When the Split Matters: Reorganizing Micro-Edited Pairs for Portuguese Pun Detection

Avelar Rodrigues de Sousa[1], **Maykon Willyam de Sousa Ferreira**[1], **Victor Oliveira Lima Pontes**[1], **Rafael Torres Anchiêta**[2]

[1] Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
São Carlos – SP – Brazil
{avelars, maykonwsf, victoroliveiralimapontes}@usp.br

[2]Institute Federal of Maranhão (IFMA)
Caxias – MA – Brazil
rafael.torres@ifma.edu.br

Abstract. *Pun detection is a challenging task in Natural Language Processing, as it relies on lexical, semantic, phonological, and contextual cues. The Puntuguese corpus contains micro-edited pairs of punning and non-punning sentences that share the same base identifier, which may create dependencies when pairs are distributed across training, validation, and test splits. This paper investigates how split organization affects pun detection evaluation in Portuguese. We compare the original Puntuguese splits with two controlled organizations: Pair-Preserving, which keeps paired instances in the same partition, and Maximum-Crossing, which maximizes pair distribution across partitions. We evaluate a BERTimbau-based classifier and an ensemble of traditional classifiers. Results show that split organization substantially affects model performance and ranking: the ensemble benefits from increasing pair crossing, with accuracy varying from 0.47 in Pair-Preserving to 0.80 in the Original split and 0.96 in Maximum-Crossing, whereas BERTimbau large achieves its highest accuracy in the Pair-Preserving setting (0.76) and does not benefit from additional crossing. Consequently, conclusions drawn from the original splits do not hold once the pairing is controlled, and we recommend that evaluations on Puntuguese report and, when appropriate, control the paired structure of the corpus.*

1. Introduction

Computational humor processing is a challenging task in Natural Language Processing (NLP), as it involves subjective, ambiguous, and context-dependent linguistic phenomena [Attardo 2020, Kalloniatis and Adamidis 2024]. Among the different forms of verbal humor, puns are especially relevant because they exploit polysemy, homonymy, or phonological similarity to produce a humorous effect [Miller et al. 2017, Kao et al. 2016]. This characteristic makes automatic pun detection difficult, since the model must distinguish humorous from non-humorous texts based on lexical, semantic, and contextual cues.

This study focuses on the *Puntuguese* corpus[1], which contains puns in Brazilian Portuguese (PT-BR) and European Portuguese (PT-PT), as well as negative examples

[1]Corpus available at: https://huggingface.co/datasets/Superar/Puntuguese

created through micro-editing [Inacio et al. 2024]. In this structure, each pun has a similar non-humorous version, created to differ primarily in humorous effect, a strategy related to the use of minimal edits in computational humor resources [Hossain et al. 2019, Inacio et al. 2024]. However, since the positive and negative versions of each pair share the same Base Identifier (IDB), the distribution of these pairs across different splits may compromise the independence between training and evaluation.

For example, if the pun "*A faxineira não luta judô porque ela luta capoeira.*" ("The cleaning lady doesn't practice judo because she practices *capoeira*.") is in the training set, and its non-pun version, "*A faxineira não luta judô porque ela luta com a poeira.*" ("The cleaning lady doesn't practice judo because she fights with dust"), is present in the test set, a model may be trained on one version of the pair and evaluated on the other highly similar version. Although this case does not constitute direct label leakage, it creates a dependency between textually similar examples across different partitions. This configuration is related to the discussion by Kaufman et al. [Kaufman et al. 2012] on leakage in data mining, insofar as characteristics of the corpus construction process may influence the estimation of model performance.

This study investigates whether the distribution of micro-edited pairs across different partitions of *Puntuguese* may affect the evaluation of models for pun detection in Portuguese. We analyze the original splits and construct two controlled organizations: Pair-Preserving, in which instances with the same IDB remain in the same partition, and Maximum-Crossing, which maximizes the number of pairs spanning training and evaluation while preserving split sizes and class balance. Both controlled organizations are generated using six split seeds. We then evaluate a BERTimbau-based neural baseline [Souza et al. 2020, Sousa et al. 2025] and an ensemble of traditional classifiers [Leal et al. 2025]. With this, we seek to answer the following Research Questions (RQs):

- **RQ1:** Are there micro-edited pairs with the same IDB distributed across training, validation, and test sets in the original splits of *Puntuguese*?
- **RQ2:** How does model performance change across the Original, Pair-Preserving, and Maximum-Crossing split organizations?
- **RQ3:** Does the relative performance of pun detection models remain stable across different organizations of micro-edited pairs?

