Tipo de Artigo: Original

CLASSIFICATION OF FEELINGS EXPRESSED IN TEXTS ON SOCIAL NETWORKS THROUGH NATURAL LANGUAGE PROCESSING TECHNIQUES

CLASSIFICAÇÃO DE SENTIMENTOS EXPRESSOS EM TEXTOS NAS REDES SOCIAIS ATRAVÉS DE TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL

Recebido em: 06/03/2024

Aprovado em: 25/05/2024

Arthur Avelino Pereira Ferreira (Orcid: https://orcid.org/0009-0008-3048-2190)
Especialista em Ciência de Dados. Centro Universitário UNDB. São Luís, Maranhão, Brasil.

Allan Kássio Beckman Soares da Cruz (Orcid: https://orcid.org/0000-0002-2631-2032)
Doutorando em Ciência da Computação. Programa de Pós-Graduação Doutorado em Ciência da Computação - Associação UFMA-UFPI. São Luís, Maranhão, Brasil.

Pamela Torres Maia Beckman da Cruz (Orcid: https://orcid.org/0000-0002-9147-6682)
Doutoranda em Ciência da Informação. Faculdade de Letras – Universidade de Coimbra. Coimbra, Portugal.

Mario Meireles Teixeira (Orcid: https://orcid.org/0000-0001-8771-1478)
Doutor em Ciência da Computação. Programa de Pós-Graduação Doutorado em Ciência da Computação - Associação UFMA-UFPI. São Luís, Maranhão, Brasil.

Carlos de Salles Soares Neto (Orcid: https://orcid.org/0000-0002-6800-1881)
Doutor em Ciência da Computação. Programa de Pós-Graduação Doutorado em Ciência da Computação - Associação UFMA-UFPI. São Luís, Maranhão, Brasil.

DOI: https://doi.org/10.5281/zenodo.14231708

Autor para correspondência:

Nome Allan Kássio Beckman Soares da Cruz
Endereço: Av. dos Portugueses, 1966 - Vila Bacanga, São Luís - MA, 65080-805, Brasil
E-mail: allankassio@gmail.com

ABSTRACT

To understand how the application of the main NLP techniques improves the performance of a model, a database of more than 280 thousand records was compiled, since there is no good basis for training and testing natural language processing in Portuguese. In Brazil, most of the material is in English or has been automatically translated. Therefore, the model was trained without applying any technique and then each technique was applied, recording the results of each technique, in order to compare all the techniques at the end and understand how the performance gain was at each stage. After performing all the techniques, it became clear that NLP is essential for working with any type of text in data science, because in our model we were able to increase the accuracy by 19.47%.

Keywords: Natural Language Processing. Feelings Classification. Social Networks.

RESUMO

Para entender o quanto as aplicações das principais técnicas de NLP irão de fato melhorar a performance de um modelo, foi coletada uma base de dados com mais de 280 mil registros, pois não existem boas bases para treino e teste de processamento de linguagem natural em português brasileiro, a grande maioria do material está em inglês ou foi traduzida de forma automática. Para tanto, foi treinado o modelo sem aplicação de nenhuma técnica e depois foram aplicadas cada uma das técnicas, sempre guardando os resultados de cada uma, para no final, comparar todas elas e entender como foi o ganho de performance em cada etapa. Após realizar todas as técnicas ficou claro que a NLP é fundamental para se trabalhar com qualquer tipo de texto em Ciência de Dados, pois no nosso modelo conseguimos aumentar a acurácia em 19,47%.

Palavras-chave: Processamento de Linguagem Natural. Classificação de Sentimentos. Redes Sociais.

INTRODUCTION

Maintaining your company's image on social media is no longer just a matter of marketing, but a matter of survival in the marketplace. It is important to appeal to your customers, to position yourself in relation to political, ethnic and social issues, and to have a good reputation in the public eye. The cancelation culture that is so much discussed today, where a celebrity who commits a discriminatory act loses multiple sponsorships and contracts due to public pressure on the company, supports the person who has been "canceled."

