Instituto Superior
de Engenharia
Politécnico de Coimbra

DEPARTMENT OF SYSTEMS AND COMPUTER ENGINEERING

Sequence Labeling for Pun Location and Detection in Portuguese

Dissertation to fulfill the Master's degree in Informatics Engineering

Specialization in Software Engineering

Author

Patrícia Moura Gameiro

Supervisor

Ana Cristina Costa Oliveira Alves

Co-Supervisor

Hugo Gonçalo Oliveira

Coimbra, Setembro 2024

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Resumo

Detectar humor é um passo necessário para a compreensão da linguagem. No entanto, o trabalho sobre humor computacional para português ainda é limitado. Para este idioma, abordamos a tarefa de localização de trocadilhos. Com um *dataset* de textos anotados com trocadilhos, realizámos o *fine-tuning* de modelos de linguagem para categorizar palavras, num dado contexto, como trocadilhos ou não. A categorização foi realizada através de uma abordagem com *sequence labeling*. Alcançámos uma medida F1 de 0,75 no modelo de linguagem BERT e mostramos que é possível melhorar a sua precisão com pós-processamento. Além disso, mostrámos que um modelo treinado para localização de trocadilhos pode ser usado também para deteção de trocadilhos, atingindo um desempenho quase tão bom como um modelo treinado especificamente para esta última tarefa, mas com a vantagem de identificar as palavras do trocadilho, contribuindo assim para a explicabilidade do humor. Foi também explorada a capacidade de generalização dos modelos de linguagem para português de Portugal e português do Brasil, ainda que com resultados pouco conclusivos.

Palavras-chave: Humor Computacional, Localização de trocadilhos, Deteção de trocadilhos, Desambiguação de trocadilhos, Processamento Computacional de Português

Abstract

Detecting humor is a necessary step towards language understanding. However, work on computational humor for Portuguese is still limited. For this language, we tackle the task of pun location. With a corpus of annotated punning texts, we fine-tune available language models for labeling words in context as punning or not. We achieve an F1 of 0.75 with a BERT-based model and further improve precision with post-processing. Moreover, we show that a model trained for pun location can be used for pun detection as well, performing close to a model specifically trained on the latter task, but with the advantage of identifying the pun words, thus contributing to explainability. We also explored the language models' capacity to generalize across different language variants, yet with inconclusive results.

Keywords:

Computational Humor, Pun location, Pun detection, Pun disambiguation, Computational Processing of Portuguese

Epígrafe

Estão duas bolas de berlim à beira de um precipício, e uma delas pergunta:
- Não tens medo?
-Não é bem medo, é recheio.

Agradecimentos

A conclusão desta dissertação marca o final de uma jornada de profunda mudança profissional, iniciada há quatro anos atrás e feita com o apoio e incentivo de várias pessoas na minha vida, às quais gostaria de agradecer.

Em primeiro lugar aos meus orientadores: À professora Doutora Ana Oliveira, por acreditar em mim, pela incrível disponibilidade, atenção e empatia ao longo de todo o processo; ao professor Doutor Hugo Gonçalo Oliveira, pela motivação, atenção ao detalhe e pela capacidade de, nos momentos certos, trazer leveza e humor, não fôssemos nós esquecer-nos de qual era o tema; e ao Márcio Lima, que, apesar de não ser orientador oficial, foi imprescindível em todas as fases, ajudando sempre que necessário e partilhando ideias e conhecimentos determinantes na concretização deste trabalho.

À minha mãe, que esteve sempre ao meu lado e que me ajudou em tudo o que foi necessário, assegurando que não faltava nada, nem a mim nem aos seus netos.

Ao meu namorado e melhor amigo, Wilson, por me mostrar que sou capaz de mais do que julgava e que não há caminhos lineares na vida. Deste-me sempre força quando precisei, apaziguaste dúvidas e incertezas, estiveste presente e apoiaste-me incansavelmente ao longo de todos estes anos de mudança. Sem ti, não só não haveria esta dissertação, como não também haveria "divlóper".

Por último, a todas as pessoas que de alguma forma marcaram este caminho e me ajudaram a chegar aqui, aos amigos de sempre e às novas pessoas que esta área me trouxe. Foi um caminho difícil mas bonito e recompensador, que me ensinou que não há destinos escritos na pedra e que se está sempre a tempo de começar de novo, seja em que momento for.

Index

Index of tables

Index of figures

List of acronyms

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
CRF	Conditional Random Fields
DT	Decision Trees
GPT	Generative Pre-trained Transformer
IDF	Inverse Document Frequency
K-NN	K-Nearest Neighbors
LLM	Large Language Model
LS	Last Sequence
LSTM	Long Short-Term Memory
LW	Last Word
MLP	Multilayer Perceptron
NB	Naïve Bayes
NER	Named Entity Recognition
NH	Non-Humorous
NLP	Natural Language Processing
PMI	Pointwise Mutual Information
POS	Part-of-Speech Tagging
RF	Random Forest
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
WSD	Word Sense Disambiguation

1 Introduction

Humor is a complex and context-dependent aspect of human communication. Since using and understanding it denotes fluency in the target language (Tagnin, 2005), the automatic detection of humor has been tackled in the scope of Natural Language Processing (NLP) to further push research toward more complex linguistic phenomena. As the use of machine learning and language models becomes more widespread, enabling systems to detect humor can have truly impactful outcomes, as evidenced by a documented failure of stock market algorithms to interpret an April's Fool joke press release (Scholer, 2015).

While there is diverse published work in computational humor, including humor recognition and pun related tasks, in various languages, research in Portuguese has primarily focused on identifying humor in general (Clemêncio et al., 2019; Inácio et al., 2023) or specifically detecting puns (Inácio and Oliveira, 2024), where texts are classified as humorous or not.

This study intends to contribute to the specific area of pun related tasks in the Portuguese language, with three main goals: (i) Fine-tune various language models for the task of pun location; (ii) Compare the performance of a pun classifier based on pun location against text classifiers, to assess whether models that locate pun words offer better performance and explainability in pun detection tasks; (iii) Explore how the identification of pun triggers can reveal cultural differences in humor between Brazilian Portuguese and European Portuguese contexts.

This work introduces the task of pun location to Portuguese, which, to the best of our knowledge, has not been explored previously. Pun location involves identifying pun words within texts, thereby not only detecting humor but also contributing to the explainability of the results. This is a significant step beyond mere detection, as it provides insights into the specific elements within the text that trigger the humorous effect.

