A Full Pipeline for Context-Aware Pun Generation

Marcio Lima Inácio and **Hugo Gonçalo Oliveira**
University of Coimbra
CISUC/LASI – Centre for Informatics and Systems of the University of Coimbra
Department of Informatics Engineering
Polo II, Pinhal de Marrocos
3030-290 Coimbra, Portugal
{mlinacio, hroliv}@dei.uc.pt

Abstract

Among the different forms of humor, puns are the most researched in Computational Creativity and Natural Language Generation. However, existing systems generate puns by taking a pair of ambiguous words as input, without addressing how such pairs are obtained, which disconnects the generation from any contextual background. Furthermore, the majority of the works focus on English, leaving other languages, such as Portuguese, behind. In this paper, we present a full pipeline for creating puns in Portuguese, including the creation of homographic and homophonic word pairs from news headlines. We evaluate ten different Transformer-based approaches for generating jokes given these word pairs — fine-tuned T5 models and different LLM prompting techniques — through a questionnaire with 24 participants from Brazil and Portugal, who rated the puns in terms of humor and relation to the base headline. Results suggest that including the words' definitions in the prompt can harm humor ratings and that few-shot prompting outperformed zero-shot. Additionally, the T5 model fine-tuned without word definitions produced texts that were more closely related to the base headline, but at the expense of humor.

Introduction

Humor is a fundamental aspect of human communication, playing a crucial role in social interactions (Kuipers 2008). Among the various forms of humorous content, verbal humor is especially prevalent in present days, with social relations mediated by the Web, where textual content is abundant (Chiaro 2018). Within this context, researchers have been developing methods to automatically generate verbal humorous content over the past decades.

Due to its constrained nature, most research on verbal humor has focused on puns, which are defined by Miller, Hempelmann, and Gurevych (2017) as:

> "[...] a form of wordplay in which one sign (e.g., a word or phrase) suggests two or more meanings by exploiting polysemy, homonymy, or phonological similarity to another sign, for an intended humorous or rhetorical effect."

Initially, we believed that narrowing Humor Generation to these constraints would make the task simpler. However, we found that they introduce additional challenges, as we need to first find suitable pun and alternative words to generate the puns, a problem that most previous works do not address.

Additionally, by just generating a funny text from a pair of ambiguous words, the generation process becomes detached from any context, which limits its quality and utility. Thus, we propose a full pipeline for generating punning humor that includes the creation of context-aware homographic and homophonic word pairs. Including this first step as a task on itself makes the pipeline compatible with previous works, as the created pairs can be used as inputs. Furthermore, our study focuses on the Portuguese language, which is underrepresented not only in Computational Creativity (CC) but also in Natural Language Processing (NLP) literature.

Even though we could use almost all literature models as Natural Language Generation (NLG) methods, we decided to use Transformer-based models. We fine-tuned T5 models — similar to Sun et al. (2022) — and explored different prompting techniques with Large Language Models (LLMs), namely Llama3.3 (Grattafiori et al. 2024) and Llama3.3-70B distilled from Deepseek-R1 (DeepSeek-AI et al. 2025).

We evaluated our models by generating puns from a set of 30 Brazilian and Portuguese news headlines and asking 24 native speakers to rate the generated content on humor and relation to the original headline. In sum, the main contributions of this paper are:

- A full pipeline for creating puns, including the creation of homographic and homophonic word pairs;
- An evaluation of ten different Transformer-based approaches for generating puns in Portuguese within the mentioned pipeline;
- A rule-based Phoneme-to-Grapheme (P2G) system for Portuguese, which generates all possible spellings for a given phonetic transcription.

The general results suggest that fine-tuned T5 models are unfit for generating funny puns, while still creating texts that are closely related to the input headline when not including the words' definitions in the prompt. On the other hand, LLMs presented better humor ratings in some scenarios, especially when prompted with few-shot examples. We also found that including definitions in the prompt can harm humor ratings regardless of the method used.

The remainder of the paper is organized as follows: first, we present the related work in Humor Generation; then, we describe our methodology, including the steps for creating word pairs, generating puns, and evaluating the methods. Later, we present the evaluation results and discussions. Finally, we conclude with some limitations of our work and future research directions.

Related Work

Humor Generation has been a topic of research for at least three decades, with a special focus on puns. Following the trends in NLG, the automatic creation of punning humor has been explored through various approaches, ranging from template-based systems — such as JAPE-1 (Binsted and Ritchie 1994) and T-PEG (Hong and Ong 2009) — through rewriting techniques (He, Peng, and Liang 2019; Yu, Zang, and Wan 2020), to neural networks (Yu, Tan, and Wan 2018; Luo et al. 2019; Diao et al. 2020; Mittal, Tian, and Peng 2022), and more recently LLMs (Chen et al. 2024; Inácio and Gonçalo Oliveira 2024). However, with the exception of T-PEG, which receives a keyword to be used in the joke, all of these works take as input a pair of ambiguous words to generate the puns, neglecting the process of obtaining such pairs.

One of the few works addressing the problem of obtaining input word pairs was conducted by Sun et al. (2022). By training a classifier with contextual keywords as input, they select one pair of punning and alternative words from a pre-conceived list. Yet, this approach is limited to the words in the list and does not address the automatic creation of word pairs. Besides, automatic portmanteaux generation methods could be used in such pair-creation phase, even though they are not usually mentioned within this context (Smith, Hintze, and Ventura 2014; Deri and Knight 2015; Das and Ghosh 2017; Gangal et al. 2017; Simon 2018; Vivek Kulkarni et al. 2018).

It is also worth mentioning other efforts in Humor Generation that are not limited to puns, especially those that avoid the so-called "mere generation" (Veale and Pérez y Pérez 2020) problem by incorporating contextual background into the generation process. For instance, Mendes and Gonçalo Oliveira (2020) proposed a method that adapts news headlines to generate humorous texts based on proverbial expressions. Similarly, Winters and Delobelle (2021) transformed news headlines into satirical texts using Genetic Algorithms. Another example is Witscript 2 (Toplyn 2022), which creates jokes based on a given context description. In the multimodal domain, Gonçalo Oliveira, Costa, and Pinto (2016) proposed a system that generates Internet memes from news headlines and proverbs. Although not focused on humor, Gatti et al. (2015) developed a system that does concept blending by replacing words in well-known expressions by keywords related to a set of news headlines.