But how can companies monitor this gigantic amount of data generated every day? For example, according to Ahlgren (2021), Twitter records more than 500 million messages per day, and there are more than 330 million monthly active users. The answer to this question is artificial intelligence, which uses technology to capture what people are feeling on social media based on their posts, comments, tweets, captions, and any other form of textual interaction.

To understand the concept of artificial intelligence, one must first understand the concept of intelligence. According to Russel *et al.* (2021), "people are intelligent to the extent that their actions are capable of achieving their goals." That is, all aspects can be evaluated according to their ability to achieve the defined goals. For artificial intelligence, the concept could be the same, except that humans are replaced by machines.

The difference will be that the machine does not have its own goals, but we humans are responsible for assigning them. The machine learning process is very similar to that of a human (Bell, 2022). How can we tell a dog from a duck? If you have never seen a dog or a duck, it will be impossible to tell them apart. You can only do that if you have seen several dogs and several ducks and know the characteristics of each animal. It is the same with machine learning. You need to present historical data so that the computer can perform statistical calculations and be able to find patterns and features and thereby learn and achieve its goal.

One of the areas of artificial intelligence that has grown rapidly in recent years is natural language processing (Klyuchnikov *et al.*, 2022). According to Bird *et al.* (2009), "Natural language processing, in its broadest sense, includes any kind of manipulation of natural language by computers. At one extreme, it can be as simple as counting word frequencies to compare different writing styles. At the other extreme, NLP is about fully understanding human expressions, at least to the point of being able to make useful responses to them".

We will soon understand that natural language or human language, such as Portuguese, English, Spanish and other languages, cannot be understood by the computer without manipulating these texts (Pirc, 2022). The computer understands only binary language, so natural language processing is exactly the treatment to make human language understandable to the computer.

The main objective of the developed work is to understand the main NLP techniques. For this purpose, a Twitter database was created with more than 280 thousand records. To create this database, a web scraper was developed using the Python programming language, capable of extracting tweets from the official platform based on a topic. Several topics were selected where there were mainly negative tweets and positive tweets, and these were then classified.

The dataset of 280 thousand records was matched between positive and negative sentiments to ensure the quality of the natural language processing database. The idea of the project is to use all these datasets and train them without any interference or processing. natural language.

At the end, we get feedback on the success metrics of the algorithm, such as accuracy, f1 score, recall, precision, and others (Wang, 2022). Then, NLP techniques are applied and the model is re-trained with each of these techniques to understand if there was no improvement in the indicators or if there was and how big this improvement was. Techniques such as tokenization, removal of stop words, removal of accents and numbers, stemmer, TF-IDF, NGrams, and bag of words are applied.

RELATED WORKS

Sentiment analysis on social media has garnered significant research attention due to the proliferation of user-generated content and its impact on various domains. The following works highlight recent advancements and challenges in this field.

Jim *et al.* (2024) explored the application of deep learning techniques for sentiment analysis, focusing on the improvements in accuracy and scalability. Their work highlights the shift from traditional machine learning algorithms to more complex neural networks, which offer superior performance in understanding and classifying sentiments in text.

Islam *et al.* (2024) provided a comprehensive review of deep learning methods used in sentiment analysis. They discussed various applications, including product recommendation and opinion mining, and introduced a novel hybrid approach combining multiple deep learning models to enhance sentiment classification accuracy.

Hartmann *et al.* (2023) analyzed the effectiveness of sentiment analysis in marketing research, emphasizing its role in understanding consumer behavior. Their study showcased how advanced NLP techniques can be applied to large datasets to extract valuable insights for business strategies and customer engagement.

Ahmet and Abdullah (2020) examined the challenges and future directions in sentiment analysis, particularly focusing on the limitations of current deep learning approaches. The paper provided a detailed analysis of the gaps in existing methods and proposed new directions for research to address these challenges.

Qian *et al.* (2023) developed a sentiment knowledge-enhanced self-supervised learning model for multimodal sentiment analysis. This approach leverages both textual and visual data to improve the accuracy of sentiment predictions, highlighting the growing importance of integrating multiple data sources in sentiment analysis.