For this purpose, a sequence-labeling approach is employed on Puntuguese (Inácio et al., 2024), a recently compiled dataset of punning texts in Portuguese, where pun words are manually annotated. Various language models are fine-tuned for this task, and post-processing is applied to further enhance the models' performances. Moreover, we leverage on the resulting models for the task of pun detection, where performance was close to previous approaches exclusive for this task, with the advantage of identifying which words contribute to the pun. Since the dataset includes puns in both European and Brazilian Portuguese, the capacity of the language models to gen-

eralize across these variants is also explored, providing insights into how cultural and linguistic differences are handled in the context of humor.

The main contributions of this work are: (i) an assessment of the performance of Portuguese language models in humor-related tasks; (ii) explorations on the identification of pun words in Portuguese texts, towards further advances in computational humor for this language; (iii) a methodological framework that both detects puns and locates the pun words, allowing for outcome explainability; (iv) and an exploration on how language models manage cultural and linguistic variations in humor.

Furthermore, this work resulted in the publication of the paper "Sequence Labeling for Pun Location and Detection in Portuguese" Gameiro et al. (2024) in the proceedings of the EPIA 2024 International Conference on Artificial Intelligence.

The remainder of this document is structured as follows: Chapter 2 provides the necessary background knowledge about humor fundamentals as well as concepts of NLP. Chapter 3 reviews related works in computational humor and pun related tasks, particularly focusing on pun detection, location and interpretation. Chapter 4 describes the methodology, including the dataset, models, and experimental setup. Chapter 5 reports the results of the models in both pun location and detection tasks, as well as the exploratory analysis of language variants. Chapter 6 closes with the main conclusions of this work and potential directions for future research.

2 Background Knowledge

This chapter provides the theoretical and technical background necessary for understanding the tasks of pun location and detection within the context of NLP. It covers the various theories and classifications of humor and puns, as well as the essential concepts and techniques in NLP. This includes the phases of text processing, sequence labeling, and the evolution of NLP approaches from traditional methods to state-of-the-art transformer models.

2.1 Fundamentals of Humor

Humor is a key aspect of human behavior and cognition, raising interest across multiple academic fields, including philosophy, psychology, sociology, anthropology, and linguistics. In general, research has focused into several aspects, including the mechanisms behind humor, its psychological impact and its function in social interactions (Shahaf et al., 2015).

Given that humor is a subjective human experience, its roots are explained through different theories. Raskin (1979) and Attardo (2008) categorize humor theories into three groups: Hostility, Release and Incongruity. Hostility theories suggest that humor derives from the feeling of superiority over others or a situation, or the act of targeting someone or something with aggression. Therefore, the root of the joke is putting someone down, mocking, or displaying dominance. Release theories argue that humor is a mechanism for releasing accumulated psychic energy or emotional tension. According to this view, humor allows individuals to express thoughts and feelings typically repressed by societal norms and personal inhibitions, or even to cope with tragic situations. The Incongruity theories propose that humor occurs when there is a cognitive dissonance between what is perceived and what is expected. Tagnin (2005) extends this by illustrating how humor emerges from breaking the conventional use of language, such as idiomatic expressions and formulaic speech.

Humor can be classified into positive and negative styles. Positive humor includes affiliative and self-enhancing humor, which foster social connection and personal well-being, respectively. For example, affiliative humor might involve a joke that brings people together, such as "Why don't scientists trust atoms? Because they make up everything." Self-enhancing humor involves maintaining a humorous outlook on life, like saying, "I'm on a seafood diet. I see food and I eat it."

Negative humor comprises aggressive and self-defeating styles, which may involve sarcasm or self-mockery, often at one's own expense. Aggressive humor might include a sarcastic comment, such as "I'm not arguing, I'm just explaining why I'm right." Self-defeating humor can involve making oneself the target of a joke, such as saying, "I put the 'pro' in procrastination." (Martin et al., 2003).

These styles of humor seem to appeal to different demographics. According to Tsai et al. (2021), adolescents and college-aged individuals seem to prefer affiliative humor, while adults generally favor both affiliative and self-enhancing humor. Older adults predominantly prefer self-enhancing humor. Furthermore, the study notes a gendered difference in humor preferences: men tend to appreciate aggressive humor more, while women are more inclined towards affiliative humor. This subjectivity in humor preferences among different demographics presents an additional challenge in NLP humor recognition, requiring models to be finely tuned to recognize and interpret a wide range of humorous styles accurately.

The inherent complexity of this task is further heightened when dealing with puns. A pun is a form of wordplay that exploits multiple meanings of a term or of similar-sounding words for an intended humorous or rhetorical effect (Attardo, 2009). This duality in meaning makes puns particularly challenging for NLP models to detect and interpret accurately, as it requires an understanding of context, phonetics, and semantics.

Puns can be classified as homographic, taking advantage of words with different meanings but the same orthography (homonymy); or heterographic, when their orthography is different. Concerning pronunciation, puns can also be classified as homophonic, when they sound the same. If their pronunciation is similar (but not the same), we mention them as heterophonic. For instance, in the pun *"I used to be a banker, but I lost interest."*, *"interest"* refers both to a feeling of concern or curiosity and to a charge for borrowed money. This pun is both homographic and homophonic, since both meanings of the word are written and pronounced the same away. In *"Need an ark to save two of every animal? I Noah guy."*, the pun word is *"Noah"*, which sounds like *"know a,"*. Therefore, the pun is also homophonic regarding pronunciation, but heterographic concerning orthography, since these words are written differently.

2.2 Concepts and Techniques in Natural Language Processing

NLP is a sub-field of Artificial Intelligence (AI) that focuses on enabling systems to understand, interpret, and generate human language automatically, in a manner that is significant and appropriate to the context. Such a task involves the development of

algorithms and models that can analyze and comprehend the structure and meaning of text or speech (Chopra et al., 2013).

The traditional NLP pipeline is composed of five main levels of linguistic analysis: Morphological and Lexical Analysis (examining the structure of words and their meanings), Syntactic Analysis (analyzing how words combine to form sentences), Semantic Analysis (interpreting the meanings of sentences), Discourse Integration (understanding how sentences connect and relate to each other in a larger context), and Pragmatic Analysis (determining the intended meaning or context behind the language used) (Chopra et al., 2013).