Our results show that the split organization substantially affects both the accuracy estimates and the relative ranking of the two models: the ensemble improves steadily as pair crossing increases, whereas BERTimbau large performs best when no pair crosses.

The main contributions of this paper are: (i) the quantification of micro-edited pairs distributed across the original splits of *Puntuguese*; (ii) the construction and evaluation of Pair-Preserving and Maximum-Crossing organizations across six split seeds; and (iii) the analysis of how pair distribution affects model performance and the relative comparison between approaches. The remainder of the paper is organized as follows: Section 2 presents the related work; Section 3 describes the original corpus and presents an analysis of its organization; Section 4 describes the three split organizations, the experimental protocol, and the evaluated models; Section 5 presents and discusses the results; and Section 6 presents the conclusions and future work.

2. Related Work

Puns in NLP can be addressed at different levels — detection, localization, and interpretation [Miller et al. 2017] — so identifying a pun is only an initial step, and computational models must handle the interplay between linguistic form, meaning, and context [Kao et al. 2016].

For Portuguese, the *Puntuguese* corpus [Inacio et al. 2024] provides puns in PT-BR and PT-PT with non-humorous counterparts created through micro-editing — a strategy related to Humicroedit [Hossain et al. 2019] — which reduces superficial differences between classes and makes the task more controlled. The organization of these pairs, however, requires experimental care, since distributing similar examples across partitions may affect evaluation.

Neural and traditional models have already been evaluated for pun detection in Portuguese. Inacio et al. [Inacio et al. 2024] reported experiments with BERTimbau on *Puntuguese*, while Leal et al. [Leal et al. 2025] proposed an ensemble based on traditional classifiers and Term Frequency–Inverse Document Frequency (TF-IDF) representations, obtaining competitive results with lower computational cost. Later studies also investigated related tasks, such as locating the pun's pivot word with BERT-based models and Mixture of Experts [Anchiêta et al. 2026], as well as incorporating graph-based features into the traditional ensemble [Sousa et al. 2026]. These studies advance the computational treatment of puns in Portuguese, but they assume the available splits of *Puntuguese* as their starting point.

Unlike these studies, this paper focuses on the experimental organization of the corpus splits. We frame the phenomenon as a statistical dependency (correlation) between partitions rather than as classical data leakage: because the two versions of a pair carry opposite labels, no test label is exposed during training. This dependency nonetheless relates to the literature on leakage in data mining, which discusses how dependencies between training and evaluation can compromise performance estimates [Kaufman et al. 2012]. Specifically, we investigate how the distribution of micro-edited pairs with the same IDB affects model evaluation by comparing the original splits with two controlled pair organizations representing opposite crossing conditions.

3. Corpus Description and Analysis

The corpus analyzed in this study is divided into three splits: training, validation, and test. Each instance has an Identifier (ID) ending in "`.H`" or "`.N`". The suffix "`.H`" indicates an instance containing a pun, whereas the suffix "`.N`" indicates its non-pun counterpart. We refer to the common part between these two versions as the IDB. Table 1 presents examples of pun/non-pun pairs that share the same IDB and illustrates how micro-editing changes a specific part of the text to remove the humorous effect.

Table 2 presents the distribution of labeled instances by split. The three splits are balanced with respect to the classes, with the same number of pun and non-pun instances in each split.

Although the class distribution is balanced, the analysis by IDB reveals that micro-edited pairs are not necessarily restricted to the same split. This is relevant because the pun and non-pun versions share a large part of their lexical structure, differing mainly in

Table 1. Examples of pun/non-pun pairs with the same IDB.