Finally, we note that most of the aforementioned works are for the English language. Specifically for Portuguese, the focus of this paper, despite some work on-Humor Generation (Gonçalo Oliveira, Costa, and Pinto 2016; Gonçalo Oliveira and Rodrigues 2018; Mendes and Gonçalo Oliveira 2020; Inácio et al. 2024), it is still limited.

Methodology

We propose a full pipeline for creating puns, including the creation of pun (w_p) and alternative (w_a) word pairs, a step that has been neglected in previous works. The overall framework is depicted in Figure 1. In the following sections, we describe each step of the pipeline in more detail.

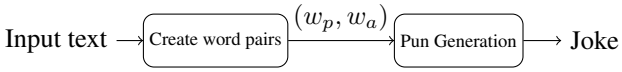

Figure 1: General framework for Pun Generation

Word Pair Creation

We create two kinds of word pairs: homographic and homophonic. For homographic pairs, w_p and w_a are written identically, without pronunciation constraints. Meanwhile, homophonic ones are pronounced the same but necessarily spelled differently. As it is known, some puns exploit words that are just similar in pronunciation or writing, but we decided to focus on these two types for simplicity.

To create the word pairs while avoiding mere generation, we first extract five keywords from each input text, in our case headlines. We then expanded this initial set by using word embeddings to find the five most similar words (based on cosine similarity) for each original keyword. Finally, word pairs are created from this expanded list of candidates.

Homographic Word Pairs The process of creating homographic word pairs is straightforward: we need to find ambiguous words among the candidates. This may resort to a dictionary or to a WordNet lexicon (Fellbaum 1998), where the definitions of all the senses of each word can be retrieved from. Since the word senses can be too similar, not providing enough semantic distinction to create a pun, we do not simply consider a word as ambiguous when it has more than one sense, e.g. the word "*caneta*" (pen) could have two different senses refering to a fountain pen and a ballpoint pen. Instead, we gather all definitions of the senses and calculate the consine similarity between them using sentence embeddings (Reimers and Gurevych 2019). If the minimum similarity between any pair of definitions is below a threshold, we consider the word as ambiguous and create its corresponding pair (w_p, w_a), where $w_p = w_a$. More details about how we set the necessary threshold can be found in the Experimental Setup section.

Homophonic Word Pairs To create homophonic word pairs, we need to add more steps to account for pronunciation. First, we used a text-to-phones conversion tool to obtain the phonetic transcription of each candidate word. Then, we use a Phoneme-to-Grapheme (P2G) system that returns all possible orthographic realizations for a given phonetic transcription. Finally, the candidates are filtered twice: first, using Phonemizer to confirm that they have the same pronunciation as the original word; and second, using the dictionary to just keep existing words.

Since a word can have multiple homophones, we must consider which ones to select for the pair. Thus, we used

the same similarity threshold strategy from the homographic word pairs, this time comparing the senses of paired homophonic words (rather than senses from the same word). Homophonic pairs with a similarity below the threshold are selected as the final output.

Pun Generation

For the Pun Generation step, we explore transformer-based models. Nonetheless, we highlight that, in our pipeline, this stage can be replaced by any other model that generates jokes from word pairs, as most models of the literature (Yu, Tan, and Wan 2018; Tian, Sheth, and Peng 2022; Chen et al. 2024).

Experimental Setup

Although this work focuses on the Portuguese language, the pipeline we propose is language-agnostic; hence, we present the specific resources and tools used in our experiments.

Word Pair Generation

As a first step, we required inputs to create the word pairs from. To this end, we used the top news headlines in Google News for Brazil and Portugal on January 30$^{\text{th}}$, 2025. For each country, we selected 15 headlines — five from each of the following categories: Science and Technology, Entertainment, and Sports — resulting in a total of 30 headlines.

For extracting keywords from the headlines, we used KeyBERT (Grootendorst 2020), with the sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 model[1] (Reimers and Gurevych 2019) and the NILC 300-dimension GloVe Embeddings (Hartmann et al. 2017) for expanding the keywords set.

When generating the word pairs, we had to determine the appropriate similarity threshold. To this end, we used PUNTUGUESE[2] (Inácio et al. 2024), a corpus of puns in Portuguese annotated with pun and alternative words. We applied the procedure described in the previous section to compute the minimum cosine similarity for homographic word pairs in the corpus. The distribution of these values can be seen in Figure 2. Based on this analysis, we set the threshold at 0.2, which lies between the mean (0.17) and the 70$^{\text{th}}$ percentile (0.23) of the distribution.

We adopted OpenWordNet-PT (OWN-PT) (de Paiva, Rademaker, and de Melo 2012) as our WordNet lexicon. To calculate the semantic similarity between word definitions, we used the multilingual SentenceTransformers model sentence-transformers/all-MiniLM-L6-v2[3]

For the creation of homophonic word pairs, we implemented our own rule-based P2G system, since, to the extent of our knowledge, there is none in Portuguese that retrieves all possible spelling representations. The P2G rules are derived from the standard Brazilian pronunciation described by Perini (2021) and Silva (2022). To validate the rules, we ensured that they could correctly recreate the vocabulary of the Floresta Sintá(c)tica corpus (Afonso et al. 2001), with the exception of some foreign names and terms, e.g. "Schwarzenegger" and "baguette". Our P2G system can correctly recreate the pronunciation for 20,213 out of 27,270 (74.1%) words in the corpus. It works by iterating through each phoneme in the transcription and mapping it into a set of possible graphemes; the final orthographic realization is obtained as the Cartesian product of all these grapheme sets. For example, the phonetic transcription of the word "*chá*" (tea) is [ʃˈa], resulting in the following mappings:

- [ʃ]: {ch, x, s, z, sh}
- [ˈa]: {a, á, à, ha, há}

As the Cartesian product can become computationally expensive for longer words, we included pruning heuristics to eliminate unlikely realizations. For instance, we discard mappings of [ʃ] to {s, z} at the beggining of words, as they only have this pronunciation at syllable endings. Another pruning used is avoiding results with "hh", as this does not happen in Portuguese orthography; additionally, "à" just exists in rather particular cases. Thus, the final writing candidates for this pronunciation are {"*sha*", "*shá*", "*cha*", "*chá*", "*xa*", "*xá*"}. After the filtering process, described in the previous section, we end up with the final homophonic word pair {"*chá*" (tea), "*xá*" (shah)}

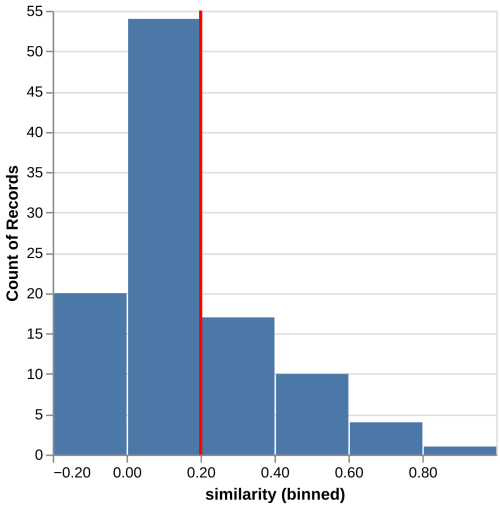

Figure 2: Distribution of cosine similarity between definitions of ambiguous words in the PUNTUGUESE corpus

Pun Generation

We evaluated ten different NLG approaches, divided into two groups: fine-tuned T5 models and different LLM prompting techniques.