In order to prepare the text to be handled by NLP systems, the inputs are commonly preprocessed through text normalization. This step can involve tokenization, which breaks down a text into its constituent elements, known as tokens. These tokens are often words, but they can also include punctuation and other symbols. Therefore, the input data is simplified into manageable units for further processing (Jurafsky et al., 2024). Lemmatization is another process that further simplifies the text, by reducing words to their base or dictionary form. It involves analyzing the morphology of words to accurately transform them into their simplest form, known as lemmas. For instance, "running", "ran", and "runs" are all forms of the word "run". This is particularly relevant for morphologically complex languages (Jurafsky et al., 2024).

After preprocessing and linguistic analysis, various methods are applied to categorize and extract information from the text. Among these methods, sequence labeling is a commonly employed technique. This approach involves assigning a category to each element within a sequence of text. Two specific tasks under this approach are Part-of-Speech (POS) Tagging and Named Entity Recognition (NER).

POS Tagging assigns grammatical categories to each word in a sentence, such as nouns, verbs, adjectives, etc., based on the word's role and context. This helps in syntactic parsing and in understanding sentence structures (Jurafsky et al., 2024). An example of POS, done with library spaCy (Honnibal and Montani, 2017), can be seen on Figure 2.1. The sentence is first tokenized into discrete units: "The," "cat," "sat," "on," "the," "warm," and "windowsill. Each token is then assigned a POS tag: The words "The" are labeled as Determiners (DET), "Cat" and "windowsill" are identified as Nouns (NOUN), "Sat" is tagged as a Verb (VERB), "On" is a Preposition (ADP) and "Warm" is classified as an Adjective (ADJ). In the same figure, it is also possible to observe dependency relations between words that are syntactically connected.

NER identifies and categorizes key information in text into predefined groups such as names of people, organizations, locations, and times (Jurafsky et al., 2024). The example on Figure 2.2 shows the NER results obtained for the sentence "On March 2015 Kendrick Lamar released the album *To Pimp a Butterfly*" with spaCy library. "Kendrick Lamar" is recognized as a Person, identifying the individual mentioned. "March 2015"

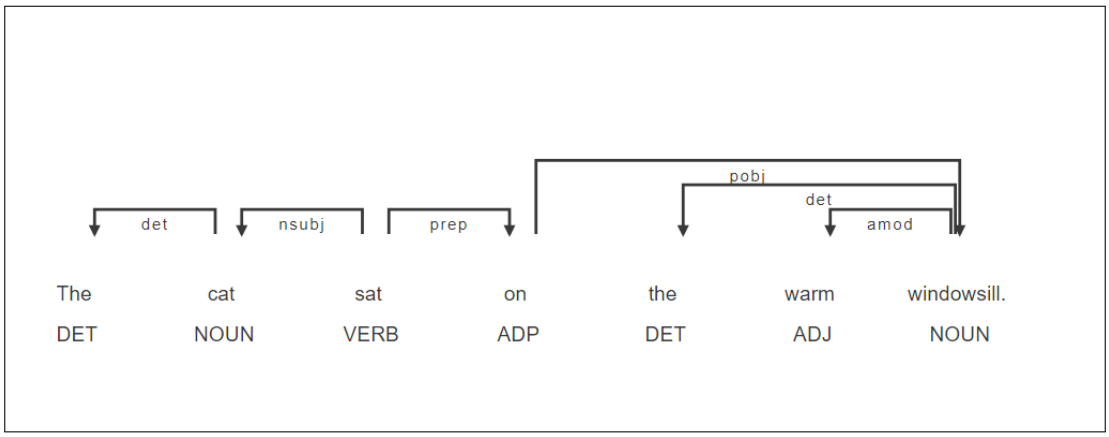

Figure 2.1: POS Tagging for sentence "The cat sat on the warm windowsill."

is classified as a Date, referring to a specific point in time. Additionally, while "To Pimp a Butterfly" is not a named entity in the traditional sense, it is categorized as a Work of Art (e.g., an album) depending on the context of the analysis.

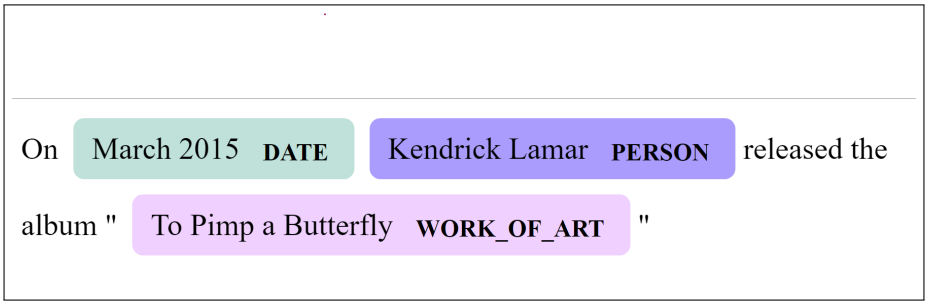

Figure 2.2: NER for sentence "On March 2015 Kendrick Lamar released the album 'To Pimp a Butterfly'"

One of the main challenges in NLP is handling ambiguity, such as in cases of polysemy, where words like "bank" can refer to a financial institution or the land beside a river depending on the context. This can be handled through tasks like Word Sense Disambiguation (WSD). WSD is the process of figuring out which meaning of a word is being used in a specific context when the word has several possible meanings, such as the example provided in Figure 2.3 with the word "bank". Oftentimes, the correct meaning is determined by linking the textual information to a lexicon, such as WordNet (Princeton University, 2010), an extensive online thesaurus that organizes words into sets called synsets. Each synset represents a distinct concept, and all the words within a synset are interchangeable terms that can refer to the same underlying concept. For example, in the synset for "dog," terms like "dog," "domestic dog," and "Canis familiaris" all refer to the same concept. Additionally, WordNet maps relationships between words, such as the "IS-A" relationship, showing how a dog is a type of mammal, and part-whole relationships, like an engine being part of a car (Jurafsky et al., 2024).