ID	Text	Comment
`1.3.H`	*Se um pato perde a pata, ele fica manco ou viúvo?* (If a duck loses its leg/female duck, does it become lame or widowed?)	The pun version exploits the ambiguity of *pata*, which may refer both to an animal's leg/paw and to a female duck.
`1.3.N`	*Se um pato perde a esposa, ele fica manco ou viúvo?* (If a duck loses its wife, does it become lame or widowed?)	The non-pun version replaces *pata* with *esposa* (wife), removing the lexical ambiguity that supports the pun.
`2.14.H`	*Por que o astronauta cometeu um homicídio no espaço e não foi preso? Porque foi um crime sem gravidade.* (Why did the astronaut commit homicide in space and was not arrested? Because it was a crime with no gravity/seriousness.)	The pun version exploits the double meaning of *gravidade*, which is associated with both gravity in physics and the seriousness of a crime.
`2.14.N`	*Por que o astronauta cometeu um homicídio no espaço e não foi preso? Porque foi um crime sem dolo.* (Why did the astronaut commit homicide in space and was not arrested? Because it was a crime without intent.)	The non-pun version preserves an answer from the legal domain, but removes the lexical association with the space context.

Table 2. Distribution of labeled instances by split.

Split	Total	Pun	Non-pun
Validation	570	285	285
Training	3990	1995	1995
Test	1140	570	570

the segment responsible for the presence or removal of the pun.

To quantify this distribution, Figure 1 presents a cross-tabulation between pun and non-pun instances with the same IDB. Rows indicate the split of the pun version and columns the split of its non-pun counterpart. Diagonal cells (in bold) represent pairs contained within the same split, whereas off-diagonal cells represent cross-split pairs. Each cell also reports its percentage of the 2,850 pairs.

The matrix shows that class balance does not guarantee the separation of micro-edited pairs. Out of 2,850 pun/non-pun pairs, 1,544 remain within the same split, whereas 1,306 (45.8%) are distributed across different partitions. Among these, 780 pairs (27.4% of all pairs) specifically cross the training-test boundary.

These results show that class balance alone does not prevent highly similar paired instances from being distributed across different partitions, motivating the controlled split analysis described in Section 4.

4. Methodology

The adopted methodology builds on the analysis presented in Section 3, in which micro-edited pairs distributed across different partitions of *Puntuguese* were identified. Based on

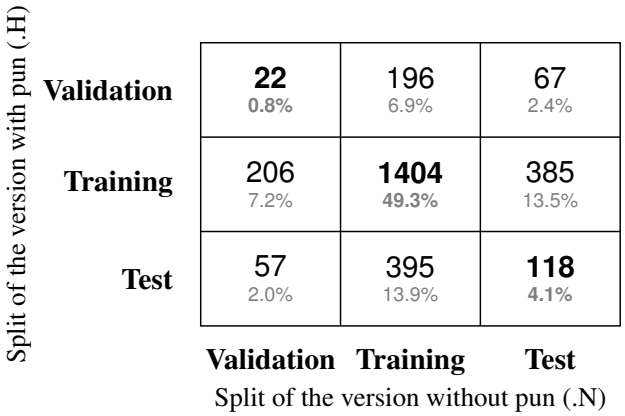


Figure 1. Cross-tabulation between pun and non-pun versions by split.

this finding, two controlled split organizations were constructed: Pair-Preserving, which keeps both instances of each IDB in the same partition, and Maximum-Crossing, which maximizes the number of pairs spanning the training and evaluation partitions. The original split was preserved as the reference configuration, as illustrated in Figure 2.

