Extending an Ensemble Baseline with Corpus-Based Graph Features for Portuguese Pun Detection

Avelar Rodrigues de Sousa
University of São Paulo (USP)
avelars@usp.br

Camilla Soares Sousa
Federal Institute of Piauí (IFPI)
camillasoares818@gmail.com

Carlos Henrique Santos Barros
Federal University of Piauí (UFPI)
carlos.barros.cb@ufpi.edu.br

Rafael Torres Anchiêta
Federal Institute of Maranhão (IFMA)
rafael.torres@ifma.edu.br

Abstract

Automatic pun detection remains challenging because it depends on lexical ambiguity and contextual interaction, which are not explicitly captured by linear text representations. In Portuguese, TF-IDF-based ensemble methods provide competitive and interpretable baselines, but remain limited by surface-level features. This work investigates whether corpus-based graph information can complement such methods. Three graph representations are constructed from the *Puntuguese* corpus: a Co-occurrence graph, a PPMI-weighted graph, and a Pun-Context graph. In the current pipeline, each graph is converted into low-dimensional node embeddings with TruncatedSVD, which are then aggregated into document-level features and concatenated with TF-IDF representations in a soft-voting ensemble. Experimental results on the test set show that graph-based enrichment does not uniformly improve performance: Pun-Context and PPMI yield the strongest graph-augmented results, whereas combining all graphs degrades performance. These findings indicate that the usefulness of graph-based information depends strongly on how lexical relations are encoded and aggregated at the document level.

1 Introduction

Automatic pun detection remains challenging in Natural Language Processing (NLP) because it depends on lexical ambiguity and contextual interaction, which may yield multiple interpretations within a single utterance. These properties complicate semantic disambiguation and linear meaning composition, making pun detection particularly sensitive to representational choices (Miller et al., 2017; Kao et al., 2016). More broadly, humor recognition depends on subtle linguistic cues and on the availability of annotated resources (Kalloniatis and Adamidis, 2024), while related ambiguity phenomena have also been explored in applied scenarios, including interactive and game-based systems (de Sousa et al., 2025).

Methodologically, pun detection has largely relied on shallow vector representations such as bag-of-words and Term Frequency–Inverse Document Frequency (TF-IDF) combined with traditional supervised classifiers (Miller et al., 2017; Kao et al., 2016). More recently, neural and transformer-based models have also been explored, especially to capture broader contextual dependencies in figurative language (Devlin et al., 2019; Liu et al., 2019). Yet both traditional and neural approaches predominantly represent text as linear token sequences, limiting the explicit modeling of relations among words. This is particularly relevant for pun detection, since puns often arise from semantic contrast, phonetic similarity, or contextually activated alternative meanings.

In Portuguese, progress in pun detection has been supported by dedicated, methodologically controlled corpora. In particular, *Puntuguese* (Inacio et al., 2024) provides punning one-liners paired with non-humorous counterparts generated through controlled micro-edits, offering a reliable benchmark for supervised approaches. Building on this resource, Leal et al. (2025) proposed a TF-IDF-based ensemble with soft voting over traditional classifiers, achieving competitive results with low computational cost, interpretability, and experimental reproducibility.

Despite these advantages, the ensemble approach proposed by Leal et al. (2025) relies exclusively on linear vector representations and does not explicitly incorporate structural or relational information between lexical units. Similar limitations have been observed in other Portuguese NLP tasks involving semantic ambiguity and meaning interaction, where surface-oriented models struggle to capture more nuanced relations (Sousa and Anchiêta, 2025). This motivates the investigation of complementary representations that encode ex-

Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)
Vol. 1, pages 759–769

plicit relationships between terms while preserving the efficiency and interpretability of traditional machine learning models.

In this work, we extend the ensemble approach of Leal et al. (2025) by incorporating structural information derived from corpus-based graphs. Specifically, we explore three graph representations: a lexical Co-occurrence graph, a Positive Pointwise Mutual Information (PPMI)-weighted graph, and a Pun-Context graph built exclusively from instances labeled as puns. In the current implementation, each graph is converted into low-dimensional node embeddings, aggregated into document-level features, and integrated with TF-IDF representations in the ensemble framework, enabling analysis of each graph type's contribution to performance.

Experimental results show that graph-based enrichment does not uniformly improve over the lexical baseline. Among the graph-augmented configurations, Pun-Context yields the strongest performance, followed closely by PPMI, whereas combining all graph sources degrades results. These findings suggest that the usefulness of structural information depends not only on the inclusion of graph-derived signals, but also on how selectively lexical relations are encoded and aggregated at the document level.

The remainder of this paper is organized as follows. Section 2 reviews previous work on pun detection and related semantic tasks. Section 3 describes the dataset and preprocessing procedures. Section 4 presents the baseline ensemble approach, and Section 5 details the construction of corpus-based graph representations and their integration into the model. Section 6 reports the experimental setup and results, and Section 7 concludes the paper.