Regarding approaches to implement NLP systems, the field has evolved from rule-

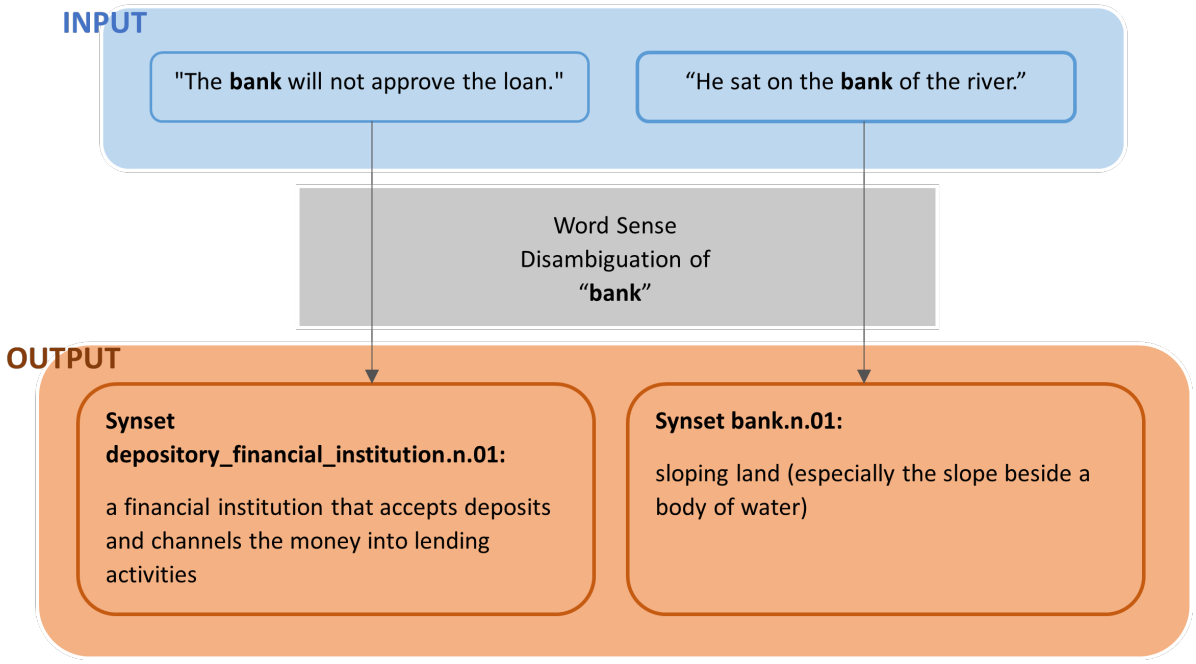

Figure 2.3: WSD of word "bank" in two contextually different sentences using WordNet

based and statistical methods to sophisticated neural network approaches. Initially, NLP applications leveraged simpler neural network architectures like feed-forward networks and, later, recurrent neural networks (RNNs), including Long Short-Term Memory (LSTM) units and Gated Recurrent Units (GRUs). While these RNNs have provided improvements at processing sequences, they still struggled with long-range dependencies and computational efficiency.

Before the introduction of complex neural architectures, distributional models like word2vec Mikolov et al. (2013) and GloVe Pennington et al. (2014) played a crucial role in advancing NLP. These models transformed words into dense vectors, based on their contextual usage in large corpora, enabling the capture of semantic relationships between words. Word2vec employs a shallow, two-layer neural network to predict word occurrences in a given context, whereas GloVe leverages global word co-occurrence statistics from a corpus to generate word embeddings. Both methods significantly improved the performance of various NLP tasks by providing rich, pre-trained word representations.

The introduction of transformer models by Vaswani et al. (2017) marked a significant shift in NLP by replacing recurrent layers with self-attention mechanisms, allowing the model to weigh the importance of different words in a sentence, regardless of their positional distance from each other. This architecture has significantly improved the handling of context and long-range dependencies in text, leading to state-of-the-art performance in many NLP tasks.

Following the transformer models, the concept of transfer learning has become a staple

in NLP. This approach involves pre-training a model on a large corpus of text and then fine-tuning it on smaller task-specific datasets. One of the landmark models utilizing this approach is BERT (Bidirectional Encoder Representations from Transformers), developed by Devlin et al. (2018). BERT and its variants utilize a bidirectional training of transformers, which is a fundamental difference from previous models that processed text in a single direction, either from left to right or right to left.

Other notable transformer-based models RoBERTa, which optimizes BERT's methodology by adjusting key hyperparameters and removing the next-sentence pre-training objective (Liu et al., 2019); DistilBERT, which offers a distilled version of BERT that maintains most of the original model's effectiveness while being smaller and lighter (Sanh et al., 2020); and GPT (Generative Pre-trained Transformer) by OpenAI, which excels in generating coherent and contextually relevant text based on prompts (Radford et al., 2018). Additionally, GPTNeo Black et al. (2021) offers an open-source implementation of model and data-parallel architectures similar to GPT-2 and GPT-3.

Beyond their capabilities in text generation and understanding, some of the larger transformer models, such as GPT-3, have also demonstrated impressive capabilities in few-shot learning, where they can perform tasks effectively with minimal examples Brown et al. (2020).

These transformer models have significantly advanced the state of the art across a wide range of NLP tasks, including Text Classification, Machine Translation, and Question Answering. Their ability to handle complex language tasks with greater speed and accuracy has also enabled practical applications such as real-time language translation and automated content generation, making advanced NLP capabilities more accessible and effective.

On top of their capabilities, transformer models and the tools needed to leverage them are now widely accessible through various platforms, with Hugging Face being the most prominent. The Hugging Face Hub[1] hosts an extensive collection of pre-trained transformer models, while the Transformers library(Wolf et al., 2020) offers a seamless interface for implementing and fine-tuning these models. This combination of resources has made it significantly easier to incorporate transformer models into several projects, such as this one, thereby accelerating research and application in the field of NLP.

[1] `https://huggingface.co/` (Accessed 16 September 2024)

3 Related Work

This chapter addresses related work in the fields of humor recognition and pun related tasks, divided into two main sections for a more structured and specific analysis.

The first section, Humor Recognition, focuses on the methods and approaches used to identify and classify humor in various forms of text. This includes examining linguistic features, machine learning techniques, and computational models that contribute to the detection of humorous content.

The second section, Pun Related Tasks, targets the specific challenges and methods for detecting, locating, and interpreting puns. Compared with general humor recognition, puns are particularly tricky due to their reliance on wordplay and double meanings. This requires specialized techniques and approaches to enable language models to perform these tasks, making them a distinct and more intricate part of humor research.