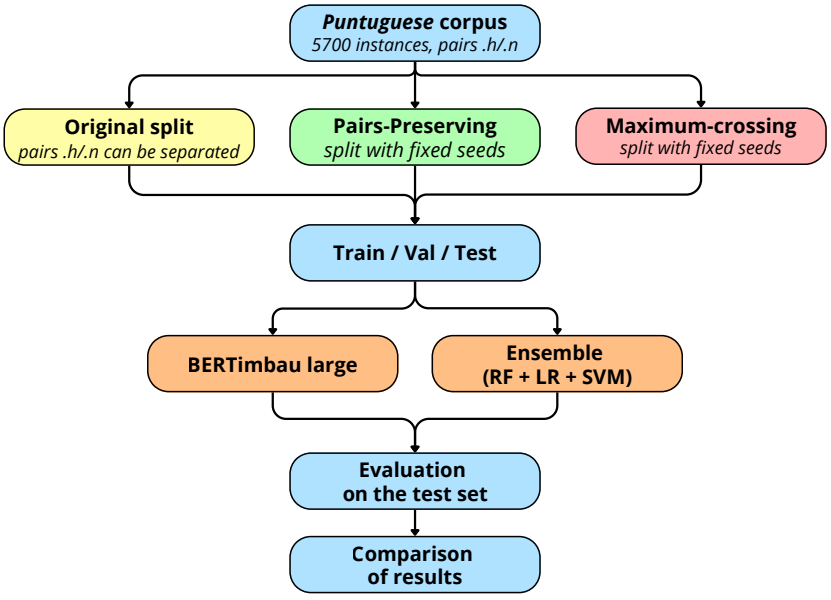


Figure 2. Representation of the methodological approach.

The reorganization procedure was implemented by first merging the original training, validation, and test files into a single dataset. The IDB was extracted from each instance by removing the final suffix "`.H`" or "`.N`" from its identifier. The suffix "`.H`" corresponds to pun instances, labeled as 1, whereas the suffix "`.N`" corresponds to non-pun instances, labeled as 0. After this extraction, the corpus contained 5,700 instances and 2,850 IDBs, each associated with exactly two examples: one pun version and one non-pun version.

The pairs were then treated as the unit of redistribution. Since each pair consists of exactly two instances, one from each class, and the original partition sizes (3,990, 570, and 1,140 instances) are even numbers, redistributing by pairs reproduces exactly

the same partition sizes and the same class balance as the original corpus, without any approximation: each partition receives half pun and half non-pun instances (1,995/1,995 in training, 285/285 in validation, and 570/570 in test). To reduce the dependence on a single random partition, each reorganized configuration was generated with six fixed seeds (13, 21, 40, 42, 73, and 101).

In the Pair-Preserving configuration, the two versions of a given IDB are always assigned to the same partition, so that no pair crosses the boundaries between training, validation, and test (0 crossing pairs). In the Maximum-Crossing configuration, the assignment is designed to maximize the number of pairs whose pun and non-pun versions fall into different partitions, subject to the constraint that at least one version of each pair remains in the training set; under the corpus sizes, this yields 1,710 crossing pairs, or 60% of all pairs. Both configurations preserve the partition sizes and the class balance, isolating the organization of the pairs as the only variable manipulated with respect to the original split.

In all configurations, two binary classification approaches were evaluated for pun detection. The first is a neural classifier based on BERTimbau large[2]. The texts were tokenized with a maximum sequence length of 256 tokens. The model was trained for up to 6 epochs, with a learning rate of 2×10^{-5}, a batch size of 8 for training and evaluation, a weight decay of 0.01, a warmup ratio of 0.1, and a maximum gradient norm of 1.0. Early stopping with a patience of 2 epochs was applied, and the best checkpoint was selected based on macro-F1 on the validation set. The random seed used for model training was 40.

The second approach is an ensemble of traditional machine learning classifiers, following the lower-cost classification setting explored by Leal et al. [Leal et al. 2025]. Texts were represented using TF-IDF features with unigram and bigram ranges and Portuguese stopword removal. Three classifiers were trained on these representations: Random Forest (RF), Logistic Regression (LR), and Support Vector Machine (SVM). The RF used 100 estimators, the entropy criterion, a maximum depth of 15, and a random seed of 40; LR used a random seed of 40 and a maximum of 100 iterations; and the SVM was configured with probabilistic outputs. The final prediction was obtained through soft voting among the three classifiers. The same scripts and hyperparameters were used across all configurations; only the input corpus files were changed.

The evaluation was performed on the test partition of each experimental configuration. Since the corpus is balanced between the pun and non-pun classes, accuracy was adopted as the main comparison metric, complemented by precision, recall, and F1-score to assess variations in model performance across classes. For each model and configuration, the metrics are reported as the mean and standard deviation over the six split seeds.