2 Related Work

Research on automatic pun detection lies within the broader study of verbal humor and figurative language and focuses on computational strategies for identifying semantic ambiguity and wordplay. Much of this work has been developed in shared evaluation settings such as SemEval, especially for English (Miller et al., 2017). Early approaches relied mainly on shallow vector representations combined with traditional supervised classifiers, showing that lexical and statistical cues can partially capture wordplay. However, the ambiguous and relational nature of puns exposes the limitations of models that treat text as linear sequences, motivating representations capable of handling multiple senses and contextual interactions (Kao et al., 2016; Miller et al., 2017; Kalloniatis and Adamidis, 2024).

In Portuguese, advances in pun detection have been supported by dedicated, methodologically controlled datasets. The *Puntuguese* corpus (Inacio et al., 2024) is particularly relevant because it combines positive pun instances with negative examples generated through controlled textual micro-edits, reducing superficial biases such as trivial differences in length or vocabulary. Even under these controlled conditions, pun detection remains challenging.

More recently, neural approaches based on contextualized language models have also been investigated for Portuguese pun detection. In particular, Êmylle Sousa et al. (2025) applied the BERTimbau Large model to the *Puntuguese* corpus, showing that transformer-based architectures can capture relevant linguistic cues associated with wordplay and semantic ambiguity. Although the results are competitive, the study also highlights challenges related to generalization, overfitting, and the computational cost of large neural models. Moreover, despite their representational power, such models often operate as opaque sequence encoders, offering limited interpretability regarding which lexical relations or ambiguity mechanisms drive their decisions. Thus, their performance gains do not eliminate the need for lighter, more interpretable approaches that explicitly encode structural relations between terms.

From a methodological perspective, ensemble learning strategies have been explored in NLP to combine different inductive biases and improve the robustness of supervised models, particularly when using traditional vector representations. For Portuguese pun detection, Leal et al. (2025) proposed a TF-IDF-based ensemble that combines classifiers such as Random Forest (RF), Logistic Regression (LR), and Support Vector Machine (SVM) through soft voting. The reported results show that combining traditional classifiers can achieve competitive performance with low computational cost and high experimental reproducibility. However, the method remains grounded in linear vector representations and does not explicitly incorporate semantic or relational structures between words.

Taken together, the studies reviewed in this sec-

tion indicate that most approaches to pun detection rely on vector-based text representations and supervised classifiers, ranging from traditional machine learning methods to transformer-based neural models. In particular, the approach of Leal et al. (2025) provides a strong baseline while leaving open the question of how structural information might complement linear representations. Building on this perspective, the present work investigates the impact of incorporating corpus-based graph information into ensemble models for pun detection in Portuguese.

3 Dataset Description

The experiments reported in this work use the *Puntuguese* corpus, introduced by Inacio et al. (2024) for automatic pun detection in Portuguese. The corpus comprises short humorous *one-liner* texts in two linguistic varieties: Brazilian Portuguese (PT-BR) and European Portuguese (PT-PT). In its full version, *Puntuguese* contains 4,903 humorous instances collected from heterogeneous sources. Due to licensing constraints, only a publicly available subset of 2,850 humorous instances is released, comprising 2,053 PT-BR and 797 PT-PT examples. Because each humorous instance is paired with a manually produced non-humorous counterpart through micro-editing, the classification setting used here operates over pun/non-pun pairs derived from this public subset. Table 1 presents illustrative examples of puns from both linguistic varieties in the corpus.

Linguistic variety	Pun
PT-BR	*Deve ser difícil ser professor de natação. Você ensina, ensina, e o aluno* ***nada***. (It must be hard to be a swimming teacher. You teach and teach, and the student **swims / learns nothing**.)
PT-PT	*Hoje vi o Jorge Jesus num anúncio de detergentes em que ele dizia: Este é o melhor detergente que* ***tenho Tide****!* (Today I saw Jorge Jesus in a detergent advert in which he said: This is the best detergent I **have Tide / have had**. The pun relies on the phonetic similarity between the detergent brand name *Tide* and the Portuguese expression *tenho tido* [I have had].)

Table 1: Examples of puns in Brazilian Portuguese and European Portuguese from the *Puntuguese* corpus.

A central characteristic of *Puntuguese* is its parallel structure. For each humorous instance, a non-humorous counterpart was manually created through micro-editing, with minimal textual modifications that preserve grammaticality, discourse coherence, and overall meaning while removing the humor. This design reduces superficial cues, such as differences in length or syntax, and provides a controlled benchmark for evaluating models beyond surface-level textual representations. Table 2 illustrates this micro-editing process with a PT-PT example.

Pun (PT-PT)
Um homem ia-se mandar de um prédio quando um físico gritou lá de baixo: Não faça isso! Você tem muito ***potencial****!*
(A man was about to jump from a building when a physicist shouted from below: Don't do that! You have a lot of **potential**.)
Non-humorous version (Micro-edit)
Um homem ia-se mandar de um prédio quando um físico gritou lá de baixo: Não faça isso! Você é muito importante!
(A man was about to jump from a building when a physicist shouted from below: Don't do that! You are very important!)