3.1 Humor Recognition

Humor recognition in NLP has seen a variety of approaches within the last decade, each prioritizing different sets of features and methodologies. This task has traditionally been approached as a binary classification task, in which a sentence is classified between humorous or non-humorous, through traditional machine learning models such as Support Vector Machine (SVM) (Pérez, 2012; Castro et al., 2016; Clemêncio et al., 2019), Naïve-Bayes (NB) (Pérez, 2012; Castro et al., 2016; Clemêncio et al., 2019), Decision Trees (DT) (Castro et al., 2016; Clemêncio et al., 2019), k-Nearest Neighbors (KNN) (Castro et al., 2016) and Random Forest (Yang et al., 2015). In general, SVM seems to perform better in this task (Pérez, 2012; Castro et al., 2016; Clemêncio et al., 2019), with NB (Pérez, 2012) and KNN (Castro et al., 2016) also presenting satisfactory results.

These methods require careful feature extraction, namely linguistic and semantic features. The most commonly used features in the researched studies are Ambiguity (Pérez, 2012; Yang et al., 2015; Castro et al., 2016; Liu et al., 2018; Clemêncio et al., 2019; Inácio et al., 2023) and Incongruity (Pérez, 2012; Yang et al., 2015; Liu et al., 2018; Inácio et al., 2023). In fact, Clemêncio et al. (2019) found that Incongruity and Ambiguity were the most significant features for the task of Humor Recognition. Ambiguity is the linguistic phenomenon that occurs when a word or sentence can have multiple interpretations. Incongruity refers to the creation of apparent contradiction by mixing

two incongruous frames in one statement. Here, frames are different contexts or perspectives that provide background and meaning to a statement. When these frames clash or are unexpectedly combined, it creates humor through incongruity. Other commonly extracted linguistic features include Emotion-based features (Pérez, 2012; Ortega-Bueno et al., 2018), Subjectivity (Yang et al., 2015; Liu et al., 2018) and Negation (Castro et al., 2016; Ortega-Bueno et al., 2018). Table 3.1 provides a summary of the mentioned studies and the linguistic and semantic features they used.

Table 3.1: Overview of linguistic and semantic features used in Humor Recognition studies

Feature	**References**
Ambiguity	Pérez (2012) Yang et al. (2015) Castro et al. (2016) Liu et al. (2018) Clemêncio et al. (2019) Inácio et al. (2023)
Incongruity	Pérez (2012) Yang et al. (2015) Liu et al. (2018) Inácio et al. (2023)
Emotion-based features	Pérez (2012) Ortega-Bueno et al. (2018)
Subjectivity	Yang et al. (2015) Liu et al. (2018)
Negation	Castro et al. (2016) Ortega-Bueno et al. (2018)

Several humor recognition studies also utilize lexical resources, such as WordNet, which provide comprehensive repositories of words, phrases, idioms, and their meanings and connotations. These allow to extract relevant features for humor recognition tasks, such as Alliteration (Yang et al., 2015; Liu et al., 2018; Ortega-Bueno et al., 2018), Antonymy (Ortega-Bueno et al., 2018; Clemêncio et al., 2019; Inácio et al., 2023), Polarity (Pérez, 2012; Yang et al., 2015; Liu et al., 2018), Sentiment-based features (Yang et al., 2015; Liu et al., 2018; Ortega-Bueno et al., 2018; Clemêncio et al., 2019; Inácio et al., 2023), Adult Slang (Castro et al., 2016; Ortega-Bueno et al., 2018; Clemêncio et al., 2019; Inácio et al., 2023), Rhyme (Yang et al., 2015; Liu et al., 2018), Stylistic features (Castro et al., 2016; Ortega-Bueno et al., 2018; Clemêncio et al., 2019; Inácio et al., 2023) and Human-Centric features (Ortega-Bueno et al., 2018). Table 3.2 shows an overview of the lexical resources employed in these studies.

In recent years, humor recognition has seen a paradigm shift with the introduction of deep learning techniques, particularly neural networks and attention mechanisms. Ortega-Bueno et al. (2018) propose a model that combines linguistic features with an

Table 3.2: Overview of lexical features extracted with lexical resources in Humor Recognition studies

Feature	**References**
Alliteration	Yang et al. (2015) Liu et al. (2018) Ortega-Bueno et al. (2018)
Antonymy	Ortega-Bueno et al. (2018) Clemêncio et al. (2019) Inácio et al. (2023)
Polarity	Pérez (2012) Yang et al. (2015) Liu et al. (2018)
Sentiment-based features	Yang et al. (2015) Liu et al. (2018) Ortega-Bueno et al. (2018) Clemêncio et al. (2019) Inácio et al. (2023)
Adult Slang	Castro et al. (2016) Ortega-Bueno et al. (2018) Clemêncio et al. (2019) Inácio et al. (2023)
Rhyme	Yang et al. (2015) Liu et al. (2018)
Stylistic features	Castro et al. (2016) Ortega-Bueno et al. (2018) Clemêncio et al. (2019) Inácio et al. (2023)
Human-Centric features	Ortega-Bueno et al. (2018)

Attention-Based Recurrent Neural Network to detect humor on Spanish tweets. The architecture features Bidirectional Long Short-Term Memory (Bi-LSTM) and an attention mechanism. This attention mechanism, which assesses the significance of each word, generates a context vector. Subsequently, another LSTM model utilizes this context vector to estimate if the tweet is considered humorous or not.

Transformer architectures have recently gained prominence in humor recognition tasks. With the release of BERT, several studies in the latest years started to use this model (Weller and Seppi, 2019; Ismailov, 2019; Grover and Goel, 2021; Garcıa-Dıaz and Valencia-Garcıa, 2021; Wang et al., 2021; Faraj and Abdullah, 2021; Peyrard et al., 2021), as well as its variations, namely RoBERTa (Grover and Goel, 2021; Faraj and Abdullah, 2021; Peyrard et al., 2021; Pan et al., 2021; Xiong et al., 2022), ERNIE 2.0 (Pang et al., 2021) and AlBERT (Pan et al., 2021). Other language-specific variations of BERT have also been explored, namely BERTO (Grover and Goel, 2021) for Spanish, and BERTimbau (Inácio et al., 2023) and Albertina (Inácio and Oliveira, 2024) for Portuguese.

3.2 Pun Related Tasks

Several studies have focused on the computational task of pun disambiguation, using a variety of different approaches and features to either detect, locate or interpret puns. Pun detection refers to the task of classifying a sentence or a text based on whether it contains a pun or not (Miller et al., 2017). Given its binary nature, this has been approached through application of supervised classification algorithms, namely RNN, Multilayer Perceptron (MLP), SVM, DT, RF, NB and K-NN. A study comparing all these algorithms in pun detection, showed that RNN and MLP provided better performances, with RF also demonstrating good results (Jaiswal and Monika, 2019).

