FazGame: A Game Based Platform that Uses Artificial Intelligence to Help Students to Improve Brazilian Portuguese Writing Skills

Jéssica S. Santos and **Gabriel Coelho** and **Sidney Melo** and
Oniram Atila and **Carla Zeltzer**
FazGame, Rio de Janeiro, Brazil
{jessica.soares,gabriel.coelho,sidney.melo}@fazgame.com.br

Abstract

This article describes the FazGame platform, which is a tool developed to assist students' learning process of Brazilian Portuguese as a native language through the creation of narrative and interactive games. The textual content of students' games, such as dialogues between characters and game messages, is evaluated by an Artificial Intelligence based module that uses Natural Language Processing techniques, language models, and classifiers to identify problems such as misspelled words, lack of punctuation and word capitalization, verbal and nominal agreement errors, and cohesion and coherence problems. Based on this analysis, pedagogical interventions are displayed to assist students in the process of improving writing skills. In addition, reports are provided to help teachers monitor students' progress and also to highlight the main difficulties of a given student or class.

1 Introduction

Students in elementary school may face several challenges in the learning process of Brazilian Portuguese (BP) as their native language. These difficulties can stem from a combination of factors, including the complexity of the language, individual learning styles, and the teaching methods employed. Among some common challenges, we can cite:

1. **Spelling and grammar**: Spelling and grammatical errors are common during the language learning process, particularly when students need to learn specific rules. For example, the use of verb conjugations, punctuation, capitalization and accentuation rules, and other grammatical aspects such as agreement can pose a challenge. Also, students may know the words but not know how to write them correctly according to the grammar. It can be common, for example, when it comes to terms that have a phonetic pattern with the same pronunciation but are written differently. One example of this occurs with the following terms in BP: *voce*, *voçe*, *vossê*, *você*. Even though they sound the same, the last case is the only one that corresponds to the correct word, which denotes the pronoun *you* in English.
2. **Cohesion and Coherence**: Expressing thoughts clearly and coherently in writing can be challenging, especially in long texts. Students may face difficulties in organizing ideas, constructing sentences and paragraphs, and developing a cohesive and coherent narrative.
3. **Reading Comprehension**: Limited vocabulary can be a barrier to comprehension. Students may struggle to understand the meanings of new words, affecting their focus and their ability to interpret texts.
4. **Lack of Motivation**: Motivation and interest play a crucial role in learning. Disinterested classes may not fully engage students in BP classes, impacting their understanding and performance.

In this context, the FazGame platform[1] is an educational game-based platform that is being developed to tackle the aforementioned issues. It uses Natural Language Processing (NLP) techniques and Artificial Intelligence (AI) to automatically detect writing problems in the textual content of narrative games created by students. After identifying those problems, the AI module exhibits a set of pedagogical interventions to help the students correct the textual errors or inconsistencies. The results of the AI-based analysis are also stored in a database system in order to persist information and provide reports about the student's learning process, such as the most common category of errors in different periods of time. Such reports aim to help the

[1] https://www.fazgame.com.br

teachers have a vision of the class's problems with regard to learning textual production, and allow them to deal with the main difficulties in an interactive and personalized way, optimizing students' time and learning. Throughout the entire development process, we count on the collaboration of a pedagogical team that plays a crucial role in validating the techniques used and the data provided through interactions with students and teachers on the platform.

2 FazGame platform

On the FazGame platform, students learn by creating narrative games. In addition to create textual elements (like dialogues between characters and game messages), users must select logical connectors (and/or/end of game/scene change) to set game flow according to user actions (like clicking on characters, objects, items, and so on). In this way, it is possible for the same game to result in different stories that vary according to the user's actions, as each action can lead to a different path. For example, the student may create a question dialog that is associated with two possible answers, and each answer redirects the user to a different scene. It is possible to assign points as a reward when the correct answer is selected. Also, games can be associated with different pedagogical tracks that define, for example, the theme that must be explored by the student in the narrative game. Figure 1 illustrates an overview of the entire scope of the project.