Fine-tuned T5 Models The first generation approach we explored was fine-tuning a pre-trained T5 model to generate puns given the word pair under two settings: one without word definitions and one with word definitions. As a pre-trained T5 model, we adopted PTT5-v2 (Piau, Lotufo, and

[1] https://hf.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2.

[2] https://github.com/Superar/Puntuguese.

[3] https://hf.co/sentence-transformers/all-MiniLM-L6-v2.

Nogueira 2025), a Portuguese version of T5. We fine-tuned this model on PUNTUGUESE (Inácio et al. 2024).

We initially filtered the corpus to keep only homographic or homophonic puns that have exactly one pair of pun (w_p) and alternative (w_a) words, excluding jokes with multiple pairs. This filtering resulted in a total of 964 examples. For the second approach, we further filtered the corpus to include only the puns whose words are present in OWN-PT, allowing us to obtain their definitions (d_p and d_a), resulting in 207 examples.

Then, we created input commands for the model:

- `PTT5`: "*Gerar trocadilho:* w_p / w_a";
- `PTT5+def`: "*Gerar trocadilho:* w_p (d_p) / w_a (d_a)".

Since w_p and w_a may have more than one definition, we selected the definitions with the lowest cosine similarity using the sentence embeddings previously described.

For the fine-tuning, we used the stratified splits provided in the PUNTUGUESE Huggingface repository[4], keeping the previous filtering mentioned above. This resulted in 676 training, 98 validation, and 190 test examples for the `PTT5` model; and 142 training, 17 validation, and 48 test examples for the `PTT5+def` model.

Hyperparameter tuning was performed with Weights & Biases[5] via a Bayesian search for the best learning rate and batch size, minimizing validation loss across 20 runs. The optimal hyperparameters for the `PTT5` model were: a learning rate of 1.7×10^{-6} and a batch size of 16; for the `PTT5+def` model, the best hyperparameters were: a learning rate of 7.9×10^{-6} and a batch size of 16. Both models were fine-tuned for a maximum of 200 epochs, with early stopping if the validation loss did not improve for three consecutive evaluations. The generation sampling was done with a temperature of 0.6 and top-p of 0.95.

LLM Prompting Techniques Since LLMs are general-purpose, instruction-based models, we did not carry out fine-tuning. Instead, we relied entirely on prompting:

- `Llama3.3` and `Deepseek`: Zero shot, with a prompt including the task definition and the word pair;
- `Llama3.3+def` and `Deepseek+def`: Zero shot, with a prompt including the task definition, word pair, and the definitions of the words;
- `Llama3.3+shot` and `Deepseek+shot`: Few shot (using 4 random examples from PUNTUGUESE, 2 of each type), with a prompt including the task definition and the word pair;
- `Llama3.3+shot+def` and `Deepseek+shot+def`: Few shot, with a prompt including the task definition, word pair, and definitions of the words.

For LLM prompting, we used the Ollama tool[6] with two state-of-the-art models: Llama3.3 (`Llama3.3`) (Grattafiori et al. 2024) and Llama3.3 distilled from Deepseek-R1 (`Deepseek`) (DeepSeek-AI et al. 2025), each with 70B parameters, temperature 0.6, and top-p 0.95. Full prompts (in Portuguese) for each technique, as well as the selected few-shot examples, can be seen in Table 1. Each prompt is preceeded by the following system prompt:

> *Você é um gerador de piadas baseado em trocadilhos em português. Sua tarefa é criar piadas curtas e engraçadas baseadas nos pares de palavras fornecidos. Crie os trocadilhos com base nos exemplos. Apresente o resultado exclusivamente em formato JSON contento duas chaves: "palavras" e "trocadilho", contendo as palavras fornecidas e a piada criada, respectivamente. Retorne apenas o objeto JSON.* (You are a pun-based joke generator in Portuguese. Your task is to create short and funny jokes based on the word pairs provided. Create the puns based on the examples. Present the result exclusively in JSON format containing two keys: "words" and "pun", containing the words provided and the joke created, respectively. Return only the JSON object.)

All prompts are given in Portuguese except for the definitions, which are in English because the OWN-PT defintions used in our final experiments are in English. The few-shot definitions were manually written by the authors based on the corresponding entries in the Priberam dictionary[7].

Evaluation

To evaluate the models, we first generate puns from a set of 30 headlines, as descibed in the Word Pair Creation section. Three of the headlines did not result in any word pair, i.e. no ambiguous words were found. The remaining 27 headlines resulted in 124 word pairs, which served as inputs for the Pun Generation models. This process resulted in 1,240 puns, from which we select the examples for evaluation.

Note that, for the same headline, different pairs can be obtained, thus resulting in different puns. This is frequent when the input text has an ordinal, which can be a fraction, and is semantically-related to other ordinals. One of such cases in the used headlines happens for "*E assim foi o terceiro episódio de A Herança*" (And so was the third episode of "The Inheritance"), illustrated with three of the eight examples of word pairs and respective jokes by `Llama3.3+def`:

- *quarto*, *quarto* (fourth, bedroom): P*or que o quarto da casa foi ao médico? Porque estava com um 'quarto' de litro de febre!* (Why did the room in the house go to the doctor? Because it had a 'quarter' of a liter of fever!)
- *quinto*, *quinto* (fifth): *Por que o quinto elemento foi ao quinto andar? Porque estava procurando por sua fração perfeita em uma série de eventos!* (Why did the fifth element go to the fifth floor? Because it was looking for its perfect fraction in a series of events!)
- *cesto*, *sexto* (basket, sixth): *Eu tentei encontrar o cesto, mas ele estava no sexto lugar da lista de prioridades!* (I tried to find the basket, but it was sixth on the priority list!)