The most common approaches to pun detection include deep learning methodologies (Diao et al., 2018, 2019; Ren et al., 2021). Diao et al. explored different approaches for homographic (2018) and heterographic puns (2019). For homographic puns, the authors applied an improved word embedding with WordNet, a Bi-LSTM model to extract latent semantic information and a neural attention mechanism to capture the words' collocation. For heterographic puns, the developed model consists of a hierarchical attention convolutional neural network and a multi-level embedding attention network, combined by a gated attention mechanism. Another study proposed an attention-based multi-task learning algorithm to recognize humor and detect if a sentence is a pun at the same time (Ren et al., 2021).

Pun location implies a more complex task of identifying the specific word or words in the sentence that are being used as a pun (Miller et al., 2017). This task was explored mostly through deep learning approaches (Cai et al., 2018; Mao et al., 2020; Liu et al., 2021). One of these studies considered both long-distance and short-distance semantic relations between words in order to locate puns, through a Compositional Semantics Network with Multi-Task learning (Mao et al., 2020). Another approach employed a sense-aware neural model, built on WSD, in order to handle different senses within the same sentence, to locate homographic puns. A more recent approach employed also sense-aware modules, with extraction of semantic and pronunciation information simultaneously. This allowed for pun location in both homographic and heterographic puns (Liu et al., 2021).

Some researchers explored both pun detection and location in the same approach. One of these studies employed Bi-LSTM networks to capture contextual information, on top of Conditional Random Fields (CRF), to both homographic and heterographic puns (Zou and Lu, 2019). Another study employed self-attentive embedding with contextualized and phonological features for pun detection and location in both types of puns (Zhou et al., 2020). In a different approach, logistic regression and feature engineering were applied for these joint tasks, using statistic and semantic properties of puns (Feng et al., 2020).

Pun interpretation involves understanding the multiple meanings of the punning word identified in the pun location task. A study approached pun interpretation as a classification task, through construction of Pun-Gloss Pairs (Liu et al., 2021).

At the 11th edition of the International Workshop on Semantic Evaluation (SemEval) (Miller et al., 2017), the tasks of pun detection, location and interpretation were proposed to the natural language processing research community. The participants developed different systems to perform these tasks in both homographic and heterographic puns. The approaches included Lesk-Like algorithms (Oele and Evang, 2017), Word Sense Disambiguation (Pedersen, 2017), Rule-based approaches (Özge Sevgili et al., 2017; Vechtomova, 2017), probabilistic models (Doogan et al., 2017) and supervised learning (Mikhalkova and Karyakin, 2017; Das and Pramanick, 2017).

The JOKER track in CLEF (Ermakova et al., 2023) was a more recent evaluation that encompassed the three previously mentioned subtasks and also pun translation in three languages: English, Spanish, French. Most participants relied on transformer architectures, namely GPT-3, BLOOMZ, and Simple T5 for the three related tasks. In general, the approaches using Simple T5 achieved the best performances for pun detection (Galeano, 2022). For pun location, performance varied across languages, with Simple T5 also achieving the best performance for English (Ohnesorge et al., 2023) and Spanish (Popova and Dadić, 2023). One of the presented approaches relied on XLM-RoBERTa for locating puns via sequence labelling (Dsilva, 2023), showing promising results in English and was the best performing in French; however, for both French and Spanish, accuracy remained below 0.6. This method resembles the one proposed in our current study.

To enhance the performance of the applied methodologies, researchers have leveraged a range of linguistic features to detect, locate, and interpret puns.

A study with a probabilistic model demonstrated that ambiguity of meaning and distinctiveness of viewpoints in a sentence seem to be helpful in distinguishing puns from non-puns (Kao et al., 2016). Ambiguity in puns is usually resolved through WSD techniques and tools, such as Lesk-like methods, WordNet, SenseGram, Word2Vec and sense embedding cosine distances (Oele and Evang, 2017; Özge Sevgili et al., 2017; Pedersen, 2017; Cai et al., 2018; Diao et al., 2018; Liu et al., 2021).

POS tagging is commonly applied across NLP studies, hence it has also been used in pun related tasks (Miller and Gurevych, 2015; Das and Pramanick, 2017; Mikhalkova and Karyakin, 2017; Vechtomova, 2017; Zou and Lu, 2019; Diao et al., 2020; Feng et al., 2020; Ren et al., 2021; Özge Sevgili et al., 2017). One of these studies explored calculating the probability of different POS being a pun, with nouns having the highest probability. This study also considered contextual factors, such as the POS of adjacent words to modify the probability of a word being a pun (Das and Pramanick, 2017). Another approach considered specific POS related features, such as the number of words

with certain POS tags (Feng et al., 2020).

Semantic analysis features are used to understand the semantic aspects of words and sentences in both homographic and heterographic puns. Semantic transparency allows to analyze the clarity of semantic meanings of words within a sentence, while semantic relevance is used to measure the semantic connection between the selected word and the source sentence (Diao et al., 2020). Another study also explored mining semantic fields (groups of words sharing a common semantic property) using Roget's Thesaurus (Mikhalkova and Karyakin, 2017).

Word embeddings of different types are frequently used to capture specific aspects of language in pun-filled sentences. Contextualized word embeddings, such as those obtained from BERT, are useful for deriving word representations in context (Zhou et al., 2020; Ren et al., 2021). Pronunciation embeddings focus on phonological characteristics of words by breaking them into phonemes (Zhou et al., 2020). Character-level embeddings are able to capture structural and lexical features relevant for both types of puns (Zou and Lu, 2019; Diao et al., 2019). Moreover, traditional word embeddings, such as Word2Vec and GloVe, allow words to be represented compactly, making it easier to measure their semantic connections. This approach is versatile and applied to the detection and interpretation of both types of puns (Özge Sevgili et al., 2017; Feng et al., 2020). Multi-Level Embeddings are also applied, which include character, word, and interacting embeddings to capture diverse aspects of data (Mao et al., 2020).

The frequency and semantic relationships in words and phrases are also analyzed, particularly through Google n-grams. By examining the co-occurrence patterns of words, researchers can gain a better understanding of the contextual usage of words with multiple meanings (Doogan et al., 2017).