Gandhi *et al.* (2023) conducted a systematic review of multimodal sentiment analysis, presenting the history, datasets, and advancements in the field. Their review emphasized the benefits of combining text, audio, and visual inputs to create more robust and accurate sentiment analysis models.

EXPLORATORY DATA COLLECTION AND ANALYSIS

Natural language processing is a much discussed and studied topic worldwide, but it is difficult to find databases that are not in English. In the elaboration of this project, the first step was to look for a ready database. However, all the databases found in Portuguese were either directly translated from English or they were very small databases, which made a good analysis impossible.

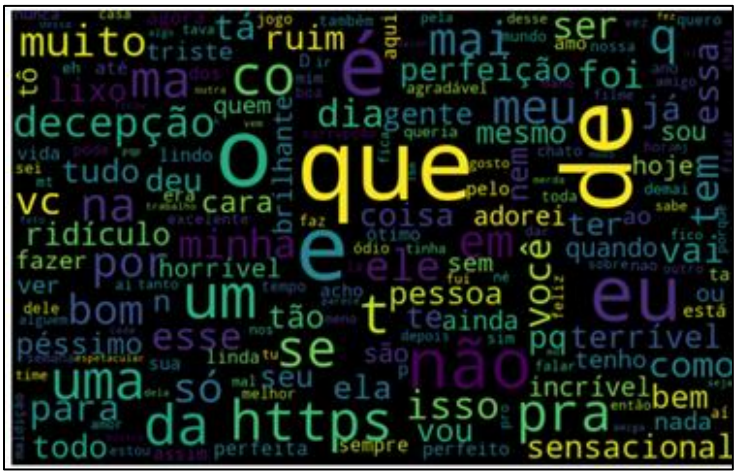
Another goal of this work is to leave a legacy for the Brazilian data science community by creating a large database of tweets in Portuguese collected with a WebScrapper developed in Python. About 280 thousand tweets were collected and divided into negative and positive sentiments.

The following data were collected: "date", i.e. the date when the tweet was written, "tweet", i.e. the text itself and also our object of study, "language", i.e. the language to ensure that all tweets are in Portuguese, and "sentiment", i.e. the classification of our text, where the value 1 represents a positive sentiment and the value 0 represents a negative sentiment.

At the beginning of an exploratory analysis, since we are dealing with textual data, a word cloud was created. This is a graph that shows the most frequently occurring words in the dataset. To create this cloud, you must first understand the concept of a "corpus" and how to create one. The most common definition of this important component of natural language processing is that a corpus is a collection of natural language, which in our case means that all tweets are grouped together in one place. Python's NLTK library makes it easy to perform this process.

This is a simple way to create a corpus. After processing the dataset, it was possible to create the word cloud shown in Figure 1 to facilitate visual analysis of the unique words found.

Figure 1 - Word Cloud



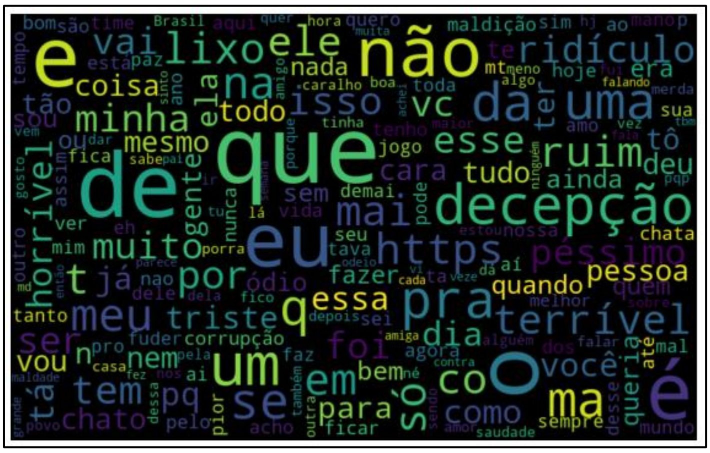
Font: Created by the authors.