5. Results and Discussion

This section presents the results obtained in the experimental configurations considered: the original splits of *Puntuguese* and two IDB-controlled reorganizations: Pair-Preserving and Maximum-Crossing. As discussed in Section 3, the analysis of the distribution of pun/non-pun pairs showed that the original splits allow micro-edited versions associated

[2]https://huggingface.co/neuralmind/bert-large-portuguese-cased

with the same IDB to be distributed across different partitions, answering the first RQ affirmatively. Based on this finding, the experiments were conducted to verify whether this organization affects the evaluation of pun detection models.

5.1. Overall model performance

Table 3 presents the results obtained on the original corpus. In this scenario, the ensemble reported by Leal et al. [Leal et al. 2025] achieves the best overall performance, with an accuracy of 0.80 and balanced results between the non-pun and pun classes. The BERTimbau large model used in this analysis achieves an accuracy of 0.71. Unlike the ensemble, BERTimbau presents a higher recall for the pun class, but lower performance in identifying non-pun instances, resulting in a more asymmetric behavior between classes.

Table 3. Results obtained on the original corpus.

Approach	Class	Precision	Recall	F1-score	Accuracy
Ensemble [Leal et al. 2025]	Non-pun	0.79	**0.81**	**0.80**	**0.80**
	Pun	**0.80**	0.79	**0.80**	
BERTimbau	Non-pun	**0.79**	0.56	0.66	0.71
	Pun	0.66	**0.85**	0.74	

Table 4 presents the mean and standard deviation of results obtained across six split seeds for the Pair-Preserving and Maximum-Crossing controlled split strategies. The results reveal opposite behaviors for the two models and, most notably, a reversal of their ranking depending on the degree of pair crossing. In the Pair-Preserving configuration, BERTimbau large achieves the highest mean accuracy (0.756 ± 0.010), with balanced performance across classes, while the ensemble drops to 0.471 ± 0.009, with a particularly low recall for the pun class (0.261 ± 0.061). In the Maximum-Crossing configuration, the pattern is reversed: the ensemble reaches a mean accuracy of 0.960 ± 0.003, whereas BERTimbau drops to 0.691 ± 0.020.

Table 4. Results on the reorganized corpus for the Pair-Preserving and Maximum-Crossing split strategies (mean $\pm$ standard deviation over six split seeds).

Split	Approach	Class	Precision	Recall	F1-score	Accuracy
Pair-Preserving	Ensemble	Non-pun	0.479 ± 0.007	0.680 ± 0.068	0.561 ± 0.028	0.471 ± 0.009
		Pun	0.449 ± 0.016	0.261 ± 0.061	0.327 ± 0.047	
	BERTimbau	Non-pun	$\mathbf{0.735 \pm 0.016}$	$\mathbf{0.803 \pm 0.024}$	$\mathbf{0.767 \pm 0.008}$	$\mathbf{0.756 \pm 0.010}$
		Pun	$\mathbf{0.783 \pm 0.015}$	$\mathbf{0.709 \pm 0.031}$	$\mathbf{0.744 \pm 0.015}$	
Maximum-Crossing	Ensemble	Non-pun	$\mathbf{0.963 \pm 0.009}$	$\mathbf{0.957 \pm 0.006}$	$\mathbf{0.960 \pm 0.003}$	$\mathbf{0.960 \pm 0.003}$
		Pun	$\mathbf{0.957 \pm 0.005}$	$\mathbf{0.963 \pm 0.009}$	$\mathbf{0.960 \pm 0.004}$	
	BERTimbau	Non-pun	0.674 ± 0.028	0.746 ± 0.061	0.706 ± 0.025	0.691 ± 0.020
		Pun	0.718 ± 0.037	0.635 ± 0.061	0.672 ± 0.028	

Ordering the three configurations by increasing pair crossing — Pair-Preserving, the original split, and Maximum-Crossing — the ensemble's accuracy grows monoton-

ically, whereas BERTimbau does not benefit from crossing and achieves its best result precisely when no pair crosses (Tables 3 and 4).