Several studies have also considered the word position in a sentence (Huang et al., 2017; Vechtomova, 2017; Zou and Lu, 2019; Diao et al., 2019). In particular, some of these approaches extracted this feature based on the assumption that most of the puns seemed to be located at the end of the sentence (Huang et al., 2017; Zou and Lu, 2019).

Pointwise Mutual Information (PMI) is frequently used to assess semantic associations between words in the pun text. This metric computes word association scores to identify distinctive words highly associated with the pun (Vechtomova, 2017; Özge Sevgili et al., 2017; Feng et al., 2020). Inverse Document Frequency (IDF) is also used to measure the rarity of each word in the corpus (Vechtomova, 2017).

Since heterographic puns rely on the interplay between different spellings and similar pronunciations, phonetic features are frequently used in studies involving these specific types of wordplay. These include phonetic edit models, phonetic lexicons, and phonetic similarity measures (Jaech et al., 2016; Doogan et al., 2017; Zhou et al., 2020).

In summary, the fields of pun detection, location, and interpretation have been subject

to a wide range of computational approaches, which highlight the complexity of these tasks. However, most of these approaches were done for the English language, and to the best of our knowledge there are currently none for the Portuguese language.

4 Methodology and Data

This chapter describes the methodology adopted in this work, including the description of the dataset used, the models selected for the task, and the experimental setup, which details the approach taken for pun location.

4.1 Dataset

All the tasks were performed using the Puntuguese (Inácio et al., 2024) dataset, as available in the Hugging Face platform[1]. Although previous work by Oliveira et al. (2020) provided a general corpus of humorous texts in Portuguese, Inácio et al. (2023) identified potential data leakage in humor recognition tasks. Besides showing better performance results in machine learning tasks, Puntuguese is specifically focused on puns, making it more suitable for this task.

Puntuguese is a curated collection of punning texts in both Brazilian and European Portuguese with a total of 2,850 puns, 2,053 attributed to Brazilian Portuguese and 797 to European Portuguese. The diversity in language variants presents an advantage, since the models will be exposed to a wider range of unique vocabulary, idiomatic expressions, grammatical structures and cultural contexts. This is evident in the examples shown in Figure 4.1. The European Portuguese pun requires knowledge of both regional accents and cultural context, as in certain areas of Portugal, "tenho Tide" sounds like "tenho tido" (meaning "I have had"), and the pun involves Jorge Jesus, a well-known Portuguese football coach. Similarly, the Brazilian Portuguese pun also relies on cultural background and linguistic nuances. MC Kevinho is a Brazilian funk musician, and the pun plays on his name: "Kevinho" sounds like "Quer vinho" (meaning "Want wine"), and "Kessuco" sounds like "Quer suco" (meaning "Want juice"), when pronounced with a Brazilian accent. Additionally, the word "suco" is not commonly used in European Portuguese in this context.

European Portuguese	"Hoje vi o Jorge Jesus num anúncio de detergentes em que ele dizia: Este é o melhor detergente que tenho **Tide**!"
Brazilian Portuguese	"Qual o nome do filho do Mc Kevinho? MC **Kessuco**."

Figure 4.1: Examples of European Portuguese and Brazilian Portuguese puns in Puntuguese dataset.

[1] `https://huggingface.co/datasets/Superar/Puntuguese` (Accessed 16 September 2024)

The dataset further categorizes puns into four distinct types, namely: homophonic (572 puns), homographic (6 puns), a combination of both homophonic and homographic (504 puns), and those that are neither (1,827 puns). Table 4.1 provides an example of pun for each one of these types. The homophonic pun relies on a play on words that sound the same but are written differently. In Brazilian Portuguese, the pronunciation of "regue" (water) sounds like "reggae". The homograph pun, on the other hand, involves words that are written the same but have different meanings and pronunciations. The words "de cor" can mean either "by heart" or "by color", and they are pronounced differently according to the meaning. The example of a pun both homophonic and homographic contains the word "console" which can either refer to a video game console or the act of consoling someone, and it is pronounced and written the same in both contexts. Lastly, an example that is neither homographic nor homophonic involves the words "fulano" (a generic term for a person) and "flan" (a type of dessert). These words are similar in both spelling and pronunciation, but not identical, making this an example of paronymy.

Table 4.1: Examples of different types of puns in Puntuguese dataset.

Pun type	**Pun example**
Homophonic	"Qual é o estilo musical favorito das plantas? **Regue**."
Homographic	"Porque é que os daltónicos têm péssima memória? Porque não sabem nada **de cor**."
Homophonic and homographic	"O que o videogame triste espera que a gente faça? Que o **console**."
Neither	"Qual é a sobremesa preferida dos canibais? Pudim **fulano**."

Each pun is manually annotated with its punning mechanisms, namely: the pun words (i.e., the triggers for the text to be considered a pun) and their respective alternative signs (i.e., the different ambiguous meanings of the pun words). In addition to the punning texts, Puntuguese includes a non-humorous counterpart for each entry, which was achieved through micro-editing. An example [2] [3] of this is displayed on Figure 4.2. This allows to train models for pun location and pun detection tasks.

Pun	"Porque é que os polícias não gostam de sabão? Porque preferem **deter gente**."
NH Counterpart	"Porque é que os polícias não gostam de sabão? Porque preferem sabonete."

Figure 4.2: Example of pun and respective non-humorous (NH) counterpart in Puntuguese dataset.

[2] In English, "Why don't police officers like soap? Because they prefer to arrest people ("deter gente" sounds like "detergent" in Portuguese)."

[3] In English, "Why don't police officers like soap? Because they prefer bar soap."

Each example in the dataset contains the corresponding sequence of labels for the words in text: 1 for pun words and 0 for non-pun words. Figure 4.3 shows two examples of punning texts from the dataset and their respective labeling. In the first example[4] "*estado*" is the trigger for the pun, hence it is the only word labelled as 1, with all the others as 0. The second example[5] contains two pun triggers, "*Hobbits*" and "*Hobbyte*", therefore, the sequence includes two words labeled with 1.

Em	que	estado	se	encontra	o	rio	Mississipi	?	No	estado	líquido	.
0	0	0	0	0	0	0	0	0	0	**1**	0	0

O	que	são	oito	Hobbits	?	Um	Hobbyte	.
0	0	0	0	1	0	0	1	0

Figure 4.3: Examples of punning texts in Puntuguese and their labeling.

The Puntuguese dataset is split into training (70%), test (20%), and validation (10%) subsets, using a stratified sampling approach to maintain an even distribution of pun types and language varieties. This ensured that each subset mirrored the overall composition of the corpus in terms of the proportion of homophonic, homographic, mixed, and non-humorous text (see Table 4.2), as well as the balance between Brazilian and European Portuguese entries (see Table 4.3).