On the surface, the diagram in Figure 1 does not offer much insight. It merely shows that we have most of the different kinds of words in our corpus. To understand if there is a pattern, we also clustered by type of emotion and created a cloud of words specific to each emotion.

Another thing we can see is that there are the same words treated as different words, for example: "perfect" and "perfection" both have basically the same meaning, but they have different radicals, the machine cannot understand that these words have the same meaning. Same origin, but through NLP it is possible to treat these cases and leave the words with only their root, for example: "bonito e bonita", if you leave out the root it becomes "bonit" and the machine will unify all these words and understand that they have the same meaning.

After clustering, it is already possible to get a better picture of each corpus. Figure 2 shows the most important negative words. We see some words like "terrible", "disappointing" or "bad", but also many words that do not express any feelings, like "que, não, de, eu, pra, só" (what, not, of, I, for, only), which occur frequently in sentences but do not contribute anything to our study.

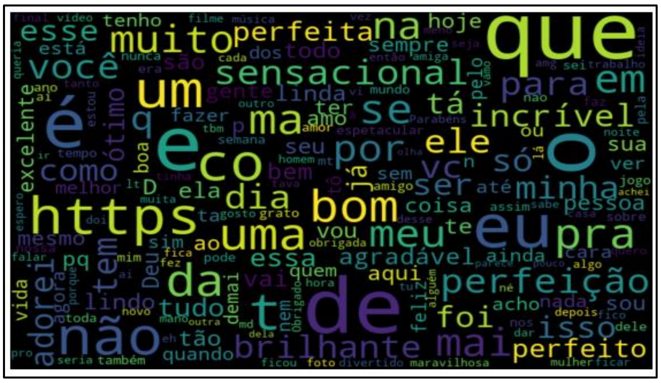
Figure 2 - Cloud of negative words



Font: Created by the authors.

In the cloud of positive words in Figure 3, we can see some expressions such as "brilliant," "enjoy," "beautiful," "good," "perfection" that actually convey a positive feeling. However, as in Figure 2, there are also words among the negative words that are irrelevant to our study, and these words are very numerous in our corpus, so we will have to deal with them later. Since this seems to be a word frequency issue in our dataset, take a closer look at how often they occur.

Figure 3 - Cloud of positive words.



Font: Created by the authors.

At this moment, the model has a serious problem with words that are irrelevant and that we cannot tell whether they are part of a positive or negative sentiment. We do not have a single word in the top 10 frequencies that is relevant. Through this exploratory analysis, we have a direction for the kind of data processing we need to do.

TRAINING THE MODEL WITHOUT ANY NATURAL LANGUAGE PROCESSING

One of the main objectives of the study is to find out to what extent the application of each of the main NLP techniques affects the results. Therefore, in order to have this computational basis, it is first necessary to train the model without any treatment. For this purpose, we have chosen an algorithm commonly used for working with texts, the Naive Bayes Multinominal NB (Joshi and Abdelfattah, 2021).

Before performing the training, it is important to emphasize that the machine cannot read texts, only numbers, and therefore it is essential to perform this treatment before training the model. This process is called vectorization and works as shown in Table 1.

Table 1 - Word vectorization example.

Words	NLP	IS	GOOD	BAD	Feeling
Text 1	1	1	1	0	Negative
Text 2	1	1	0	1	Positive

Font: Created by the authors.

The vectorizer proceeds exactly as in Table 1, using the corpus to check which words are present and then creating an array of the total number of words that each text in the dataset contains, so that each text becomes the following array:

Text 1 = [1,1,1,0]

Text 2 = [1,1,0,1]

It is important to note that there are two different arrays representing each of the texts. Now that the text has been converted to numbers, the model can be trained. To facilitate the vectorization of a large set of documents, the CountVectorizer package from the Sklearn library was used and the Naive Bayes MultinomialNB classifier, also from Sklearn (Khan, 2021), was used to train the model.