We hypothesize that this contrast reflects how each model uses the pairwise structure. The ensemble relies on TF-IDF features, so when a pair crosses the training-test boundary the near-duplicate training counterpart provides an almost verbatim lexical shortcut, inflating accuracy as crossing increases; when no pair crosses (Pair-Preserving), this shortcut disappears and the model, unable to rely on surface overlap, collapses below the majority-class baseline. BERTimbau, in contrast, does not gain from this shortcut: its slightly higher accuracy under Pair-Preserving is consistent with a cleaner training signal, in which each pair contributes a coherent contrast between a pun and its non-pun counterpart within the same partition, rather than with any leakage-driven inflation. Importantly, the Maximum-Crossing configuration was designed precisely to test whether the reorganization had introduced a new bias: the fact that only the ensemble benefits from crossing, monotonically, indicates that the effect stems from pair crossing itself and not from an artifact of the redistribution.

The comparison across configurations shows that reorganizing the splits changes the observed performance, answering the second RQ. However, the effect is not uniform across models: pair crossing strongly favors the ensemble and is neutral or slightly detrimental to BERTimbau. As a consequence, the relative ranking between the models reverses depending on the organization of the pairs: under Pair-Preserving, BERTimbau outperforms the ensemble, whereas under Maximum-Crossing (as well as on the original split), the ensemble appears ahead. Thus, the conclusions drawn from the original corpus prove unstable when the evaluation controls the distribution of micro-edited pairs via IDB, thereby answering the third RQ.

It is important to note that the test partitions are balanced, so a majority-class baseline would reach about 0.50. The ensemble's accuracy under Pair-Preserving falls below even this trivial baseline, indicating that, once pair crossing is removed, the model becomes biased toward the non-pun class, as detailed in the error analysis.

5.2. Error analysis

To complement the overall comparison, we analyze the errors produced by each model in the three configurations. Table 5 reports the confusion matrices in summarized form — True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). For the Pair-Preserving and Maximum-Crossing configurations, which involve six seeds, we report the highest-accuracy seed of each model; the Original split corresponds to a single run.

For BERTimbau, the direction of the errors changes with the degree of pair crossing. On the Original split, the model produces far more FP (249) than FN (86); that is, it over-predicts the pun class, which is consistent with its high recall for puns and low recall for non-puns. In the Pair-Preserving configuration the errors become more balanced (114 FP and 152 FN), matching the symmetric per-class performance observed when no pair crosses the partitions. Under Maximum-Crossing, FN rise again (218), indicating that pair crossing slightly degrades the recognition of puns for this model.

The ensemble exhibits a markedly different pattern. On the Original split its errors are balanced (111 FP and 120 FN). In the Pair-Preserving configuration, however, FN

Table 5. Summarized confusion matrices of the evaluated models. For the controlled configurations, the highest-accuracy seed of each model is reported. Bold values indicate, for each model, the best result in each column (highest TP and TN, lowest FP and FN).

Model	Split	Seed	TP	TN	FP	FN
BERTimbau	Original	—	**484**	321	249	**86**
	Pair-Preserving	13	418	**456**	**114**	152
	Maximum-Crossing	21	352	**456**	**114**	218
Ensemble	Original	—	450	459	111	120
	Pair-Preserving	101	134	423	147	436
	Maximum-Crossing	21	**558**	**543**	**27**	**12**

increase sharply to 436: once the near-duplicate counterpart is no longer available across partitions, the model fails to recognize most puns and collapses toward the non-pun class. In the Maximum-Crossing configuration the opposite occurs: with 1,710 pairs spanning training and test, the ensemble reaches almost perfect scores (only 12 FN and 27 FP).

We further analyze which pun types concentrate the false negatives. Figure 3 reports the punning-sign categories annotated in the FN cases. Since a pun may carry more than one sign, the analysis counts annotated signs rather than exclusive instances; and, because the total number of FN differs substantially across configurations, the figure should be read as absolute counts rather than proportions. Across all settings, the "None" category (false negatives not associated with any homophony or homography annotation) is frequent, and cases annotated only as homographs are rare. The "Both" category (homograph and homophone simultaneously) is comparatively more represented among BERTimbau's false negatives than among the ensemble's, whereas the ensemble's errors under Pair-Preserving are dominated simply by the sheer volume of false negatives across every category.

