Assessing the Sentiment and Value of Product Reviews in Brazilian Portuguese

Aniela Walczak[1], Piotr Kondratowicz[1], Rafał Jaworski[1]
Adam Mickiewicz University in Poznań, Poland
{aniwal4,piotr.kondratowicz,rjawor}@amu.edu.pl

Abstract

This paper presents work on automatic evaluation of written product reviews in Brazilian Portuguese. As a standard procedure, the reviews were classified with respect to the sentiment of the author. The novelty of the approach consisted in an additional evaluation of the value that a certain product review might bring to potential buyers of the product. The problem was challenging from the research perspective as the processed language was Brazilian Portuguese – a medium resourced language. Furthermore, its colloquial variety, often used by users providing product reviews, tends to express sentiment in an obscure manner. Despite these difficulties, the results obtained in experimental scenarios were satisfactory.

Keywords: sentiment analysis, llm, Portuguese, product reviews

1. Introduction

The scope of this research requires the processing of Portuguese which presents specific challenges. First and foremost, Portuguese is a language with rich inflectional morphology. Verbs are inflected for tense, number, person, and mood, while nouns and adjectives are inflected for number and gender. Therefore, a good lemmatizer or stemmer is required to reduce words to their base or contracted forms.

Another challenge is the regional variation of the language, or dialect. There are two main varieties of Portuguese: European Portuguese (PT-PT) and Brazilian Portuguese (PT-BR). European Portuguese will be called Portuguese, and Brazilian Portuguese will be called Brazilian. They differ in vocabulary and grammar. For example, the word for "refrigerator" in Portuguese is "frigorífico" and in Brazilian Portuguese it is "geladeira."

Collecting a dataset for developing the models proved to be a significant challenge. Few datasets containing real opinions in Portuguese are available online. Even if they do exist, they are usually too small to create a high-quality model. However, we managed to find one dataset that adequately meets these requirements. This is "B2W-Reviews01 - A Brazilian Portuguese reviews corpus" [Real et al.2019] which will be referred to as B2W. It consists of over 130,000 user reviews collected from the website Americanas.com between January and May 2018. All reviews are written in Brazilian Portuguese. B2W contains multiple columns, including information about the user, their gender, age, and location, as well as their reviews and ratings. Only the review column was used to develop the rating algorithm, as one of the goals of this algorithm was to determine the value of reviews based solely on the input text.

2. Sentiment analysis

2.1. Features of the dataset

The dataset is characterized by an uneven distribution of classes – positive comments constitute the largest group (79 316 comments), while neutral (16 010) and negative (33 772) comments are much less frequent. This pattern can affect the performance of classification models, leading to their bias towards more numerous classes. In particular, it can result in poor classification performance for less well-represented classes. To ensure that the assessment of model quality is not limited to overall accuracy, further analysis also includes metrics such as precision, recall and F1 score calculated separately for each class, as well as their arithmetic means and weighted means.

Comments in the dataset are utilitarian and informal in nature, containing colloquialisms, abbreviations, and linguistic errors. There are also deviations from grammatical and punctuation norms.

These features can affect classification results, especially for models sensitive to the quality and structure of text data. Furthermore, the assignment of sentiment labels was based solely on the numerical rating values, which does not take into account the linguistic context of the comment or potential irony or contradiction between the content and the rating.

The dataset served as the source for creating training and test datasets used in further experiments. For each model, the data was split 80% to 20%, with the split considering the class distribution (stratification) to ensure representativeness in both parts. The training dataset was used to adjust model parameters, while the test dataset was used solely to evaluate their performance. This split method allows for objective validation of results and a controlled and repeatable comparison of the effectiveness of different approaches.

2.2. Naive Bayes

The experiment used the MultinomialNB classifier from the scikit-learn library [Pedregosa, 2011]. This model is based on the assumption of conditional feature independence (the so-called naive assumption), meaning that each word in a document is treated as an independent feature with respect to the others. Despite this simplification, this algorithm can be effective in text classification tasks, especially when the

Aniela Walczak, Piotr Kondratowicz, Rafał Jaworski, *Assessing the Sentiment and Value of Product Reviews in Brazilian Portuguese*. In: Z. Vetulani, P. Paroubek, I. Kardava (Eds.), Human Language Technologies as a Challenge for Computer Science and Linguistics – 2025, pp. 93–97. Poznań: Adam Mickiewicz University Press, 2025. © Authors, 2025. https://doi.org/10.14746/amup.9788323245414.17. Open Access chapter, distributed under the terms of the CC licence (BY-NC--ND, https://creativecommons.org/licenses/by-nc-nd/4.0/).

input data is large and represented in a bag-of-words fashion.