[4] `https://hf.co/datasets/Superar/Puntuguese`.

[5] `https://wandb.ai/`.

[6] `https://ollama.com/`.

[7] `https://dicionario.priberam.org/`.

Model type	Prompt
Zero-shot	Palavras: "w_p" e "w_a"
Zero-shot with definitions	Palavras: "w_p" e "w_a" Definições: "d_p" e "d_a"
Few-shot	Palavras: "entre-meada" e "entre miada" {"palavras": ["entre-meada", "entre miada"], "trocadilho": "O que é que está entre dois gatos? Uma entre-meada."} Palavras: "flora" e "flora" {"palavras": ["flora", "flora"], "trocadilho": "Foi lançada uma nova manteiga com sabor a merda. Chama-se Flora Intestinal."} Palavras: "shanti-lee" e "chantilly" {"palavras": ["shanti-lee", "chantilly"], "trocadilho": "Que nome se dá a uma chinesa muito docinha? Shanti-Lee."} Palavras: "paciente" e "paciente" {"palavras": ["paciente", "paciente"], "trocadilho": "O que o médico disse para o homem que entrou gemendo de dor no hospital? "Seja paciente"."} Palavras: "w_p" e "w_a"
Few-shot with definitions	Palavras: "entre-meada" e "entre miada" Definições: "Part of the pork meat consisting of bacon with bacon in between" e "Between the voice of the cat" {"palavras": ["entre-meada", "entre miada"], "trocadilho": "O que é que está entre dois gatos? Uma entre-meada."} Palavras: "flora" e "flora" Definições: "Set of plants from a region, an environment or a geological period" e "Butter brand" {"palavras": ["flora", "flora"], "trocadilho": "Foi lançada uma nova manteiga com sabor a merda. Chama-se Flora Intestinal."} Palavras: "shanti-lee" e "chantilly" Definições: "Chinese-sounding name" e "Cream made from whipped cream and sugar" {"palavras": ["shanti-lee", "chantilly"], "trocadilho": "Que nome se dá a uma chinesa muito docinha? Shanti-Lee."} Palavras: "paciente" e "paciente" Definições: "Who or what has patience" e "Any person undergoing medical treatment or care" {"palavras": ["paciente", "paciente"], "trocadilho": "O que o médico disse para o homem que entrou gemendo de dor no hospital? "Seja paciente"."} Palavras: "w_p" e "w_a" Definições: "d_p" e "d_a"

Table 1: Prompts used for Pun Generation with LLMs. Variables w_p, w_a, d_p, and d_a represent the pun and alternative words, and their definitions, respectively.

To maintain a feasible evaluation process, we selected, for each headline and model, one pun according to an automatic scoring, regardless of the source pair. With this score, we selected 270 puns (10 per headline, one for each model) for evaluation. The score was computed as a weighted combination (each with a weight of 0.5) of two metrics:

- Typicality: The classification score by a Humor Recognition model trained on PUNTUGUESE (Inácio et al. 2024);
- Semantic Similarity: The cosine similarity between the the embeddings of the generated pun and the headline.

The evaluation was carried out by 24 participants — 14 from Brazil and 10 from Portugal — all native speakers of their respective varieties of Portuguese. Each evaluator received a set of headlines from their country and evaluated all associated jokes for each headline. The sets were split across the evaluators so that puns for each headline were evaluated by exactly 4 participants.

Participants rated each generated text based on two criteria: humor and relation to the base headline, using a 3-point scale. For humor, the scale was: "*Não tem piada*" (Not funny), "*Tem pouca piada*" (A bit funny), and "*Tem piada*" (Funny). For relation to the headline, the ratings were: "*Não tem relação*" (Not related), "*Tem pouca relação*" (A bit related), and "*Tem relação*" (Related).

Results

Before presenting the evaluation results, we depict some examples of generated text for each approach in Table 2. Each example was randomly selected from the jokes of its corresponding model with a majority rating of "Funny" from the evaluators. From these examples, we can see that the models are indeed generating puns, according to the definition of Miller, Hempelmann, and Gurevych (2017) mentioned in the Introduction, as they include words that evoke more than one meaning even if not necessarily the input word pair. For example, the joke by `Deepseek+shot+def` has both

Model	Input Headline	Generated Text	Pair	Fun.	Rel.
PTT5	***Starlink Direct to Cell passa a disponibilizar 4G via satélite para todos os smartphones.*** **(Starlink Direct to Cell now offers 4G via satellite for all smartphones.)**	*Qual é o satélite que mais se aproxima do homem? O satélite.* (What is the satellite that is closest to the man? The satellite.)	*satélite* (satellite) *satélite* (follower)	3	2
PTT5 +def	***Starlink Direct to Cell passa a disponibilizar 4G via satélite para todos os smartphones.*** **(Starlink Direct to Cell now offers 4G via satellite for all smartphones.)**	*Por que o homem não tem um satélite? Porque não tem um satélite.* (Why doesn't the man have a satellite? Because he doesn't have a satellite.)	*satélite* (satellite) *satélite* (follower)	3	2
Llama3.3	***Vodafone realiza primeira videochamada espacial do mundo com smartphone.*** **(Vodafone makes the world's first space video call with a smartphone.)**	*Por que o mundo foi ao psicólogo? Porque estava se sentindo um pouco 'girado'!* (Why did the world go to the psychologist? Because it was feeling a bit 'spun'!)	*mundo* (populace) *mundo* (world)	3	2
Llama3.3 +def	***"Não daria à minha filha um vibrador aos 12 anos. Não posso é condenar".*** **("I wouldn't give my daughter a vibrator at 12. I just can't condemn it.")**	*Eu condeno o uso excessivo de 'condenar', mas não posso condenar alguém por concordar!* (I condemn the excessive use of 'condemn', but I can't condemn someone for agreeing!)	*condenar* (condemn) *condenar* (convict)	3	2
Llama3.3 +shot	***Liga Europa: o onze provável do FC Porto na estreia de Martín Anselmi.*** **(Europa League: FC Porto's probable eleven in Martín Anselmi's debut.)**	*Por que a lâmpada foi ao terapeuta? Ela estava com problemas para se ligar.* (Why did the lamp go to the therapist? She was having trouble turning on.)	*liga* (elastic) *liga* (league)	3	1
Llama3.3 +shot +def	***O preço do Samsung Galaxy S25 cai a pique com os descontos na Samsung.*** **(The price of the Samsung Galaxy S25 drops sharply with discounts at Samsung.)**	*Por que o vendedor usava uma venda nos olhos? Para não ver a venda cair!* (Why did the seller wear a blindfold? So he wouldn't see the sale fall!)	*venda* (sale) *venda* (blindfold)	3	2
Deepseek	***Restos de salmoura no asteroide Bennu contêm minerais essenciais para a vida.*** **(Brine residues on asteroid Bennu contain essential minerals for life.)**	*O asteróide foi ao terapeuta porque estava se sentindo um pouco 'defasado'.* (The asteroid went to the therapist because it was feeling a bit 'outdated'.)	*asteróide* (asteroid) *asteroide* (asteroid)	3	3
Deepseek +def	***E assim foi o terceiro episódio de "A Herança".*** **(And so was the third episode of "The Inheritance".)**	*Por que a carteira e o DNA foram juntos ao terapeuta? Porque um carrega a herança material e o outro, a herança genética!* (Why did the wallet and the DNA go to the therapist together? Because one carries the material inheritance and the other, the genetic inheritance!)	*herança* (heritage) *herança* (heredity)	3	2
Deepseek +shot	***Restos de salmoura no asteroide Bennu contêm minerais essenciais para a vida.*** **(Brine residues on asteroid Bennu contain essential minerals for life.)**	*O que é um asteróide? É um asteroide sem sotaque.* (What is an asteroid? It's an asteroid without an accent.)	*asteróide* (asteroid) *asteroide* (asteroid)	3	3
Deepseek +shot +def	***Restos de salmoura no asteroide Bennu contêm minerais essenciais para a vida.*** **(Brine residues on asteroid Bennu contain essential minerals for life.)**	*Por que o asteroide foi ao festival de música? Porque ele amava rock e metal!* (Why did the asteroid go to the music festival? Because it loved rock and metal!)	*asteróide* (asteroid) *asteroide* (asteroid)	3	3

