews, social media posts, and news articles, to determine the overall sentiment or attitude expressed in the text. Sentiment analysis has become essential for businesses, researchers, and individuals to understand how people feel about a particular topic or product [2].

There are two options for performing sentiment analysis: creating a custom model or using pre-built tools. The first can provide more tailored results for specific datasets; it can also be time-consuming, resource-intensive, and require high expertise (as knowledge in machine learning). On the other hand, some pre-trained tools that have been tested for accuracy and reliability can be used out of the box. They require minimal coding experience, making them accessible to a broader range of users who may not have a background in machine learning. However, there still needs to be documented research comparing and evaluating their effectiveness in specific areas of

interest without going into detail on how they work. In addition, the few existing studies often focus on comparing up to two of these tools [3].

This paper aims to reduce the mentioned gap by providing a comparative evaluation of popular sentiment analysis tools for Python. To do that, this paper compares and evaluates the performance of four tools on a standardized dataset and provides insights into their strengths and weaknesses to help practitioners or researchers decide which tool might be more suitable for their problem. Two evaluations on standardized datasets from different areas of interest are comparatively evaluated for these tools.

This paper is structured as follows. Section 2 describes a brief survey of the tools under analysis. Section 3 presents the results of two evaluations to determine the effectiveness of each tool. Conclusions are mentioned in Section 4.

2 Background

Sentiment analysis tools [1] have unique strengths and limitations, and understanding these differences is critical when selecting the suitable tool for a particular task. There are some critical aspects to pay attention to that condition this selection.

The first issue is the methodology used for the prediction. There are two main approaches [4]: rule-based and machine learning. The first relies on a dictionary of words labeled as: positive, negative, or neutral. A sentence is tokenized, and each token is compared with the available words (in the dictionary). Then, a combination function such as sum or average is used to make the final prediction. This method only focuses on individual words, ignoring the context in which they are used [3], so, for example, sarcasm is often misunderstood. In contrast, machine learning methods use algorithms[1] to learn from data and identify text patterns that indicate sentiment. These methods typically require large amounts of labeled data for training but can provide higher accuracy and more flexibility. Predictions are usually slower because the computational algorithm is usually much more complex. An analysis comparing the performance of both approaches is presented in [4]. Hybrid models (that combine aspects of these two approaches) can further improve the results [5].

The second critical aspect is associated with data from different domains used in each tool (such as vocabulary present in product reviews or movie reviews) to determine what is positive and negative. These differences can lead to varying levels of effectiveness depending on the specific topic being analyzed. The last issue is that some tools give a general weighting (between -1 and 1), whereas others indicate whether the result is positive or negative and the degree of confidence (from 0 to 1).

This paper focuses on four sentiment analysis tools for Python: TextBlob[2] (using its two alternatives), Vader[3], Flair[4], and HuggingFace Transformers[5]. Table 1 presents vital distinctions among each of these tools.

---

1 For example, Naïve Bayes, Support Vector Machines, or Neural Networks.
2 TextBlob, https://textblob.readthedocs.io/en/dev/, last access: 14/03/2023.
3 Vader, https://github.com/cjhutto/vaderSentiment, last access: 14/03/2023.
4 Flair, https://github.com/flairNLP/flair, last access: 14/03/2023.
5 HuggingFace Transformers, https://huggingface.co/docs, last access: 14/03/2023.

Table 1. Comparison between sentiment analysis tools.

Tool	Approach	Dataset	Scoring
TextBlob	Ruled-based	Customer/product reviews (mostly adjectives)[6]	From -1 to 1, where -1 means very negative, 1 means very positive and 0 means neutral.
Vader	Ruled-based	Multiple domains[7] (social media)	Same as TextBlob.
TextBlob + NaieveBayerAnalyzer	Machine Learning (Naïve Bayes)	Movie reviews (NLTK)[8]	Returns a value of positivity and a value of negativity. Both in ranges between 0 and 1.
Flair	Machine Learning (Embedding-based models)	Movie reviews (IMDB)[9]	Returns a label (Positive or Negative) with a score ranging from 0 (uncertain) to 1 (very certain) about the prediction.
HuggingFace Transformers	Machine Learning (Deep Learning)	Stanford Sentiment Treebankmm-sst2[10]	Same as Flair.

3 Evaluating the performance of sentiment analysis tools

In this section, two evaluations are conducted to assess the effectiveness of the four tools presented in Section 2. Two domains were selected, taking into account the datasets with which the four tools were built: films and television and opinions on public figures. Two sets of tweets in English[11] associated with these domains were downloaded and cleaned following the guidelines provided in [7]. After that, the tweets were manually labeled by a human as positive (0.3 to 1), negative (-1 to -0.3), or neutral (-0.3 to 0.3). This process could cause biases and errors. However, it provides valuable ground truth for comparing the different sentiment analysis tools. This labeling process was essential for us to have a baseline of comparison for the sentiment analysis tools being tested and to evaluate their performance and accuracy in classifying sentiments within these domains. In order to gauge the level of efficiency of each tool, the obtained results are compared with what was manually determined; they are classified into one of the following six categories presented in Table 2.

[6] TextBlob Lexicon. https://github.com/sloria/TextBlob/blob/dev/textblob/en/en-sentiment.xml, last access: 14/03/2023.

[7] Vader Lexicon, https://github.com/cjhutto/vaderSentiment/blob/master/vaderSentiment/vader_lexicon.txt, last access: 14/03/2023.

[8] NLTK Corpus Movie Reviews Dataset. https://www.kaggle.com/datasets/nltkdata/movie-review, last access: 14/03/2023.