The model, which does not use natural language processing techniques, achieves 64% accuracy, i.e., it makes 6 classifications for every 100 texts. This percentage of correct answers is very low and, in a context where a company is working on its digital marketing, such a large number of errors can be fatal in providing the necessary answers and guaranteeing the good position of the brand in the market. So, to improve this index, we have to work on it.

MAIN NATURAL LANGUAGE PROCESSING TECHNIQUES

In the exploratory analysis conducted, the main problem identified was the large amount of irrelevant words in the dataset, i.e., words that were present in large quantities for both positive and negative sentiments. For this reason, when counting the difference between negative and positive words, the algorithm may not prioritize the correct words when creating the word collection and the final result may contain an error due to the large number of similar words between the two classifications.

To remove irrelevant words, the NLTK package, which is the main natural language processing package in Python today, already has a special function for this. In the PT-BR language, we will also take the opportunity to remove not only irrelevant words, but also punctuation marks and words that are commonly used on Twitter, such as "https", emoticons, "q" and others.

The nltk.corpus.stopwords.words function was used to create a list of stopwords in Portuguese, and then we called the WordPunctTokenizer() function, which is used to create a list of scores (Wen et al., 2021). Additionally, we create a manual list of frequent words on Twitter, as these were not included in the stopword list, but we found through our exploratory analysis that they were present in both datasets. We then iterate over all texts and create a new column with the processed set.

The phrase 'Com certeza, fiz a melhor escolha! Estou muito feliz!!!' became 'Com certeza fiz melhor escolha Estou feliz'. For us humans, of course, the first sentence makes more sense, but we must remember that the machine learns and recognizes patterns differently than we humans do, and the second sentence without stop words and punctuation will be clearer for this learning.

Using a simple technique of removing punctuation and stop words, we have already increased the accuracy of the model by more than 12%. However, there are still many techniques that we need to apply. The next one will also be very simple, since we are working with a Twitter dataset where many people use colloquialisms. It is possible that we will find the same words, but some with accent and some without accent, for example "péssimo" and "pessimo". For humans it is easy to recognize that they are the same word, but for the computer the accent makes the difference.

The first step to remove accents is to import the library “unidecode”. This library contains a list of all specific codes by language, so to remove we only need to iterate over the column to be processed, i.e. the new column created by removing the stop words.

After that we perform the same process in our list of stop words, because there are accentuations there as well, after that we simply repeat the same process and create a new column with the result of the newly edited sentence, shortly after that we apply again the function created for training the model. The model has reached 1% accuracy with another very simple and easy to use technique. Even if it seems like a small thing, it can make a big difference in a large data set.

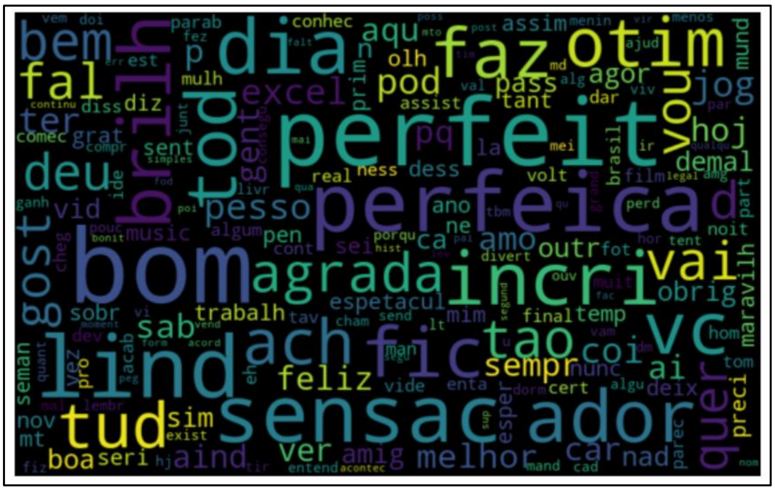
Another point that we noticed is that there are words that start with a capital letter and others with a lower-case letter or even just capital letters, for example: "Feliz", "feliz", again and with accentuation the machine will understand these words quite differently. Another simple NLP technique that can be applied is to convert all the words in our corpus to upper or lower case. We chose to use lowercase for the entire dataset.