Table 2: Examples of funniest texts generated by each model. Input headlines are in bold.

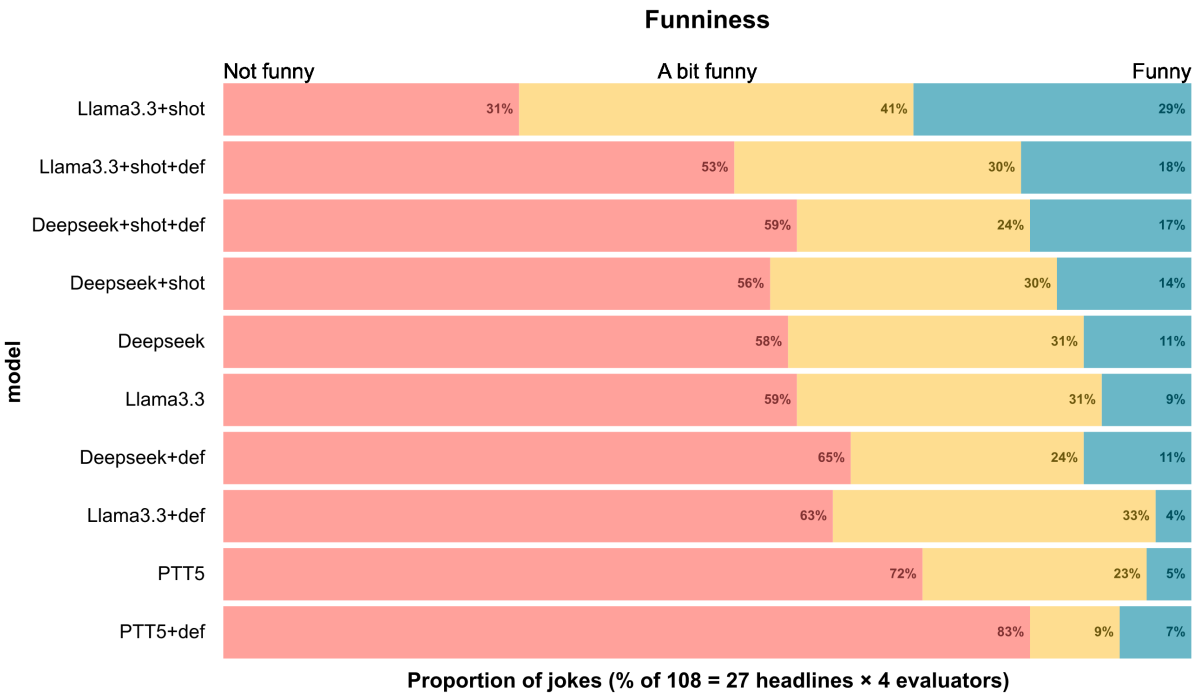

Figure 3: Evaluation distribution for the Funniness scale

"rock" and "metal" as ambiguous words, but the input pair was "*asteróide*" and "*asteroide*" (both meaning asteroid).

Another joke that derived from this same pair of words is the one by `Deepseek+shot`, which lead to an interesting observation about this specific model. Deepseek-R1 has a built-in Chain-of-Thought (CoT) mechanism that always generates a thought process in English, even if the final output is in another language. When observing the CoT output for this joke, we see that the model correctly identified the difference in the accent mark of the dipthong ("*oi*" vs. "*ói*") and decided to make a joke about it. However, the final output had a mistranslation for the word "accent", which was translated as "*sotaque*" (a regional form of pronunciation), while the intended meaning (from the CoT) would be "*acento*" (an accent mark). This "error", nonetheless, might have resulted in a funnier joke than the original intention as it created a more playful scenario with talking asteroids.

On the other hand, we can observe that LLMs also rely on the so-called "lazy pun pattern", as described by Xu et al. (2024), in which the text is composed by forcing two occurences of the pun word with different meanings, rather than just implying one of the senses. For instance, the pun created by `Llama3.3+shot+def` has the word "*venda*" sale or blindfold) repeated twice, with a clear distinction between the two meanings. Another example is the joke from `Deepseek+def`, which explicitly adds adjectives to limit the semantic interpretations of the occurrences of the word "*herança*" (inheritance or heredity).

For a quantitative analysis of the results, we plot the distribution of all 1,080 evaluations (27 headlines $\times$ 10 models $\times$ 4 evaluators) for the Funniness scale in Figure 3. We see that `Llama3+shot` is the only model that reached less than 50% of the evaluations as "Not funny", being consistently funnier than the other models. From the distributions, we can also notice that including the definitions (`+def`) consistently harmed the funniness of the generated texts, regardless of the model, resulting in higher proportions of the "Not funny" category. The fact that the definitions from are in English might have influenced these results.