The model was trained and evaluated using the metrics of precision, recall, F1-score and accuracy. The results indicate high classifier performance for the positive and negative classes. The positive class achieved an F1-score of 0.88, while the negative class achieved 0.82. However, for the neutral class, the F1-score was only 0.10, which was due to the low recall value (0.06). This means that the model recognized examples of this class to a very limited extent. Despite the high accuracy score (0.81) for the entire validation set, it should be emphasized that this value is not sufficient to assess the quality of the classifier in the case of a multiclass problem with an uneven data distribution. Analysis of class-level metrics shows that the model focuses mainly on the positive class, which is the most numerous. The high sensitivity (0.95) and precision (0.83) scores for this class may result from its dominance in the training set.

2.3. TF-IDF + SVM

The second approach used in the experiment was a classifier based on the linear version of the support vector machine – LinearSVC from the scikit-learn library. SVM (Support Vector Machine, [Cortes, 1995]) is a classification method that aims to find the optimal hyperplane separating classes in feature space. The LinearSVC class implements a linear version of this algorithm, optimized for processing large text datasets.

The model achieved an overall accuracy of 83%, with a weighted average F1-score of 0.80. The high weighted average F1-score is partly due to the dominance of the positive class in the dataset. At the same time, the arithmetic average F1-score of 0.67 indicates that the model's performance is not uniform across all classes. The difference between the weighted average and the arithmetic average confirms the limited performance of the classifier for smaller classes.

Based on the obtained results, the TF-IDF + SVM model performs better than the Naive Bayes classifier in recognizing the neutral class but difficulties remain.

2.4. LSTM

The experiment used a Long Short-Term Memory (LSTM, [Hochreiter, 1997]) neural model implemented in the PyTorch library. This architecture belongs to the class of recurrent neural networks (RNNs) and was designed to model dependencies in data sequences, such as text. Unlike simple RNNs, LSTMs can better store and process information in longer contexts, which is particularly important in sentiment analysis tasks, where the meaning of words often depends on their position in the sentence.

The LSTM model achieved the highest classification metrics for the positive class with an f1-score of 0.90 and a recall of 0.93. Good results were also obtained for the negative class with an f1-score of 0.86 and a precision of 0.82. However, significantly lower results were observed for the neutral class with an f1-score of only 0.26 and a recall of 0.20. This disparity was also observed for other tested models and may be a consequence of both the smaller size of the neutral class in the dataset and the difficulty in unambiguously interpreting such comments.

Although the LSTM architecture did not use any pre-trained text representations (such as Word2Vec, FastText, or BERT) and the embeddings were learned from scratch from the training data, the model achieved competitive results compared to classical TF-IDF-based methods. The LSTM's ability to capture local and sequential dependencies between words contributed to its effectiveness, especially with larger data sets.

Compared to classical models (Naive Bayes, SVM), the LSTM model achieved higher classification quality metrics for the dominant classes. This suggests that even a simple architecture based on recurrent neural networks can provide better results in scenarios with large amounts of data and class imbalance, provided that the input data is properly processed and the appropriate division into training and test sets is applied.

2.5. BERT

The experiment used the *BertForSequenceClassification* model provided by *HuggingFace*, trained for the Portuguese language, neuralmind/bert-base-portuguese-cased [Souza, 2020]. This model is part of the Transformer architecture, which relies on attention and allows for efficient modeling of dependencies in text sequences without recursion. This allows BERT to consider both left- and right-hand context, resulting in high-quality semantic representations.

The input data was transformed using the BERT tokenizer, which performs tokenization consistent with the previously trained model (known as WordPiece, [Xinyin, 2021]). The comments were divided into training and validation sets, preserving the class distribution. The model was trained for a single epoch, using a batch size of 4 and gradient accumulation every four steps, allowing for the simulation of a larger batch size within the constraints of GPU memory. Other training parameters included:

- learning rate 5e-6
- loss function CrossEntropyLoss
- Adam optimizer
- saving checkpoints in the working folder.

The results obtained by the BERT model are the highest among all tested solutions. In the second iteration of the experiment, the training parameters were modified to improve the classifier's performance. The number of epochs was increased from 1 to 3, allowing the model to better fit the training data. The batch accumulation rate was reduced to per_device_train_batch_size=2, while maintaining gradient accumulation every four steps to reduce GPU memory load. Additionally, an early stopping strategy was implemented, and the best model was automatically selected based on the validation loss. Extending the training time and increasing the stability further improved the results, particularly in terms of overall accuracy and stabilized predictions in the neutral class.