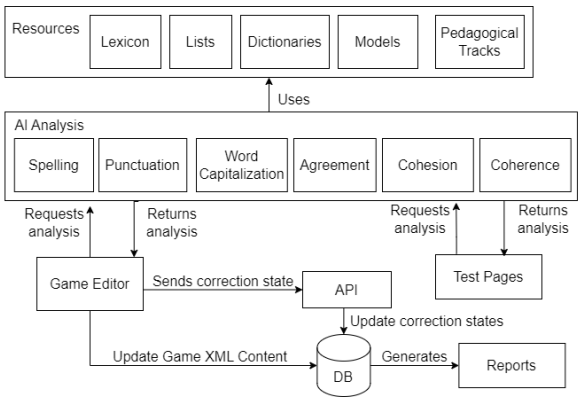

Figure 1: Representation of the entire scope of the project environment.

As can be seen in the *AI Analysis* group depicted in Figure 1, our methods detect problems from six primary categories (Spelling, Punctuation, Word Capitalization, Agreement, Cohesion, Coherence). These categories further branch into specific subcategories, each undergoing tailored treatments that account for linguistic and technological aspects, as described below. Initially, the problems are categorized by considering general mistakes and challenges associated with the structure of words and sentences when writing, particularly in the context of elementary school proficiency, as described in Section 1. For future work, we intend to refine our methods for different scenarios, depending on specific grade levels of elementary school.

Textual Analysis – Spelling and Grammar: The textual analysis related to spelling and grammar combines a set of strategies: Levenshtein (Yujian and Bo, 2007) algorithm, lexicon and auxiliary lists, phonetic patterns, Large Language Models (LLMs) for analyzing suggestions probability, Part-of-speech tagging (POS-tag) and Named Entity Recognition (NER) (Marrero et al., 2013) techniques, and a set of predefined rules. The Levenshtein algorithm and a lexicon resource (based on BRispell[2]) are used to find potential suggestions for misspelled words. In this step, a set of phonetic patterns is also considered. In cases where there are many suggestions for a given misspelled word (many lexicon words at the same Levenshtein distance), we use the BERTimbau (Souza et al., 2020) model to select the most likely word to be suggested in the given sentence context. We also save in the database the occurrence of offensive terms[3] and informal language, such as slangs, presented in the text. A list containing popular foreign words is used to prevent them from being recognized as language errors. We also use a list of names combined with POS-tag and NER models to identify characters' names and avoid recognizing them as errors, although they are not listed in the lexicon. Also, this process is important to detect capitalization and punctuation errors. Punctuation error detection also uses predefined rules. One example of that is the comma vocative error, when there is no comma in a dialogue that is started by a greeting and is followed by a proper noun (e.g.: "*Oi Maria*" which corresponds to "*Hi Maria*" in English). SpaCy library[4] and LanguageTool rules (Mozgovoy, 2011) are adopted to support part of this process. Verbal and nominal agreement errors are captured by combining predefined rules with a dependency parser

[2] https://www.ime.usp.br/~ueda/br.ispell/

[3] Additionally, these terms are replaced by special characters so that they are hidden from the game.

[4] https://spacy.io/

and morphological features analysis.

Textual Analysis – Cohesion and Coherence: Conjunctions are very important elements of cohesion in textual construction as they help to chain ideas together. To look for cohesion problems, we use the BERTimbau model to identify potential incorrect uses of conjunctions according to context. Next, the total number of correct uses of conjunctions is counted and divided by the total number of words, as this metric can be helpful to evaluate textual cohesion (Leal et al., 2023). To evaluate game coherence, we intend to measure two aspects: (i) game graph analysis: development of an algorithm that runs through the game graph and searches for paths that do not lead to other scenes and, consequently, to the end of the game; and (ii) alignment of game theme with the track that it is associated when this is the case. We are investigating strategies like Topic Modeling (Kherwa and Bansal, 2019), Topic Extraction (Campos et al., 2020), Embeddings, and LLMs to check whether the game textual content fits to the track theme.