For LLMs, we can see that the few-shot approach improved funniness regardless of the model or the presence of word definitions, especially for `Llama3.3` — which had an increase of 164% in the proportion of "Funny" texts (from 11% to 29%) — and `Llama3.3+def`, with a 350% "Funny" text increase (from 4% to 18%).

Concerning the fine-tuned models, `PTT5` and `PTT5+def` both had the highest proportion of "Not funny" ratings, 72% and 83% respectively. This shows that the fine-tuning was not enough for the model to acquire humor-production capabilities, being limited to produce texts that resemble jokes in the superficial form. On the other hand, regarding the relation to the headline (Figure 4), `PTT5` had the highest proportion of "A bit related" (41%) and "Related" (11%) ratings.

It is notable that including the definitions in `PTT5`'s input harmed the relation to the headline; we believe that this is due to our definition-selection process, which might have selected definitions that are not the most common or the most related to the headline. We need, however, to further investigate this hypothesis.

Observing the overall relation distributions in Figure 4,

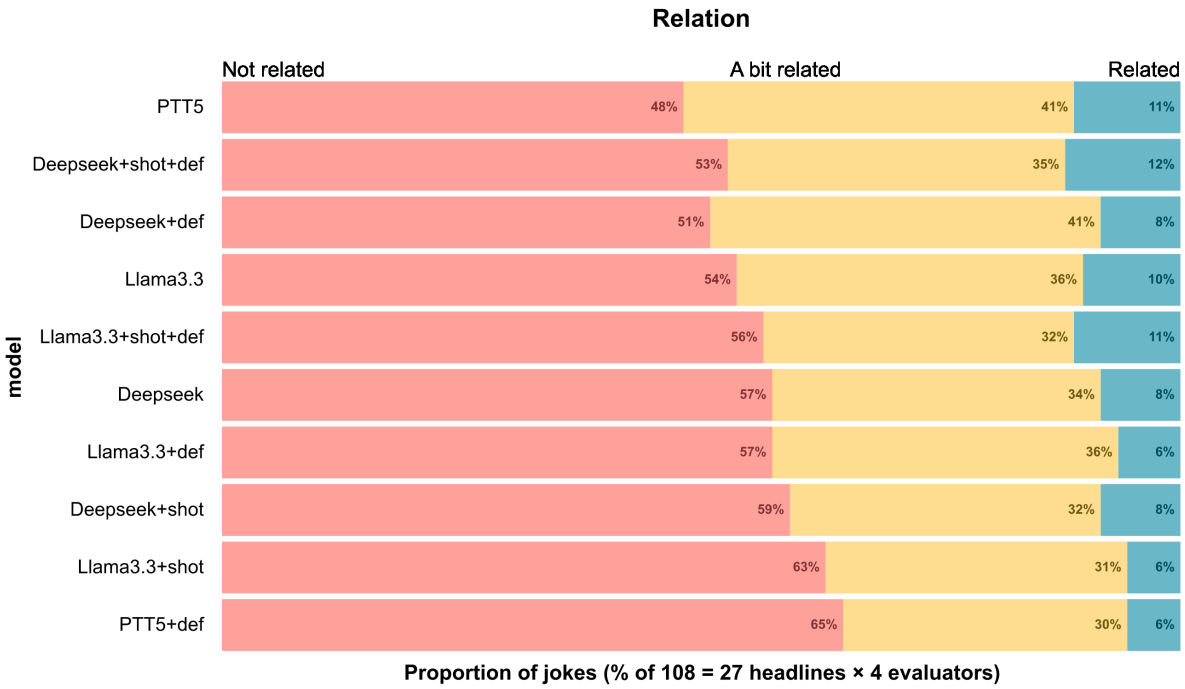

Figure 4: Evaluation distribution for the Relation scale

we can see that the models did not perform well in creating texts related to the headline, with very few ratings as "Related". By comparing the effects of including the definitions into the input, it mostly harms the results, with the exception of the Deepseek-R1-based approaches. This may be due to its built-in CoT mechanism, which could potentially bring up other definitions rather than simply relying on the ones provided; a more in-depth analysis of the CoT outputs could be helpful in testing this hypothesis.

To understand how consistent the ratings are for each model, we calculated the agreement scores of the evaluations for each model. To do so, we chose the Krippendorff's α coefficient (Krippendorff 2019) as it can handle multiple raters and missing data (as not every evaluator rated every text). The results are presented in Table 3.

	Krippendorff's α	
Model	Funniness	Relation
`PTT5`	-0.05	0.46
`PTT5+def`	0.07	0.44
`Llama3.3`	0.08	0.41
`Llama3.3+def`	0.09	0.33
`Llama3.3+shot`	0.02	0.18
`Llama3.3+shot+def`	**0.26**	0.40
`Deepseek`	0.00	**0.54**
`Deepseek+def`	0.20	0.31
`Deepseek+shot`	0.13	0.38
`Deepseek+shot+def`	0.21	0.28

Table 3: Krippendorff's α values for Funniness and Relation

As expected from such a subjective task, agreement scores for Funniness are low, with the highest value being 0.26 for `Llama3.3+shot+def`. This shows that, although having higher proportions of "A bit funny" and "Funny" ratings, the humor created by `Llama3.3+shot` did not consistently please the same way ($\alpha = 0.02$). When including the definitions (`Llama3.3+shot+def`), the agreement scores are a little higher (0.26), but still not enough to consider the evaluations as consistent.

On the other hand, agreement scores for Relation are higher, with the most consistent model being `Deepseek` ($\alpha = 0.54$). Curiously, `Llama3.3+shot` had the lowest agreement score for this scale ($\alpha = 0.18$), from 27 jokes, 19 of them had at least one evaluator desagreeing with the others. This indicates how even this task of evaluating semantic relationship can have some degree of subjectivity, especially when dealing with texts close to non-sense (as most of the generated jokes); for example, the following joke generated by `Llama3.3+shot` had all 3 different Relation ratings (2 "Not related", 1 "A bit related", and 1 "Related"):

> **Headline:** *Starlink Direct to Cell passa a disponibilizar 4G via satélite para todos os smartphones* (Starlink Direct to Cell now offers 4G via satellite to all smartphones)
>
> **Joke:** *Por que o planeta foi ao terapeuta? Ele estava se sentindo um pouco terrestre, precisava de ajuda para se sentir mais 'em terra'.* (Why did the planet go to the therapist? It was feeling a bit earthy, needed help to feel more 'down to earth'.)