Table 2: Illustration of the micro-editing process in *Puntuguese* using a PT-PT example, where semantic ambiguity is removed while preserving syntactic structure and discourse context.

In this study, only the publicly available subset of the corpus is used. The released classification data are balanced and distributed into training, validation, and test files. In the current pipeline, model fitting is performed exclusively on the training split and final evaluation on the test split, without using the validation split. This choice keeps the experimental setup methodologically explicit and avoids mixing model development with final evaluation.

4 Baseline Method

As a methodological starting point, this work adopts the ensemble learning approach proposed by Leal et al. (2025), specifically developed for automatic pun detection in Portuguese. The method combines traditional supervised classifiers trained on TF-IDF text representations, offering a favorable balance between simplicity, interpretability, and reproducibility, while achieving competitive results on the *Puntuguese* corpus.

In the current baseline pipeline, texts are represented using the TF-IDF weighting scheme (Salton

and Buckley, 1988), with unigram and bigram features. The vectorizer applies lowercasing, accent normalization, and Portuguese stopword removal, yielding a sparse lexical representation that preserves the simplicity and interpretability of the reference ensemble approach while adapting it to the current implementation.

Based on these representations, the ensemble combines classifiers with complementary inductive biases: a Random Forest classifier, a Logistic Regression classifier with class balancing, and a Linear Support Vector Classifier calibrated through CalibratedClassifierCV to produce probabilistic outputs. This design preserves the soft-voting principle of the original ensemble family while making the current implementation compatible with feature-level fusion between sparse TF-IDF vectors and graph-derived attributes. The final prediction is obtained via soft voting, in which the probabilistic outputs of the individual classifiers are aggregated.

Despite its effectiveness, the baseline relies on linear vector representations and does not explicitly model semantic or relational interactions between words, a limitation that is particularly relevant for pun detection. The following section therefore introduces graph-based representations as a controlled extension of the reference ensemble. Rather than replacing TF-IDF with contextual embeddings, the goal is to examine whether explicitly modeled relational information can complement a strong, interpretable, and computationally lightweight baseline while preserving direct comparability with prior work. This choice should be understood not as an argument against contextual embedding approaches, but as a deliberate test of graph-based enrichment under controlled and interpretable conditions.

5 Corpus-Based Graph Representations

This section introduces corpus-based graphs that enrich the baseline with structural information not explicitly captured by TF-IDF vectorization. Because puns depend not only on specific lexical items but also on the relations they establish with surrounding context and other lexical units, graphs provide a way to make this structure explicit within the reference ensemble framework. The graph construction and feature-extraction pipeline were implemented in Python and are available in the project repository[1]. The following subsections describe the graph types considered and how their information is integrated into the model.

5.1 Co-occurrence Graph

The Co-occurrence graph models distributional associations between tokens based on their joint occurrence within local context windows, following the classical formulation of lexical association grounded in co-occurrence as proposed by Church and Hanks (1990) and widely systematized in the NLP literature (Jurafsky and Martin, 2025). During graph construction, each node corresponds to a normalized token filtered by a minimum frequency threshold ($min\_freq = 5$). Two words are connected whenever they occur within a sliding window ($window = 5$), and the edge weight corresponds to the raw co-occurrence count. To control graph density and mitigate the influence of highly frequent terms, a local neighborhood pruning strategy is applied, retaining only the highest-weight connections ($top\_k = 30$).

After filtering and pruning, the Co-occurrence graph used in this work comprises 1,153 nodes and 22,730 edges, forming a single connected component with a density of 0.034225. The nodes with the highest weighted degree—defined as the sum of the weights of incident edges—are predominantly function words and connectives in the corpus (for example, "*que*" [that/which], "*qual*" [which], and "*de*" [of]), indicating that the resulting structure primarily captures recurrent syntactic and discourse-level regularities.

The graph edges are represented in a tabular structure containing three attributes: the source token (src), the target token (dst), and the observed co-occurrence frequency between them (weight). Table 3 presents real examples of such relations extracted from the constructed graph.

src	dst	weight
qual (which)	*que* (that/which)	1546
de (of)	*que* (that/which)	1232
por (by/for)	*que* (that/which)	1114
de (of)	*qual* (which)	1082
porque (because)	*que* (that/which)	958

Table 3: Examples of edges from the Co-occurrence graph, with Portuguese tokens.

Table 3 shows that the highest-weight edges pre-

[1] https://github.com/liara-ifpi/pun-detection-kg-ensemble

dominantly involve functional tokens. In the graph constructed in this work, this suggests a stronger sensitivity to recurrent surface-level regularities than to semantically specific associations.