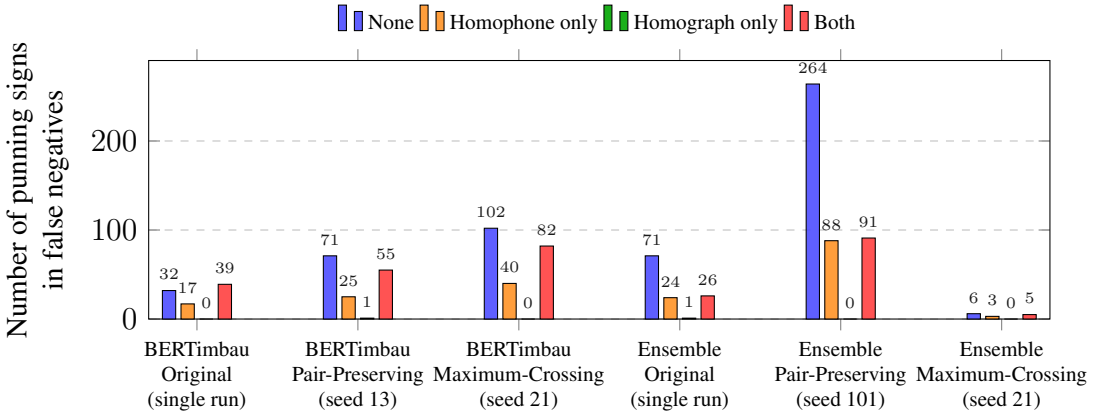


Figure 3. Distribution of punning-sign categories in false negatives (absolute counts). Controlled configurations use the highest-accuracy seed of each model, indicated per group.

Each pun in *Puntuguese* is annotated by the lexical relationship that creates it: it can be a **homograph** (same spelling, different meaning), a **homophone** (same sound, different spelling), **both**, or **neither** [Leal et al. 2025]. Table 6 presents one false negative

of the best BERTimbau run (Pair-Preserving split, seed 13) for each case — that is, puns the model classified as non-puns — illustrating why such wordplay is hard to capture from surface text alone.

Table 6. Examples of false negatives of the highest-accuracy BERTimbau run (Pair-Preserving, seed 13), following Leal et al. [Leal et al. 2025]. Each pun is marked as homographic (Hg) and/or homophonic (Hp).

Hg	Hp	Pun	Comment
	×	*Porque é que os daltónicos têm péssima memória? Porque não sabem nada de cor.* (Why do color-blind people have such a bad memory? Because they know nothing by heart.)	"de cor" (by heart) is written like "cor" (color) but pronounced differently: same spelling, different sound.
×		*Qual a seita dos maus perdedores? A seita que dói menos.* (What is the sect of sore losers? The sect that hurts less.)	"a seita" (the sect) sounds like "aceita" (accepts): same sound, different spelling.
		Qual o filme que o Jerry mais tem medo? 50 Tons de Cinza. (Which film is Jerry most afraid of? 50 Shades of Grey.)	"Tons" (shades) evokes "Toms" (the cat Tom): the wordplay holds in both spelling and sound.
×	×	*Como a manicure se refere à esposa do irmão dela? Unhadinha.* (How does the manicurist refer to her brother's wife? *Unhadinha*)	"Unhadinha" blends "unha" (nail) and "cunhadinha" (sister-in-law): a neologism, neither homograph nor homophone.

In line with Leal et al. [Leal et al. 2025], the "neither" category — neologisms and signs without a homophony or homography annotation — concentrates most errors, whereas purely homographic puns are rare (a single test instance).

Overall, the error analysis reinforces that pair crossing changes not only the aggregate performance but also the direction of the errors. The effect is clearest for the ensemble, whose collapse under Pair-Preserving is driven by a large increase in false negatives: when micro-edited pairs are kept together, the model not only confuses non-puns with puns but, above all, fails to recognize humorous instances.

6. Conclusions

This article analyzed the experimental organization of the *Puntuguese* corpus for pun detection in Portuguese. The analysis of the original splits showed that micro-edited pairs sharing the same Base Identifier are frequently distributed across different partitions: 1,306 of the 2,850 pairs (45.8%) cross partition boundaries, and 780 (27.4%) specifically cross the training–test boundary (RQ1). Rather than classical label leakage — the two versions of a pair carry opposite labels — this constitutes a statistical dependency between partitions that can affect model evaluation.