Finally, we evaluated if the simple automatic scoring we used to select the best puns when there were multiple options

for a headline was consistent with the human evaluations. Thus, we calculated the Pearson's correlation between both scores (Typicality and Semantic Similarity) and the Funniness and Relation ratings; additionally, we included the correlation between the two human ratings. The results are depicted in Table 4.

Metric	Funniness	Relation
Typicality	-0.02	-0.06
Semantic Similarity	0.06	0.15
Funniness	—	0.28

Table 4: Pearson's correlation between Typicality, Semantic Similarity, Funniness, and Relation

The correlation analysis shows that the automatic scoring we used might not be the best-suited for the task, as it had very low correlation with the human ratings, especially between Typicality and Funniness. For Relation and Semantic Similarity, the correlation was higher, but still not enough to consider the automatic scoring as a good representation of the human evaluations. Regarding the ratings for Funniness and Relation, the correlation was 0.28, indicating a slight connection between the context and the humor effect, but further analysis must be carried out to confirm this intuition.

Conclusions and Future Work

We presented a complete pipeline for generating puns, from the creation of word pair to the generation of final punning texts. We provide a straightforward methodology to create homographic and homophonic word pairs based on the similarity of their lexical definitions. To evaluate our approach, we implemented it in Portuguese. We also developed a Phoneme-to-Grapheme system to generate orthographic alternatives in Portuguese, which can be used not only in creating puns but also in projects that require spelling variations; for example, a funny translator-like platform[8] that gives a non-canon spelling for a given sentence. All code for our experiments, including the P2G system, are publicly available at `https://github.com/NLP-CISUC/full-pun-generation/`.

As for the pun generation step, we explored 10 different approaches, including fine-tuned T5 models and state-of-the-art LLMs. We also tested the impact of including the definitions of the words in the input, as well as providing few-shot examples. To the extent of our knowledge, this is the first work to use LLMs for context-aware Pun Generation in Portuguese.

The evaluation was carried out by 24 native speakers of Portuguese, from Brazil and Portugal, who rated the generated texts based on two criteria: funniness and relation to the headline from which the word pairs were created. Results showed that the LLMs consistently outperformed the fine-tuned models on funniness, with Llama3.3 with few-shot being considered "A bit funny" or "Funny" most of the times. Other approaches, however, are mostly "Not funny". The inclusion of definitions in the input harmed the funniness ratings of the generated texts, regardless of the model, but it did not have a significant impact on the relation to the headline, which is not exceptionally good in any model.

As improvement, we point out the need to use word definitions in Portuguese — either through Machine Translation, bilingual dictionaries, or LLMs — guaranteeing consistency within the prompts. For LLMs, we could use the original headline to ensure a stronger relation with the context, while exploring more techniques, such as prompt chaining or agentic prompting. We also need to further investigate the impact of the word pair creation process on the final puns, especially whether it provides more control over what parts of the context should be considered during generation. Additionally, the word pair creation can be expanded to include other types of punning phenomena, such as lexical blending or paronymy.

Finally, the evaluation can be expanded by, not only including more people to account for multiple senses of humor, but also by collecting sociodemographic data (Eiselen and van Huyssteen 2023) to analyze if the models are considered funny for any specific type of audience (for example, people who like dad-jokes). We also intend to further analyze the outputs of the CoT mechanism in Deepseek-R1, as it can give interesting insights about how the concepts are connected during the process of creating a suitable pun.

Limitations

We acknowledge that the main limitation of this work is the lack of comparison of the impact of the word pair creation step in the pipeline, especially on headline relation. However, we emphasize that adressing the challenge of obatining these pairs is already a significant contribution to the field, as most existing works on pun generation do not deal with this issue, despite relying on such pairs as input. Moreover, we believe that including the word pair creation step can provide more control and interpretability over the generated puns.

Furthermore, when combining the Typicality and Semantic Similarity scores, we did not perform weight optmization; instead, both metrics had an equal weight of 0.5. It is possible that different weights would lead to different selections. Moreover, we acknowledge that it would be ideal to compare the models' jokes with human-created ones, whether produced by professionals or laypeople, to have an upper bound on the expected performance. Finally, we mention that the generated jokes could not be entirely new, as we do not assess novelty and originality in this paper.

Acknowledgments

We thank all volunteering participants who took part in our evaluation process. This work was supported by the Portuguese Recovery and Resilience Plan through project C645008882-00000055, Center for Responsible AI; and by national funds through FCT – Foundation for Science and Technology I.P. (grant number UI/BD/153496/2022), in the framework of the Project UIDB/00326/2025 and UIDP/00326/2025.

[8]Inspired by MiGuXeiToR: `https://aurelio.net/coisinha/miguxeitor/`.

References

Afonso, S.; Bick, E.; Haber, R.; and Santos, D. 2001. Floresta Sintá(c)tica: um "treebank" para o português. In Conçalves, A., and Correia, C. N., eds., *Actas do XVII Encontro Nacional da Associação Portuguesa de Linguística (APL 2001)*. Lisbon: APL.

Binsted, K., and Ritchie, G. 1994. An implemented model of punning riddles. In Hayes-Roth, B., and Korf, R. E., eds., *Proceedings of the 12th National Conference on Artificial Intelligence*, volume 1, 633–638. Seattle: AAAI Press / The MIT Press.

Chen, Y.; Yang, C.; Hu, T.; Chen, X.; Lan, M.; Cai, L.; Zhuang, X.; Lin, X.; Lu, X.; and Zhou, A. 2024. Are U a joke master? Pun generation via multi-stage curriculum learning towards a humor LLM. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 878–890. Bangkok, Thailand and virtual meeting: ACL.

Chiaro, D. 2018. *The Language of Jokes in the Digital Age: Viral Humour*. New York: Routledge.

Das, K., and Ghosh, S. 2017. Neuramanteau: A Neural Network Ensemble Model for Lexical Blends. 576–583.

de Paiva, V.; Rademaker, A.; and de Melo, G. 2012. OpenWordNet-PT: An Open Brazilian Wordnet for Reasoning. In *Proceedings of COLING 2012: Demonstration Papers*, 353–360. Mumbai: The COLING 2012 Organizing Committee.

DeepSeek-AI; Guo, D.; Yang, D.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. In *arXiv:2501.12948*. arXiv.

Deri, A., and Knight, K. 2015. How to Make a Frenemy: Multitape FSTs for Portmanteau Generation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 206–210. Denver, Colorado: Association for Computational Linguistics.