Although raw co-occurrence makes lexical proximity relations explicit, it is strongly influenced by high-frequency tokens and very general combinations, limiting its ability to highlight associations more directly related to pun ambiguity. This motivates the use of statistical normalizations that penalize chance-level co-occurrences and emphasize more informative links, as discussed in the following subsection (Kao et al., 2016; Miller et al., 2017).

An illustrative example of a Co-occurrence graph derived from a corpus utterance is provided in Appendix A, Figure 1.

5.2 PPMI-Based Graph

While raw co-occurrence captures statistical proximity between terms, it tends to privilege associations involving high-frequency tokens, predominantly reflecting surface-level regularities and general syntactic–discursive patterns. To mitigate this bias and render lexical relations more informative, this work explores a variant of the Co-occurrence graph in which edges are weighted by PPMI, a classical measure of lexical association based on mutual information (Church and Hanks, 1990). This statistical measure normalizes co-occurrence counts as a function of the marginal frequencies of the terms, allowing the identification of associations that occur with probability higher than expected by chance (Jurafsky and Martin, 2025) and that are therefore potentially more relevant for modeling lexical ambiguity in puns (Kao et al., 2016).

Operationally, the PPMI graph preserves the basic structure of the Co-occurrence graph, including its nodes and local context-window criterion, but replaces raw co-occurrence counts with Positive Pointwise Mutual Information weights computed from joint and marginal term probabilities (Jurafsky and Martin, 2025). Negative PMI values are truncated to zero so that only statistically associative relations contribute to the graph.

After weighting, the graph is pruned with the same local neighborhood procedure used for Co-occurrence, retaining only the highest-weight connections for each node. This yields a sparser and more selective structure, reducing the influence of highly frequent terms and favoring more informative lexical relations, a behavior consistent with previous findings on statistical association measures such as PPMI (Church and Hanks, 1990; Levy et al., 2015).

src	dst	weight (PPMI)
ele (he)	*morrer* (to die)	3.564155
cantor (singer)	*mim* (me)	2.882842
para (to/for)	*volta* (return)	2.689423
da (of/from)	*herói* (hero)	2.120703
compras (shopping)	*da* (of/from)	1.532917

Table 4: Examples of edges from the PPMI graph, with Portuguese tokens.

Table 4 presents examples of edges with high PPMI values, highlighting lexical associations that are more specific when compared to those observed in the raw Co-occurrence graph. Unlike that scenario, the highlighted pairs are not dominated exclusively by highly frequent functional tokens; instead, they involve pronouns, verbs, and nouns that occur in more restricted contexts within the corpus. Relations such as *"ele"–"morrer"* (he–to die) and *"para"–"volta"* (to/for–return) illustrate contextually marked associations whose co-occurrence cannot be explained solely by the marginal frequency of the terms. This behavior indicates that PPMI-based weighting favors more informative links, making the graph more selective and potentially better suited to capturing lexical patterns relevant to the automatic detection of puns.

In summary, the PPMI graph is a statistical reweighting of the co-occurrence structure that mitigates the impact of trivial lexical regularities without altering the underlying graph-construction criterion. By emphasizing less expected and more contextually restricted associations, this representation complements the raw Co-occurrence graph by providing a more selective view of the relations between terms. In this work, the PPMI graph is explored as an additional source of structural information to be integrated into the reference method, enabling the assessment of the extent to which statistical normalization of co-occurrences contributes to the automatic detection of puns.

An illustrative example of a PPMI-based graph derived from a corpus utterance is provided in Appendix A, Figure 2.

5.3 Pun-Context Graph

While Co-occurrence and PPMI graphs capture global distributional patterns, they do not explicitly

represent the contexts in which puns occur. To address this, a third graph, the Pun-Context Graph, is constructed from instances labeled as puns, modeling the relations between terms that arise in humorous discourse contexts and lexical ambiguity.

Operationally, the Pun-Context Graph is built from corpus instances labeled as puns, but unlike the Co-occurrence and PPMI graphs, it is not generated with a local sliding-window procedure. For each positive instance, the implementation identifies tokens marked as belonging to the pun span and connects them to normalized tokens in the full textual context of the utterance, as illustrated in Appendix A, Figure 3. The resulting graph, therefore, encodes relations between pun tokens and their local discourse environments rather than general token-to-token co-occurrence alone.

After edge construction, the resulting structure is pruned through a local neighborhood filtering step, retaining only the highest-weight connections for each node. This reduces graph density and attenuates weak or highly idiosyncratic links. Because the graph is built from relations between annotated pun tokens and the surrounding tokens in positive examples, the resulting structure tends to emphasize lexical environments recurrently associated with pun occurrences rather than general corpus-wide co-occurrence patterns. In this sense, the Pun-Context graph complements the Co-occurrence and PPMI graphs by focusing more directly on the annotated humorous mechanism present in the dataset.

src	dst	weight
roupa (clothes)	*varau* (clothesline)	5.0
depressão (depression)	*panela* (pot)	4.0
consola (console)	*wii* (Wii)	4.0
jogos (games)	*wii* (Wii)	4.0
mar (sea)	*rita* (Rita)	4.0

Table 5: Examples of edges from the Pun-Context graph, with Portuguese tokens.