To do this, it was sufficient to iterate over the entire corpus using the ".lower()" function and create a new column in our data frame. After that, we trained the model and were able to increase the accuracy by 4%, which already guarantees us a great result.

Another widely used technique is stem reading. This technique consists in removing the morphological suffixes of the words and reducing them to their main stem (Alshalabi et al., 2022), example: "maravilhosas", "maravilhoso", "maravilha", we know that these words have the same meaning, but for the computer they are three completely different words, so with the stem reading technique we get the following result: "maravilh", "maravilh" and "maravilh". So we have three words with the same meaning and three identical words that are read correctly by the computer.

The Stemmer technique was used, but as you can see, the gain was lower than expected. We went from 81.33% accuracy to 81.86% accuracy, a gain of 0.53%. To better understand if the Stemmer technique works, the word cloud diagram, as shown in Figure 4, was applied to the words processed with this technique and the result was analyzed.

Figure 4 - Cloud of positive words with Stemmer



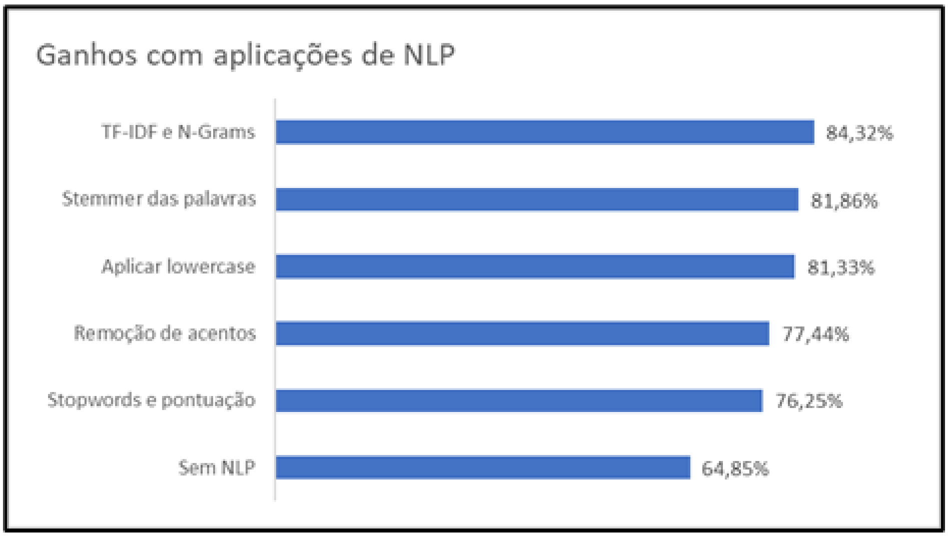
Font: Created by the authors.

Looking at the word cloud, it is clear that the technique worked, with the exception of "perfeit" and "perfeica", which should be the same radical, but also words like "sensac", which covers all its variables, "incri", which comes from unbelievable , unbelievable and other possibilities, i.e. the technique was applied correctly and gave the desired result for the corpus, but the gain in accuracy in the model was small, probably because the database is so large that the model had already learned with different radicals. In this case, perhaps the gain was in performance, because when we reduced the radicals, we also reduced the total number of unique words, i.e., we reduced the size of our corpus, and this guaranteed that the model would have a smaller number of words for iteration when trained.

To create the bag of words, we used CountVectorizer, a text vectorizer, at the very beginning of the study. However, there is a more advanced technique, TF-IDF (term frequency-inverse document frequency), which, in short, works similarly, but instead of simply counting the words, it applies a weighting depending on how often they occur (Santos *et al.*, 2022), so that some words are more relevant in the classification of the model.