To study this effect, we built two controlled organizations that isolate pair crossing while preserving partition sizes and class balance — Pair-Preserving and Maximum-Crossing — each generated with six seeds. As pair crossing increases, the ensemble's accuracy rises steadily while BERTimbau performs best when no pair crosses; consequently, both the performance estimates and the relative ranking of the models depend on how micro-edited pairs are distributed, answering RQ2 and RQ3. The Maximum-Crossing configuration further shows that the ensemble's strong result on the original corpus stems largely from lexical overlap between crossing pairs rather than from genuine pun detection.

The error analysis reinforced this interpretation. Under Pair-Preserving, the ensemble collapses toward the non-pun class, with a sharp increase in false negatives, whereas BERTimbau keeps a more balanced distribution of errors. The false negatives concentrate in puns annotated as both homograph and homophone or in puns without an explicit phonological or graphical annotation, which is consistent with the difficulty of separating a pun from its minimally edited counterpart using surface features alone.

These findings suggest that future evaluations on *Puntuguese* should explicitly report and, when appropriate, control the paired structure of the corpus, especially when comparing models or estimating their generalization ability. As future work, the analysis can be extended to other neural and traditional models, including multilingual and generative approaches; to alternative reorganization strategies; and to a deeper qualitative study of the false negatives, in order to characterize which linguistic properties of puns remain most difficult for automatic models.

Availability of Code and Materials

The source code and experimental materials used in this study are openly available at: `https://github.com/avelando/pun-detection-split-analysis`. The repository includes the scripts for reorganizing the corpus splits, the experimental pipeline, and the configuration files describing the execution environment and library versions required for reproduction.

References

Anchiêta, R. T., Santos, R. L. d. S., and Moura, R. S. (2026). Token-level pun location using multi-layer BERT with mixture of experts. In Souza, M., de Dios-Flores, I., Santos, D., Freitas, L., Souza, J. W. d. C., and Ribeiro, E., editors, *Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026) - Vol. 1*, pages 260–269, Salvador, Brazil. Association for Computational Linguistics.

Attardo, S. (2020). *The Linguistics of Humor: An Introduction*. Oxford University Press, Oxford.

Hossain, N., Krumm, J., and Gamon, M. (2019). "president vows to cut <taxes> hair": Dataset and analysis of creative text editing for humorous headlines. In Burstein, J., Doran, C., and Solorio, T., editors, *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 133–142, Minneapolis, Minnesota. Association for Computational Linguistics.

Inacio, M. L., Wick-Pedro, G., Ramisch, R., Espírito Santo, L., Chacon, X. S. Q., Santos, R., Sousa, R., Anchiêta, R., and Goncalo Oliveira, H. (2024). Puntuguese: A corpus of puns in Portuguese with micro-edits. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13332–13343, Torino, Italia. ELRA and ICCL.

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

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

Kaufman, S., Rosset, S., Perlich, C., and Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4).

Leal, J., Inácio, M., Oliveira, H., and Anchiêta, R. (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.

Miller, T., Hempelmann, C., and Gurevych, I. (2017). SemEval-2017 task 7: Detection and interpretation of English puns. In Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S. M., Cer, D., and Jurgens, D., editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada. Association for Computational Linguistics.

Sousa, A. R. d., Sousa, C. S., Barros, C. H. S., and Anchiêta, R. T. (2026). Extending an ensemble baseline with corpus-based graph features for Portuguese pun detection. In Souza, M., de Dios-Flores, I., Santos, D., Freitas, L., Souza, J. W. d. C., and Ribeiro, E., editors, *Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026) - Vol. 1*, pages 759–769, Salvador, Brazil. Association for Computational Linguistics.

Sousa, Ê., Sousa, A., and Anchiêta, R. (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.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pretrained BERT models for Brazilian Portuguese. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems: 9th Brazilian Conference on Intelligent Systems (BRACIS 2020)*, volume 12319 of *Lecture Notes in Computer Science*, pages 403–417, Cham. Springer.