Table 5 presents examples of edges from the Pun-Context graph with relatively high weights, highlighting recurrent associations in humorous utterances from the corpus. Unlike graphs constructed from the complete set of texts, the pairs shown here more frequently reflect thematic content and lexically marked combinations typical of joke scenarios and wordplay (e.g., *"roupa"–"varau"* [clothes–clothesline] and *"consola"–"wii"* [console–wii]). This pattern is consistent with the objective of the graph, namely to restrict the modeling of relations to co-occurrences observed in instances labeled as puns, so that the resulting links are more sensitive to discursive contexts associated with verbal humor in the *Puntuguese* corpus (Inacio et al., 2024).

In summary, the Pun-Context graph complements the Co-occurrence and PPMI graphs by focusing explicitly on humorous contexts in the corpus. In this work, it is integrated into the reference ensemble-based method (Leal et al., 2025) to assess whether recurrent associations in pun contexts improve the distinction between humorous and non-humorous instances.

5.4 Integration of Graph-Based Features into the Baseline Method

Graph-derived information is integrated into the baseline classifier as a modular feature-level extension. Texts are first represented through TF-IDF vectorization with unigram and bigram features, after which graph information is incorporated by appending a compact set of document-level features derived from each pre-constructed graph. The complete implementation of this integration and training pipeline, including graph loading, feature extraction, scaling, and classification, is available in the experimental repository.

For each graph, node representations are first obtained by applying *TruncatedSVD* to the weighted adjacency matrix. In this setting, *TruncatedSVD* acts as a dimensionality-reduction method that projects the original graph structure into a lower-dimensional latent space, so that each node is represented by a compact continuous vector, here referred to as a node embedding. In the present implementation, the number of latent dimensions is fixed to four. These dimensions do not correspond to predefined linguistic categories; rather, they encode compressed structural variation derived from the weighted connectivity patterns of the graph.

Given a textual instance, the system tokenizes the text, matches the resulting tokens against the nodes of a given graph, and averages the embeddings of all matched nodes. Two additional scalar features are appended to this mean embedding: (i) the proportion of text tokens that were found in the graph (coverage) and (ii) the logarithm of one plus the number of matched graph tokens ($\log(1 + hits)$). As a result, each graph contributes a six-dimensional document-level feature vector: four latent graph-embedding dimensions and two

auxiliary coverage-related features.

When more than one graph is used, the corresponding document-level vectors are concatenated horizontally. These graph-based vectors are standardized, using parameters estimated exclusively on the training split, and then converted to sparse format. Feature fusion is finally performed by concatenating the sparse graph matrix with the sparse TF-IDF matrix, characterizing an early-fusion strategy.

The resulting fused matrix is used as input to a soft-voting ensemble composed of Random Forest, Logistic Regression, and a calibrated LinearSVC. This design preserves the general ensemble-learning rationale of the reference approach while adapting the implementation to the current graph-enriched setting. All experimental configurations are evaluated under the same train/test protocol, allowing a controlled comparison of the contribution of each graph representation.

The purpose of this design is not to exhaustively optimize each graph-based configuration independently, but rather to compare them within a unified, controlled experimental setting, so that the observed differences can be interpreted primarily as effects of the graph representations themselves and of their document-level aggregation strategy. Accordingly, the graph embedding dimensionality was fixed a priori at four in order to keep the graph-derived representation compact and directly comparable across graph types, rather than as the result of a separate hyperparameter optimization stage.

6 Results and Analysis

This section reports the experimental results obtained by extending the baseline ensemble-learning method with structural information derived from corpus-based graphs. Following the methodology described in Section 4, models are trained on the training split and evaluated on the test split of the *Puntuguese* corpus. Results are reported per class (*Pun* and *Not a pun*), allowing analysis of both class-wise behavior and overall performance across configurations.

Table 6 shows clear differences across configurations. Among the graph-based extensions, TF-IDF + Pun-Context achieves the best overall accuracy (0.78) and the highest recall for the *Pun* class, whereas TF-IDF + PPMI yields the most balanced graph-based profile across classes. TF-IDF + Co-occurrence performs slightly worse than these two variants, and TF-IDF + All Graphs produces the weakest graph-augmented result, suggesting that direct concatenation of heterogeneous graph signals introduces redundancy or noise rather than consistent complementarity.

Compared to the TF-IDF baseline, none of the graph-augmented configurations improves overall accuracy. Even so, the comparison remains informative because it shows that graph-based information changes the behavior of the ensemble depending on the type of relation encoded. Since the present system is a direct graph-based extension of the ensemble strategy proposed by Leal et al. (2025), the baseline is retained as a reference point for interpreting how graph features modify prediction behavior under the same general modeling rationale.