Class	Precision	Recall	F1-score
Negative	0.88	0.92	0.90
Neutral	0.52	0.36	0.43
Positive	0.90	0.94	0.92

Table 1. Sentiment analysis results for BERT

The achieved results confirm the effectiveness of transformer models in text classification tasks, especially when large datasets and pre-trained models for a given language are available. The BERT model served as a strong foundation for transfer learning, enabling high-quality predictions without the need for manual feature creation or advanced text processing.

3. Value analysis

3.1. Idea of value analysis

The problem consists in evaluating an opinion written in Portuguese in terms of its quality. The word “quality” can be interpreted in many ways, e.g. as the usefulness of this opinion for the manufacturer, which enables them to improve their products; as a point of reference for another user, allowing them to learn what others think about a given product; or as a grammatically correct text with rich vocabulary. The evaluation should be presented in a simple format that is easy to interpret.

As part of the project, an AW algorithm (valuation algorithm) was created, which consists of five functions processing the input text and one function adjusting the final result. Each processing function uses different NLP-related concepts and word embeddings to analyze such elements as text structure, the presence of parts of speech serving a descriptive role, opinion detection, identification of thematic words, and text length. The output of each of these functions is expressed as a real number rounded to four decimal places within the interval [0, 1]. The adjusting function is responsible for summing all the obtained results, detecting certain anomalies among them, and reacting accordingly by returning an increased, decreased, or in special cases equal to 0 result.

Since AW consists of five functions returning results from the interval [0, 1], their sum will be a real number within the interval [0, 5]. To improve interpretability, the results are finally rounded to the nearest integer, and in this form they are displayed in the application. The final result can be understood as belonging to one of six value classes, where 0 means worthless opinions and 5 the most valuable.

3.1. Value analysis algorithm

The algorithm for value analysis can be described using the following pseudocode:

Value analysis algorithm
function AW(text, nlp, nouns) score_1, tokens, doc, pos, punctuation ← Structure(text, nlp) score_2 ← Important_Pos(pos, punctuation) score_3 ← Full_Opinion_Comparison(doc) score_4 ← Topic(doc, nouns) score_5 ← Opinion_Length(tokens, score_3) total_score ← Adjustment(score_1, score_2, score_3, score_4, score_5) return round(total_score)

Table 2. Value analysis algorithm.

The algorithm operates using 5 scoring functions:

- Structure
- Important_pos
- Full_opinion_comparison
- Topic
- Opinion_length

Structure

The purpose of this function is to check the overall structure of the input text by analyzing the percentage of each part of speech relative to the average percentage of each part of speech in any other text. The average percentages were calculated using the B2W dataset. The percentage of each part of speech was determined for each opinion in the dataset, and then the average for each part of speech was calculated. The result of these calculations is the following list:

POS	Frequency
NOUN	18.8%
PUNCT	13.8%
VERB	13.6%
ADP	9.6%
DET	8.9%
ADV	8.3%
ADJ	7.2%
CCONJ	3.9%
PRON	3.8%

Table 3. Part of speech frequency in the B2W corpus

Before calculating the percentages, the algorithm searches for punctuation marks using regular expressions, converts the text to a format supported by spaCy [Honnibal, 2017], and checks whether the number of OOV tokens is significantly greater than the text length. If so, the algorithm returns 0, as the text likely contains no real words. It then determines the part of speech of each token, calculates the percentage of each part of speech in the text, and calculates the absolute value of the difference between the opinion's part of speech percentage and the average part of speech

percentage for any given opinion. The results for each part of speech are summed. The function returns the difference from one, unless the result is a negative number, e.g., if the text contains only two of the eighteen available parts of speech. In that case, 0 is returned. This structure allows for the recognition of texts with a developed linguistic structure and rich syntax.

Important Part of Speech

The purpose of this function is to determine whether a given opinion contains parts of speech that serve a descriptive function. Of the eighteen parts of speech determined by spaCy, nine were selected as descriptive in the text and assigned specific weights. If an opinion contains at least one of these parts of speech, it receives a single value added to the final score. Experiments have shown that the final score is multiplied by 125%, as this method yielded better results during evaluation.

Full opinion comparison

The goal of this algorithm is to compare a new opinion with other opinions in the database, assuming the database contains only correctly written opinions in Portuguese. Using spaCy, one can obtain an embedding of a given opinion and then, using cosine similarity, determine whether the new text is similar to all opinions in the database, i.e., whether the new text is an opinion. Because calculating embeddings for many opinions is expensive, the application calculates them in advance and stores them in the database as the "comment.embedding" attribute. The average of the calculated similarities is calculated and the result is multiplied by 150% which, similarly to the relevant parts of speech function, positively impacts the results. The intuition behind this function is that if the input text is not an opinion, this function returns lower results than if the text is a true opinion.