In addition, you can also apply the N- GRAMS to TF-IDF in a separate function consisting of a sequence of specific words. This algorithm detects if there are words that are always close to each other (Yalcin, 2022). For example, "very good" in this case, since it consists of two words, is considered as bigram, in the case of "hello, how are you", since it consists of three words, it is considered as trigram and also as N-gram. This algorithm calculates the probability of co-occurrence of the words and if it is high, it is considered as a single word and gets more weight in the classification of the model. By using these techniques, we were able to improve our result significantly. We went from 81% accuracy to 84%, which means that applying the TF-IDF and N-gram techniques to our model was actually superior to Stemmer. Figure 5 shows the comparison between the techniques.

Figure 5 - Accuracy of each NLP technique.



Font: Created by the authors.

We can say that the application of NLP techniques in our database has led to a very meaningful result. We have increased the accuracy of our model by more than 19% just by processing the data, considering that we use the same machine learning algorithm for all techniques, namely Naive Bayes MultinomialNB.

We also used exactly the same "random_state" to ensure that the training and testing base was exactly the same data for all models. So the improvement is really due to the techniques used. In addition to the 84% accuracy, we also had 82% accuracy, 87% recall, and 91% specificity in our final model.

MODEL VALIDATION

Once the algorithm is ready, it is important to validate it to see if there is no overfitting or underfitting. To do this, you need to present new data that the model has never seen, either in testing or training. So, the idea is to use the same web scraper that was developed and extract data about a company. The company [SUPRESSED] is used as validation for this data.

Acquisition occurred in November 2021, and 334 tweets were acquired via [SUPRESSED]. Then, the same NLP treatments of our model were applied, because since our model learned the data without stop words, without emphasis, in lowercase, with stemmer and all other treatments, it can only make predictions if these data have the same format.

After performing data prediction, all tweets were analyzed and manually tagged. When correctly classified and compared between what the model predicted and what the sentiment was, it was found that out of the 238 tweets, the model hit 201 and 27 records were wrong, which is about 84% accuracy. This proves that there is neither over-fitting nor under-fitting, as our model behaved as expected on data it had never seen before.

CONCLUSION

Working with natural language processing requires extensive study, deep knowledge of linguistics, and a great deal of dedication. However, as this work shows, it is worth it because our model improved its accuracy by almost 20% using only a few techniques. The same algorithm was used for all the training, i.e. the same "random state", to ensure that the only difference between them is really the treatment of the text. To increase the work further, one can even try to apply ensemble methods or other algorithms, even a neural network using the same base and NLP treatments to get an even better result. However, an accuracy of 84% is quite acceptable and can serve as a basis for working with this algorithm.

Some techniques, Stemmer in particular, were quite surprising and did not yield much gain in accuracy. However, we know that even without this gain, the NLP technique greatly helps our model in terms of corpus size and strength and is therefore essential for our study. Moreover, simpler techniques such as stop word removal showed incredible results, proving that the best techniques are not the most complex or modern ones, but the ones that best adapt to your type of text, in our case to texts whose language is very informal, full of slang and irrelevant words.

Using a developed and available web scraper, the validation of the model was performed through a real test extracting a base of posts about [SUPRESSED], using the state of Maranhão and the word "[SUPRESSED]" as parameters. After collecting more than 250 records, we saved all these tweets in a CSV file and imported them into Python. There, we performed the same treatment as for our training texts and applied it to the prediction with our model. Once we received the output of the predictions, we exported them to Excel format and analyzed them manually, tweet by tweet, to count whether the model had hit or missed the prediction. After counting, it was determined that the model did not suffer from either overfitting or underfitting, as it exhibited accuracy very similar to that of training with data it had never seen before.

Sentiment Analyzer is a powerful tool that can be used by businesses, digital influencers and anyone who wants to understand how they are perceived on social networks. It also has the competitive advantage of always being responsive and able to fend off any comment that could cause more damage if it spreads too quickly.

REFERENCES

AHLGREN, M. **Mais de 40 estatísticas do Twitter de 2022:** estatísticas, dados demográficos do usuário e fatos. Disponível em: https://www.websiterating.com/pt/research/twitter-statistics/. Acesso em: 30 maio 2022.