6.1 Confusion-Matrix Comparison Across Configurations

Approach	TP	TN	FP	FN
TF-IDF + Co-occurrence	447	425	145	123
TF-IDF + PPMI	452	431	139	118
TF-IDF + Pun-Context	**464**	425	145	106
TF-IDF + All Graphs	428	421	149	142
TF-IDF (Baseline)	450	**459**	**111**	**120**

Table 7: Confusion-matrix counts for the graph-augmented configurations and the TF-IDF baseline on the test set.

Table 7 clarifies how graph-based enrichment changes the error profile of the ensemble relative to the TF-IDF baseline. In particular, the table reports the counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) for each configuration. Rather than producing uniform gains, the graph variants mainly redistribute errors across classes.

The clearest example is TF-IDF + Pun-Context, which increases TP (464) and reduces FN (106), but at the cost of more FP (145) and fewer TN (425). This indicates a shift toward greater sensitivity to humorous instances, accompanied by lower specificity for non-punning texts. TF-IDF + PPMI shows a milder version of this trade-off, yielding a more balanced error profile overall. By contrast, TF-IDF + All Graphs degrades both pun recovery and non-pun discrimination, suggesting that direct graph concatenation introduces competing

Approach	Class	Precision	Recall	F-score	Accuracy
TF-IDF + Co-occurrence	Not a pun	0.78	0.75	0.76	0.76
	Pun	**0.76**	0.78	0.77	
TF-IDF + PPMI	Not a pun	0.79	0.76	0.77	0.77
	Pun	**0.76**	0.79	0.78	
TF-IDF + Pun-Context	Not a pun	0.80	0.75	0.77	0.78
	Pun	**0.76**	**0.81**	0.79	
TF-IDF + All Graphs	Not a pun	0.75	0.74	0.74	0.74
	Pun	**0.74**	0.75	0.74	
TF-IDF (Baseline)	Not a pun	0.79	0.81	0.80	**0.80**
	Pun	**0.80**	0.79	**0.80**	

Table 6: Classification results on the test set for the graph-augmented configurations of the current pipeline, reported per class, alongside the TF-IDF baseline kept as a reference comparison to the original ensemble-oriented setup.

or redundant signals rather than stable complementarity.

Taken together, the confusion matrices indicate that graph-based enrichment is best interpreted as changing the balance between recall for *Pun* and specificity for *Not a pun*, rather than as uniformly improving the baseline.

6.2 Agreement and Disagreement Across Graph Configurations

Beyond per-model confusion matrices, agreement patterns across graph-based variants show that most of the test set lies in consensus regions: of the 1,140 instances, 767 are correctly classified by all graph-based variants and 175 are misclassified by all of them. The effective differences among graph representations are therefore concentrated in a relatively small subset of cases.

Within this disagreement region, TF-IDF + All Graphs shows limited but unstable complementarity: it correctly classifies 51 instances missed by all individual graph variants, but fails on 85 cases that all individual variants classify correctly. This asymmetry helps explain its weaker overall performance. Pun-Context is the most distinctive individual graph representation, with 13 uniquely correct cases, whereas Co-occurrence and PPMI each contribute only 3. These patterns indicate that the graph-based variants are not fully redundant, but differ mainly in localized cases, especially in the contrast between Pun-Context and All Graphs.

6.3 Error Analysis by Punning-Sign Type

Following the error-analysis protocol adopted by Leal et al. (2025), we further examined false negatives according to punning-sign type. More specifically, misclassified pun instances were categorized as involving: (i) neither homographic nor homophonic mechanisms, (ii) homophone-only mechanisms, (iii) homograph-only mechanisms, or (iv) both homographic and homophonic mechanisms. As in the reference study, these counts are computed over punning signs rather than over jokes, since a single joke may contain more than one annotated sign. This analysis is useful because it allows us to verify not only whether graph-based enrichment changes overall performance, but also whether it changes the linguistic profile of the errors. In other words, rather than asking only which configuration performs better globally, we ask which kinds of puns remain difficult for each graph-based representation. The full distribution across the graph-based configurations is reported in Appendix B, Table 8, visually summarized in Figure 4, and complemented by illustrative examples in Table 9.

Across the graph-based configurations, the most difficult cases remain those classified as neither homographic nor homophonic, as shown in Table 8 and Figure 4. This pattern is consistent with the error profile reported by Leal et al. (2025), suggesting that graph-based enrichment does not eliminate the challenge posed by less transparent and more creative punning mechanisms. However, the graph variants differ in how strongly they preserve or attenuate this difficulty.

The graph-based variants do not affect all punning-sign types uniformly. Pun-Context yields the most favorable false-negative profile, with the lowest count in the neither category and lower counts for both-type puns. PPMI shows a similar but smaller improvement, also reducing false negatives for neither and both. Co-occurrence does not mitigate the dominant error pattern, increasing

false negatives in the neither and homophone-only categories, while TF-IDF + All Graphs produces the weakest profile overall, worsening nearly all categories, especially neither. Homograph-only cases are extremely rare across all graph-based configurations.