Diao, Y.; Yang, L.; Fan, X.; Chu, Y.; Wu, D.; Zhang, S.; and Lin, H. 2020. AFPun-GAN: Ambiguity-Fluency Generative Adversarial Network for Pun Generation. In Zhu, X.; Zhang, M.; Hong, Y.; and He, R., eds., *Natural Language Processing and Chinese Computing*, volume 12430. Cham: Springer International Publishing. 604–616.

Eiselen, R., and van Huyssteen, G. B. 2023. A Comparison of Statistical Tests for Likert-Type Data: The Case of Swearwords. *Journal of Open Humanities Data*.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. Cambridge: The MIT Press.

Gangal, V.; Jhamtani, H.; Neubig, G.; Hovy, E.; and Nyberg, E. 2017. CharManteau: Character Embedding Models For Portmanteau Creation. *arXiv: Computation and Language*.

Gatti, L.; Özbal, G.; Guerini, M.; Stock, O.; and Strapparava, C. 2015. Slogans Are Not Forever: Adapting Linguistic Expressions to the News. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2452–2458. Buenos Aires: AAAI Press.

Gonçalo Oliveira, H., and Rodrigues, R. 2018. Exploring Lexical-Semantic Knowledge in the Generation of Novel Riddles in Portuguese. In *Proceedings of the 3rd Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2018)*, 17–25. Tilburg, the Netherlands: Association for Computational Linguistics.

Gonçalo Oliveira, H.; Costa, D.; and Pinto, A. M. 2016. One Does Not Simply Produce Funny Memes! - Explorations on the Automatic Generation of Internet Humor. In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Proceedings of the Seventh International Conference on Computational Creativity*, 238–245. Paris: Sony CSL Paris, France.

Grattafiori, A.; Dubey, A.; Jauhri, A.; et al. 2024. The Llama 3 Herd of Models. In *arXiv:2407.21783*. arXiv.

Grootendorst, M. 2020. KeyBERT: Minimal keyword extraction with BERT. In *Zenodo*. v0.3.0.

Hartmann, N.; Fonseca, E.; Shulby, C.; Treviso, M.; Rodrigues, J.; and Aluísio, S. 2017. Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. In Pinheiro, V., and Paetzold, G. H., eds., *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, 122–131.

He, H.; Peng, N.; and Liang, P. 2019. Pun Generation with Surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, 1734–1744. Minneapolis: ACL.

Hong, B. A., and Ong, E. 2009. Automatically Extracting Word Relationships as Templates for Pun Generation. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 24–31. Boulder: ACL.

Inácio, M., and Gonçalo Oliveira, H. 2024. Generation of Punning Riddles in Portuguese with Prompt Chaining. In *Proceedings of the 15th International Conference on Computational Creativity (ICCC'24)*. Jönköping: ACC.

Inácio, 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., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 13332–13343. Torino: ELRA and ICCL.

Krippendorff, K. 2019. *Content Analysis: An Introduction to Its Methodology*. 2455 Teller Road, Thousand Oaks California 91320: SAGE Publications, Inc.

Kuipers, G. 2008. The Sociology of Humor. In *The Primer of Humor Research*, number 8 in Humor Research. Berlin, New York: Victor Raskin. 361–398.

Luo, F.; Li, S.; Yang, P.; Li, L.; Chang, B.; Sui, Z.; and Sun, X. 2019. Pun-GAN: Generative Adversarial Network for Pun Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

Processing (EMNLP-IJCNLP), 3386–3391. Hong Kong: ACL.

Mendes, R., and Gonçalo Oliveira, H. 2020. TECo: Exploring Word Embeddings for Text Adaptation to a given Context. In Cardoso, F. A.; Machado, P.; Veale, T.; and Cunha, J. M., eds., *Proceedings of the Eleventh International Conference on Computational Creativity*, 185–188. Coimbra: ACC.

Miller, T.; Hempelmann, C.; and Gurevych, I. 2017. SemEval-2017 Task 7: Detection and Interpretation of English Puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 58–68. Vancouver: ACL.

Mittal, A.; Tian, Y.; and Peng, N. 2022. AmbiPun: Generating Humorous Puns with Ambiguous Context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1053–1062. Seattle: ACL.

Perini, M. A. 2021. *Gramática descritiva do português brasileiro*. Editora Vozes.

Piau, M.; Lotufo, R.; and Nogueira, R. 2025. Ptt5-v2: A Closer Look at Continued Pretraining of T5 Models for the Portuguese Language. In Paes, A., and Verri, F. A. N., eds., *Intelligent Systems*, volume 15413. Cham: Springer Nature Switzerland. 324–338.

Reimers, N., and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3980–3990. Hong Kong: ACL.

Silva, T. C. 2022. *Fonetica e Fonologia do Português*. São Paulo: Editora Contexto, 7 edition.

Simon, J. A. 2018. Entendrepeneur: Generating humorous portmanteaus using word embeddings. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*.

Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A Neologism Creator Nomen Ipsum. 173–181.

Sun, J.; Narayan-Chen, A.; Oraby, S.; Gao, S.; Chung, T.; Huang, J.; Liu, Y.; and Peng, N. 2022. Context-Situated Pun Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 4635–4648. Abu Dhabi: ACL.

Tian, Y.; Sheth, D.; and Peng, N. 2022. A Unified Framework for Pun Generation with Humor Principles. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 3253–3261. Abu Dhabi: Association for Computational Linguistics.

Toplyn, J. 2022. Witscript 2: A System for Generating Improvised Jokes Without Wordplay. In *International Conference on Computational Creativity*. Bolzano-Bozen: ACC.

Veale, T., and Pérez y Pérez, R. 2020. Leaps and Bounds: An Introduction to the Field of Computational Creativity. *New Generation Computing* 38(4):551–563.

Vivek Kulkarni; Kulkarni, V.; William Yang Wang; and Wang, W. Y. 2018. Simple Models for Word Formation in Slang. 1:1424–1434.

Winters, T., and Delobelle, P. 2021. Survival of the Wittiest: Evolving Satire with Language Models. In *Proceedings of the Twelfth International Conference on Computational Creativity*, 82–86. Mexico City: ACC.

Xu, Z.; Yuan, S.; Chen, L.; and Yang, D. 2024. "A good pun is its own reword": Can Large Language Models Understand Puns? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 11766–11782. Miami: ACL.

Yu, Z.; Tan, J.; and Wan, X. 2018. A Neural Approach to Pun Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1650–1660. Melbourne: ACL.

Yu, Z.; Zang, H.; and Wan, X. 2020. Homophonic Pun Generation with Lexically Constrained Rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2870–2876. Online: ACL.