AHMET, Ahmed; ABDULLAH, Tariq. Recent trends and advances in deep learning-based sentiment analysis. *In:* AGARWAL, Basant *et al.* **Deep learning-based approaches for sentiment analysis**. New York, 2020. p. 33-56.

ALSHALABI, H.; TIUN, S.; OMAR, N.; ANAAM, E. A.; SAIF, Y. BPR algorithm: New broken plural rules for an Arabic stemmer. **Egyptian Inf Journal.** n. 3, p. 363-371, 2022.

BELL, J. What is machine learning? *In:* CARTA, Silvio. **Machine learning and the city:** applications in architecture and urban design. New Jersey: Wiley, 2022. p. 207-216.

BENGFORT, B.; BILBRO, R.; OJEDA, T. **Applied text analysis with Python:** enabling language-aware data products with machine learning. Newton: O'Reilly Media, 2018.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural language processing with Python:** analyzing text with the natural language Toolkit. Newton: O'Reilly Media, 2009.

GANDHI, Ankita *et al.* Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. **Information Fusion,** v. 91, p. 424-444, mar. 2023.

HARTMANN, Jochen *et al.* More than a feeling: accuracy and application of sentiment analysis. **International Journal of Research in Marketing,** v. 40. p. 75-87, 2023.

ISLAM, Md Shofiqul *et al.* Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach. **Artificial Intelligence Review,** v. 57, p. 62. 2024.

JIM, Jamin Rahman *et al.* Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review. **Natural Language Processing Journal,** v. 6, 100059, mar. 2024.

JOSHI, S.; ABDELFATTAH, E. Multi-class text classification using machine learning models for online drug reviews. *In:* **2021 IEEE World AI IoT Congress (AIIoT).** 2021. p. 0262–0267. doi:10.1109/AIIoT52608.2021.9454250.

KEDIA, A.; RASU, M. **Hands-on Python Natural Language Processing**: explore tools and techniques to analyze and process text with a view to building real-world NLP applications. Birmingham: Packt Publishing, 2020.

KHAN, M. B. Urdu news classification using application of machine learning algorithms on news headline. **International Journal of Computer Science and Network Security,** v. 21, n. 2, p. 229–237, 2021. Disponível em: https://doi.org/10.22937/IJCSNS.2021.21.2.27.

KLYUCHNIKOV, N. *et al.* NAS-Bench-NLP: neural architecture search benchmark for natural language processing. **IEEE Access**, v. 10, p. 45736-45747, 2022.

PIRC, K. *et al.* An oomycete NLP cytolysin forms transient small pores in lipid membranes. **Science advances**, v. 8, n. 10, 2022.

QIAN, Fan, *et al.* Sentiment knowledge enhanced self-supervised learning for multimodal sentiment analysis. In: **Findings of the Association for Computational Linguistics:** ACL 2023. 2023. p. 12966-12978.

RUSSELL, S.; VARGAS, B.; VALADARES, M. **Inteligência artificial a nosso favor**: Como manter o controle sobre a tecnologia. São Paulo: Companhia das Letras, 2021.

SANTOS, A. R. S. D.; RODRIGUES, C. M. D. O.; MELO, H. B. S. D. Identifying Xenophobia in Twitter Posts Using Support Vector Machine with TF/IDF Strategy. In: **XVIII Brazilian Symposium on Information Systems,** maio 2022.

WANG, Y. *et al.* prPred-DRLF: Plant R protein predictor using deep representation learning features. **Proteomics,** v. 22, n. 1-2, 2100161, 2022.

WEN, W. *et al.* API Recommendation Based on wII-wMd. **International Journal of Cognitive Informatics and Natural Intelligence (IJCINI),** v. 15, n. 4, p. 1-20, 2021.

YALCIN, K.; CICEKLI, I.; ERCAN, G. An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding. **Expert Systems with Applications,** v. 197, 116677, 2022.

Conflict of interest

We declare that there was no conflict of interest.