Taken together, these findings indicate that graph-based features affect not only aggregate performance, but also the linguistic nature of the remaining errors. In particular, Pun-Context appears to be the graph representation that most effectively reduces difficult false negatives, although puns that are neither homographic nor homophonic remain the dominant source of error in all settings.

A plausible explanation for the weaker performance of the graph-augmented configurations is that graph information may have been compressed too aggressively at the document level. In the current pipeline, each graph is reduced to a small pooled feature vector based on mean node embeddings, token coverage, and graph-hit counts. Although computationally efficient, this design may smooth out the localized lexical interactions most relevant for pun detection. In addition, because the graph-based pipeline differs slightly from the original baseline in classifier configuration and text preprocessing, part of the observed degradation may reflect not only the contribution of graph features themselves, but also differences in the modeling setup.

7 Conclusions

This work investigated the impact of incorporating graph-derived information into an ensemble approach for automatic pun detection in Portuguese. Starting from a TF-IDF-based classifier, three graph representations were explored: Co-occurrence, PPMI, and Pun-Context. These graphs were converted into low-dimensional node embeddings with TruncatedSVD, aggregated into compact document-level features, and combined with TF-IDF through early concatenation.

The experimental results on the test split show that graph-based enrichment does not produce uniform gains. Among the evaluated configurations, Pun-Context achieves the best performance among the graph-based approaches, followed closely by PPMI, whereas the concatenation of all graph sources degrades performance. These findings suggest that the usefulness of graph information depends both on the type of relation being modeled and on how it is converted into document-level features.

More broadly, this study shows that graph-based augmentation can be integrated into an interpretable ensemble-learning pipeline without abandoning sparse lexical representations. At the same time, the results indicate that richer feature sets are not necessarily better, especially when heterogeneous graph signals are combined without additional selection or regularization.

The main limitations of the present study concern the compactness of graph aggregation at the document level and the restriction of the experiments to the publicly available portion of *Puntuguese* under a single train/test protocol. Future work may therefore investigate alternative graph-construction criteria, more expressive pooling strategies, feature-selection mechanisms better suited to preserving graph structure, and broader evaluations across different datasets, domains, and languages.

Taken together with the error patterns discussed for the original baseline by Leal et al. (2025), the present results suggest that the main challenge is not simply the absence of structural information, but the difficulty of encoding complex linguistic mechanisms of puns in a way that remains selective after document-level aggregation while still complementing a strong lexical baseline.

References

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Avelar de Sousa, Camilla Sousa, and Carlos Barros. 2025. Usabilidade e engajamento em jogos casuais diários: Estudo de caso do guess of dreams. In *Anais da XIII Escola Regional de Informática de Goiás*, pages 1–10, Porto Alegre, RS, Brasil. SBC.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcio Lima Inacio, Gabriela Wick-Pedro, Renata Ramisch, Luís Espírito Santo, Xiomara S. Q. Chacon, Roney Santos, Rogério Sousa, Rafael Anchiêta, and Hugo Goncalo Oliveira. 2024. Puntuguese: A corpus of puns in Portuguese with micro-edits. In *Pro-*

ceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 13332–13343, Torino, Italia. ELRA and ICCL.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*, 3rd edition. Online manuscript released August 24, 2025.

Antonios Kalloniatis and Panagiotis Adamidis. 2024. Computational humor recognition: a systematic literature review. *Artificial Intelligence Review*, 58(2):43.

Justine T. Kao, Roger Levy, and Noah D. Goodman. 2016. A computational model of linguistic humor in puns. *Cognitive Science*, 40(5):1270–1285.

Jhúlia Leal, Marcio Inácio, Hugo Oliveira, and Rafael Anchiêta. 2025. Improving pun detection with an ensemble of traditional machine learning methods. In *Anais do XVI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 206–219, Porto Alegre, RS, Brasil. SBC.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Avelar Sousa and Rafael Anchiêta. 2025. Sistira: Plataforma web de tutoria inteligente para avaliação automática de respostas discursivas. In *Anais do XVI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 733–737, Porto Alegre, RS, Brasil. SBC.

Êmylle Sousa, Aislan Sousa, and Rafael Anchiêta. 2025. Classificação de trocadilhos em português com bertimbau large: Desafios e resultados. In *Anais do XVII Encontro Unificado de Computação do Piauí*, pages 139–148, Porto Alegre, RS, Brasil. SBC.

A Illustrative Graph Examples

Figures 1, 2, and 3 present illustrative examples of the three graph representations discussed in Section 5, all based on the following utterance from the *Puntuguese* corpus:

> *"O que é que as freiras comem ao pequeno almoço? Papa."*
> "What do nuns eat for breakfast? Pope / porridge."

The figures show graph structures derived from this utterance under the three construction criteria adopted in this work: Co-occurrence, PPMI-based reweighting, and Pun-Context.