All strategies and technologies used for textual analysis are incorporated into the platform following a review and development procedure that consists of: I) checking whether the strategy provides solutions for BP, II) checking whether the primary purpose of the strategy aligns with the requirements of our textual analysis context, III) aligning which part of the strategy scope in our textual analysis pipeline, IV) validating the potential results and output possibilities from the analysis in collaboration with the pedagogical team, V) refining the output considering linguistic gaps from the strategy identified in the validation stage, VI) implementing pre- and post-processing to align the input and output required for our textual analysis scope, VII) testing the final method with existing games and new sentences representing diverse usage scenarios. Additional linguistic rules can be mapped by the pedagogical team in the fourth step, in addition to ensuring that the method accurately analyzes and provides correct data regarding the problem categories listed previously.

Pedagogical Interventions: Textual interventions can be viewed as tips to guide students in the correction of their errors, such as the example illustrated in Figure 2. Instead of automatically correcting the text, a set of sequential tips is presented to the students as an opportunity to think about their mistakes and correct them. This set of interventions is created in advance based on each error subcategory detected in the *AI Analysis* and represents part of the output shown in the *Game Editor* (depicted in Figure 1 by the arrow named "*Returns analysis*" going from the *AI Analysis* to the *Game Editor*). In each case, the pedagogical team formulates a systematic series of recommendations that aims to reach the solution related to the detected error step-by-step. The suggestions encompass examples of both correct and incorrect sentences or guides on locating the spelling errors and presenting alternatives for their resolution. In general, interventions are organized into three levels that differ according to their specificity and have dynamic fields for displaying the word/sentence with the error detected and for the student to rewrite it based on the recommendations. For example, the third and last intervention can be the presentation of the most likely suggestion. Students can agree with the suggestion or ignore it.

As previously stated, the suggestions strictly concern the structural aspects of writing words and sentences that relate to the challenges of elementary school proficiency mentioned in Section 1. Refining these suggestions is within the scope of future work, considering different interactions with the students on the platform according to elementary school grades.

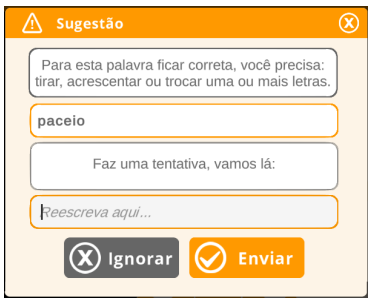

Figure 2: Spelling error - First pedagogical intervention.

Reports: Reports are generated based on the errors stored in the SQL database. An overall summary of the learning process of students and classes is exhibited, taking into account three topics: (i) errors distribution by category in a given period; (ii) most popular errors by category in a given period; (iii) total of correct words divided by the total number of words in two different periods of time to capture students/classes progress. An example of an available report is exhibited in Figure 3.

Figure 3: Error distribution by category - Pie chart

The FazGame platform integrates a Rails application with the Unity game editor, which in turn connects to NLP and AI-based modules that were implemented in Python and are made available through AWS Lambda functions. All communication is done through APIs.

3 Tool Demonstration

The FazGame platform can be accessed online through access logins. We intend to demonstrate its functionalities by giving access to the audience so that they can create narrative games and see the pedagogical interventions based on NLP and AI that are displayed to users after textual analysis. Furthermore, the audience will be able to access reports generated based on the detection of errors in BP language. We also intend to make test pages available so that users can test features related to NLP and AI modules that are not yet integrated into the game editor.

Acknowledgements

This work was partially funded by FAPESP and CNPq.

References

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Pooja Kherwa and Poonam Bansal. 2019. Topic modeling: a comprehensive review. *EAI Endorsed transactions on scalable information systems*, 7(24).

Sidney Evaldo Leal, Magali Sanches Duran, Carolina Evaristo Scarton, Nathan Siegle Hartmann, and Sandra Maria Aluísio. 2023. Nilc-metrix: assessing the complexity of written and spoken language in brazilian portuguese. *Language Resources and Evaluation*, pages 1–38.

Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. 2013. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489.

Maxim Mozgovoy. 2011. Dependency-based rules for grammar checking with languagetool. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 209–212. IEEE.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. Bertimbau: pretrained bert models for brazilian portuguese. In *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9*, pages 403–417. Springer.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.