[9] IMDB Large Movie Review Dataset. https://github.com/flairNLP/flair/blob/master/tests/resources/ tasks/imdb/README.md, last access: 14/03/2023.

[10] The Stanford Sentiment Treebank dataset. https://huggingface.co/datasets/sst2 last access: 14/03/2023.

[11] Note that some tools lack support for languages other than English.

Table 2. Categories in which the results are classified.

Category	Condition
True positive *(TP)*	when it is positive and correctly predicted as positive.
True negative *(TN)*	when it is negative and correctly predicted as negative.
True neutral *(TNL)*	when it is neutral and correctly predicted as neutral.
False positive *(FP)*	when it is not positive and incorrectly predicted as positive.
False negative *(FN)*	when it is not negative and incorrectly predicted as negative.
False neutral *(FNL)*	when it is not neutral and incorrectly predicted as neutral.

The accuracy rates are calculated by dividing the total number of correctly classified tweets (TP + TN + TNL) by the total number of tweets evaluated.

The first selected domain to evaluate the performance of sentiment analysis tools when dealing with tweets was 'Films and Television'. To accomplish this, we searched for tweets containing the term 'Wakanda Forever', corresponding to the latest Marvel movie release on Disney+. One hundred tweets were collected and then manually labeled using the before mention criteria. The dataset generated comprised 37 positive, 43 neutral, and 20 negative tweets. After that, we compared each tool's efficiency in classifying tweets according to their level of sentiment, as presented in Table 3. According to these results, TextBlob obtained the highest accuracy rate.

The second domain was 'Opinion on public figures'. We collected and manually categorized two hundred tweets with opinions about 'Elon Musk', who has been a controversial figure since he acquired Twitter; 46 were positive, 48 were neutral, and 106 were negative. As shown in Table 4, HuggingFace Transformers had the highest accuracy rate, followed by Flair.

Table 3. Results of the evaluation - Domain 'Films and Television'.

Tool	TP	TN	TNL	FP	FN	FNL	Accuracy
TextBlob	12	6	36	9	1	36	54%
Vader	15	5	24	21	4	31	44%
TextBlob+NaieveBayerAnalyzer	19	6	23	17	10	25	48%
Flair	28	12	3	34	14	9	43%
HuggingFace Transformers	34	17	0	24	25	0	51%

Table 4. Results of the evaluation - Domain 'Opinion on public figures'.

Tool	TP	TN	TNL	FP	FN	FNL	Accuracy
TextBlob	10	14	46	6	3	121	35%
Vader	20	42	44	23	4	67	53%
TextBlob+NaieveBayerAnalyzer	17	24	17	67	67	64	29%
Flair	29	88	3	24	39	17	60%
HuggingFace Transformers	33	97	0	25	45	0	65%

4 Conclusions

Four tools for sentiment analysis were compared to show how it is possible to use them without training a custom model. Each tool has its peculiarities; some were developed on a rule-based approach (TextBlob, Vader), while others use machine learning techniques (such as TextBlob+NaieveBayerAnalyzer, Flair, HuggingFace Transformers). Tables 3 and 4 allow observing that the approach used for each tool does not impact the accuracy. For example, TextBlob obtained 54% versus HuggingFace Transformers, with 51% in Table 3. Generally, machine learning techniques are considered less efficient because they assign word-by-word weighting without considering the message context [3]. Additionally, two tools that use the same approach may have widely different accuracies, such as TextBlob (35%) and Vader (53%) in Table 4.

On another side, some tools use domain-specific datasets, and others use multi-domain or general-purpose data. However, this is not a conditional aspect of the tool's accuracy. For example, for the first evaluation focused on film and television, TextBlob scored the highest accuracy (54%) in Table 3, despite using a lexicon from a different domain.

In both evaluations, the percentages of correct answers were relatively low, reaching a maximum of 65%. The manual labeling of tweets was subjective, and the datasets used for the evaluations were relatively small (100 and 200), so more evaluations are required to define a conclusion.

We expect this paper will contribute to discussing how practitioners or researchers could choose the correct sentiment analysis tools without requiring expert knowledge.

References

1. D'Andrea A., Ferri F., Grifoni P., Guzzo T.: Approaches, Tools and Applications for Sentiment Analysis Implementation. International Journal of Computer Applications 125(3), 26-33 (2016).
2. Rambocas M., Gama, J.: Marketing Research: The Role of sentiment Analysis. Universidade do Porto, Faculdade de Economia do Porto, paper 489 (2013).
3. Urmita M., Dhanraj V.: Sentiment Analysis of Facebook Using Textblob and Vader. Journal of Innovative Engineering and Research 4(1), 10-14 (2021).
4. Srivastava R., Bharti P.K., Verma P.: Comparative Analysis of Lexicon and Machine Learning Approach for Sentiment Analysis. International Journal of Advanced Computer Science and Applications 13(3) (2022).
5. Mahmood A., Kamaruddin S., Naser R., Mohd Nadzi M.: A Combination of Lexicon and Machine Learning Approaches for Sentiment Analysis on Facebook. Journal of System and Management Sciences 10(3), 140-150 (2020).
6. Ray P., Chakrabarti A.: A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis. Applied Computing and Informatics 18(1/2), 163-178 (2022).
7. Akbik A., Blythe D., Vollgraf R.: Contextual String Embeddings for Sequence Labeling. In: Proceedings of the 27th international conference on computational linguistics, pp. 1638-1649, Association for Computational Linguistics, USA (2018).