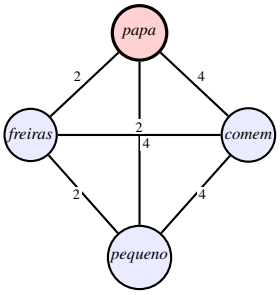

Figure 1: Illustrative example of a Co-occurrence graph derived from the example utterance. Edge weights correspond to co-occurrence counts.

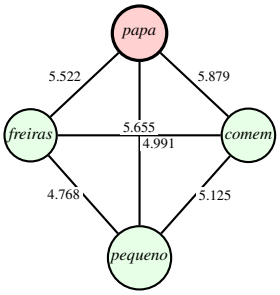

Figure 2: Illustrative example of a PPMI-based graph derived from the same utterance. Edge weights reflect PPMI-based association strength.

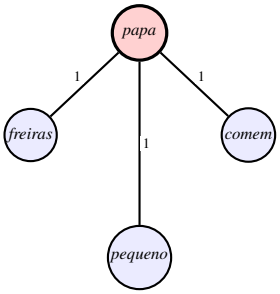

Figure 3: Illustrative example of a Pun-Context graph derived from the same utterance. The highlighted node corresponds to the pun token and its links to context tokens.

B Additional Error-Analysis Tables and Figures

Approach	Neither	Homophone-only	Homograph-only	Both
TF-IDF + Co-occurrence	73	30	0	22
TF-IDF + PPMI	69	31	0	20
TF-IDF + Pun-Context	**60**	28	0	**20**
TF-IDF + All Graphs	86	30	0	29

Table 8: Distribution of FN punning signs by type across the graph-augmented configurations. Counts are computed over annotated punning signs rather than over jokes.

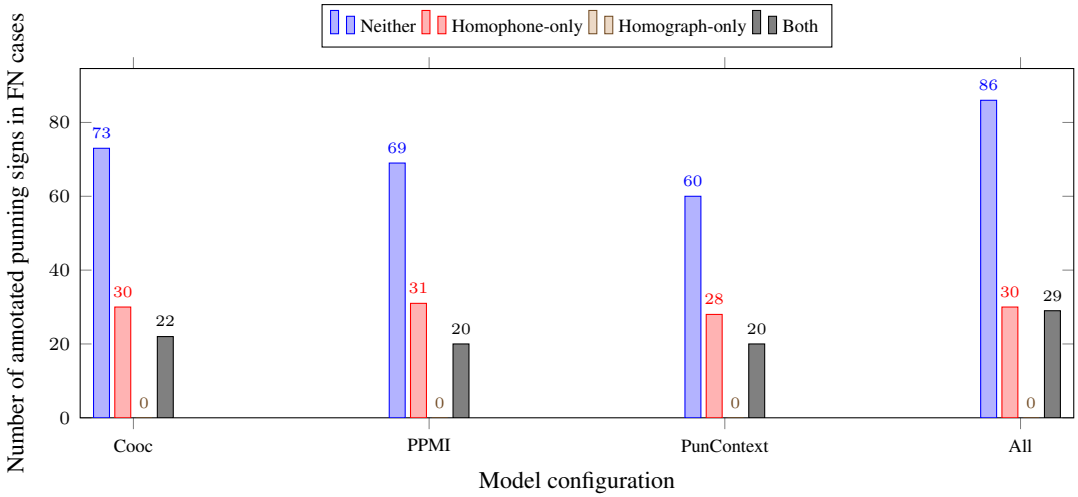

Figure 4: Distribution of annotated punning signs found within false-negative cases on the test set, grouped by punning-sign type for the graph-augmented configurations. Counts are computed over signs rather than over jokes.

Homographic	Homophonic	Pun	Comment
×	×	*Hoje vi o Jorge Jesus num anúncio de detergentes em que ele dizia: Este é o melhor detergente que tenho Tide!* (Today I saw Jorge Jesus in a detergent advert in which he said: This is the best detergent I **have Tide / have had**.)	The humorous effect depends on the approximation between *Tide* and *tido* [have had], but in the annotation file this pun is marked as neither homographic nor homophonic.
×		*Porque é que os polícias não gostam de sabão? Porque preferem deter gente.* (Why do police officers not like soap? Because they prefer **to arrest people / detergent**.)	This pun is annotated as homophone-only because *deter* [to arrest] + *gente* [people] evokes *detergente* [detergent] through phonetic similarity.
	×	*O meu sonho é ser probre um dia. É que ser probre todos os dias é lixado.* (My dream is to be **probre** one day. Being **probre** every day is awful.)	This pun is annotated as homograph-only in the dataset, with the relevant sign marked as *um* [one / a].
		Um homem ia-se mandar dum prédio, passa um físico lá em baixo: Não faça isso! Você tem muito potencial! (A man was about to jump from a building when a physicist passed below and said: Do not do that! You have a lot of **potential**!)	This pun is annotated as both homographic and homophonic, with the humorous effect centered on *potencial* [potential].

Table 9: Illustrative examples of puns across different punning-sign types.