Topic

The purpose of this feature is to analyze text for the presence of nouns specific to a given product. Three products were created for testing: a phone, a robot vacuum cleaner, and a watch. For each product, 50 reviews were written in Portuguese, each containing keywords describing the individual product components. After extracting such words into the topic dictionary, describing product components can be used to check whether they are present in a new review. Their presence indicates that the user is paying attention to the product's details, which increases the value of their review. However, users can provide synonyms to the extracted words, so each noun should be checked for cosine similarity with words from the topic dictionary. The BERTopic library was used for topic modelling, which relies on the BERT model to transform text into contextual embeddings. The library also utilizes UMAP dimensionality reduction and topic extraction using c-TF-IDF, an extension of the standard TF-IDF that can operate on document clusters. Using BERTopic is very simple. Simply load the appropriate model (in this case, BERTopic(language="portuguese")) and run it on the selected dataset. BERTopic automatically determines thematic words for a given set, but as it turned out during testing, it determines them nondeterministically, meaning it can produce different solutions for the same set. Therefore, the model was run 20 times, and the noun words that BERTopic determined most frequently were selected.

Opinion length

The final processing function is a simple check of the number of tokens in the review. For very short reviews, 0.0 is returned, and for very long reviews, 1.0. For reviews of average length, a logistic function of the form:

$$\frac{1}{1+e^{-k(x-x_0)}}$$

is returned, for: $k = 0.068$, $x_0 = 27$. The parameter k was chosen during testing, and the parameter x_0 comes from the calculated average number of tokens from the B2W review dataset. After calculating the result, another condition is checked: whether the review received a score of 0.0 but received a score greater than 0.7 in the opinion detection function. This means that the review is very short, but the author wrote it in a way that makes it similar to other reviews, so it carries value despite the small number of tokens.

3.2. Value analysis evaluation

To evaluate the AW algorithm, a set of one hundred opinions was prepared, rated by a group of independent annotators on a scale of 0 to 5. The opinions were constructed to represent various cases, e.g., very long with rich vocabulary, medium-length general, short with thematic words, etc. The set also included texts that were not opinions but were written correctly or contained thematic vocabulary. The algorithm's results were also tested on random strings of characters or texts containing one or two parts of speech. Because the annotator group was small and consisted of only 15 people, the following assumptions were made during the algorithm evaluation:

1. The algorithm's result for a given opinion was compared to the average result of the annotating group, with the average being trimmed by two extreme scores to avoid outliers.
2. The results of the algorithm and the annotating group are rounded to the nearest multiple of 0.5.
3. A classification is considered correct if the difference between the algorithm's score and the annotating group's score is no greater than 0.5.
4. A classification is considered incorrect if the difference between the algorithm's score and the annotating group's score is greater than 0.5.

Because the results depend on the database content for the opinion detection function, the algorithm was tested using two different databases consisting of 150 different correctly written reviews. One database was used to fine-tune the algorithm's parameters, and the other was the application's default database. Accuracy was used as the evaluation metric. The evaluation results are as follows:

1. Using the first database, the algorithm achieved 84% accuracy on a test database of 100 reviews. Of these, 37 out of 100 were accurately assessed, while 47 out of 100 were assessed with an error of 0.5, which is acceptable. The remaining 16 out of 100 reviews had an unacceptable error of 1.0. None of the tested reviews had an error above 1.0.

2. Using the application's default database, the algorithm achieved 82% accuracy on a test database of 100 reviews. Of these, 39 out of 100 were rated accurately, while 43 out of 100 were rated with an error of 0.5. Another 15 out of 100 reviews had an error of 1.0. Three of the tested reviews had an error above 1.0.

The experimental results show that the AW algorithm produces satisfactory results in assessing the value of reviews in Portuguese. In most cases, the algorithm is within 0.5 of its accuracy. This can be concluded that the presented methods for assessing the value of reviews, consisting of several functions processing text using various NLP methods and one function fine-tuning the final result, were effective in solving the problem.

Conclusions

The presented solution is a comprehensive approach to opinion analysis in Brazilian Portuguese. The opinions are analyzed with respect to their sentiment and potential value for other users.

Despite the challenges presented by the language and the availability of the resources, the designed algorithms proved to be effective. The solution has been implemented in a proof-of-concept project PortuSentio and is ready to be deployed in production scenarios.

References

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

Hochreiter, S., & Schmidhuber, J"urgen. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

Real, L., Oshiro, M., Mafra, A. B2W-Reviews01. 2019. url: https://open cor.gitlab.io/corpora/real19b2wreviews01/ Accessed: 08. 07. 2025.

Souza, Fábio & Nogueira, Rodrigo & Lotufo, Roberto. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. 10.1007/978-3-030-61377-8_28.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece Tokenization. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.