Table 4.2: Pun type distribution in Puntuguese dataset across train, validation and test subsets.

Pun type	**Train**	**Val**	**Test**	**Total**
Only homographic	5	0	1	6
Only homophonic	401	57	114	572
Both homophonic and homographic	352	51	101	504
Not homophonic nor homographic	1,283	182	362	1,827

Table 4.3: Language variant distribution in Puntuguese across train, validation and test subsets.

Language variant	**Train**	**Val**	**Test**	**Total**
Brazilian Portuguese	1,437	206	410	2,053
European Portuguese	558	79	160	797
Total	1,995	285	570	2,850

4.2 Models

Multiple encoder models of the BERT (Devlin et al., 2018) family and one decoder model based on GPT-Neo (Black et al., 2021), were fine-tuned for the task of pun

[4]In English, "In what state is the Mississippi River? In the liquid state."
[5]In English, "What are eight Hobbits? A Hobbyte."

location. The models were selected from high-performing available open-source options for the Portuguese language, aiming to achieve diversity in both architecture and size. The following models, all available on the Hugging Face Hub, were considered: BERTimbau (Souza et al., 2020a), base[6] and large[7]; BERTugues[8]; RoBERTa PT-BR[9]; Albertina (Rodrigues et al., 2023), PTPT[10] and PTBR[11]; and GlórIA[12].

BERTimbau and BERTugues are both based on BERT. BERTimbau base and BERTugues are relatively lightweight, each with 110 million parameters across 12 layers, while BERTimbau large has 335 million parameters and 24 layers. BERTimbau has shown to perform effectively on Portuguese datasets, achieving competitive results in various benchmark evaluations (Souza et al., 2020b). Similarly, BERTugues has also shown reliable performance, even outperforming both BERTimbau base and large in certain tasks (Zago, 2023).

RoBERTa PT-BR, based on RoBERTa, offers a slight increase in complexity over the lighter models, with 125M parameters. While specific benchmarks for RoBERTa PT-BR are not available, the base model has shown improved performance compared to BERT (Liu et al., 2019).

Albertina is among the more resource-intensive models selected, with 887 million parameters and 24 layers. Based on DeBERTa, Albertina shows remarkable performances on benchmarks, scoring higher than BERTimbau large in some metrics.

Finally, GlórIA, has 1.3 billion parameters and 24 layers, making it the largest model among those considered. Based on GPT-Neo, it demonstrates state-of-the-art performance on Portuguese NLP generative tasks and shows competitive results in Portuguese benchmarks (Lopes et al., 2024).

All models are pretrained in Brazilian Portuguese text, with the exception of Albertina PTPT and GlórIA, which are pretrained for European Portuguese.

4.3 Experimental setup

The designed approach addresses pun location as a sequence-labeling task. The goal is to obtain a model capable of, given a short text with a pun, output its words correctly

[6] `https://huggingface.co/neuralmind/bert-base-portuguese-cased` (Accessed 16 September 2024)

[7] `https://huggingface.co/neuralmind/bert-large-portuguese-cased` (Accessed 16 September 2024)

[8] `https://huggingface.co/ricardoz/BERTugues-base-portuguese-cased` (Accessed 16 September 2024)

[9] `https://huggingface.co/josu/roberta-pt-br` (Accessed 16 September 2024)

[10] `https://huggingface.co/PORTULAN/albertina-900m-portuguese-ptpt-encoder` (Accessed 16 September 2024)

[11] `https://huggingface.co/PORTULAN/albertina-900m-portuguese-ptbr-encoder` (Accessed 16 September 2024)

[12] `https://huggingface.co/NOVA-vision-language/GlorIA-1.3B` (Accessed 16 September 2024)

labeled as pun (1) or non-pun (0).

The data was preprocessed to convert all text to lowercase, in order to reduce token variation, which seems to improve the models' performances in these sort of tasks (Chen et al., 2020; Kostić et al., 2023). Each example was then tokenized with the respective tokenizer of each model, ensuring that labels were properly aligned. All subwords and special tokens were assigned a label of -100, which were ignored during model training and evaluation. Table 4.4 shows an example of the tokenization process and label assignment.

Table 4.4: Example of tokenization with BERTimbau base and label alignment.

Tokens	O	que	são	oito	**Ho**	##bb	##its	?	Um	**Ho**	##bby	##te	.
Labels	0	0	0	0	**1**	-100	-100	0	0	**1**	-100	-100	0

After this process, the models were fine-tuned using the transformers library from Hugging Face (Wolf et al., 2020). The hyperparameters selected for training each model were empirically obtained and included a learning rate of 3e-5, batch sizes of 8 for both training and evaluation, a training duration of 4 epochs, and weight decay set to 0.02.

All models were evaluated with Precision, Recall and micro F1, considering the results for sequence labeling of pun words (i.e., words labeled as 1). These metrics were computed with the Hugging Face *evaluate* library[13] and the python framework for sequence labeling evaluation *seqeval* (Nakayama, 2018).

Fine-tuned models were assessed in two distinct scenarios: (i) considering exclusively the positive examples in Puntuguese, i.e., texts with at least a pun word, enabling the model to focus its attention on identifying exactly where the puns are, making it more effective for pun location tasks; (ii) considering both positive and negative examples, i.e., not only the punning texts, but also the texts without pun words. In this latter case, the model is expected to distinguish further whether a text contains a pun, making it also suited for pun detection.

[13] `https://huggingface.co/docs/evaluate/index` (Accessed 16 September 2024)

5 Results and Discussion

This chapter describes the results achieved for the tasks of pun location, followed by the exploration of post-processing, aiming at improving performance. Additionally, pun detection is addressed through pun location and its results are compared with those of models fine-tuned for pun detection directly. In closing, an exploratory analysis of the models' capacity to generalize across language variants is presented, specifically examining their performance on both European and Brazilian Portuguese puns. The fine-tuned models[1] and the source code used [2] are both publicly available at their respective repositories.

5.1 Pun location

Table 5.1 shows the models' performances in the pun location task, using only positive examples in the fine-tuning process. In this scenario, BERTimbau base achieved the best overall performance, with a Precision of 0.74, a Recall of 0.76, and an F1 of 0.75. BERTimbau large also achieved solid results, with an F1 of 0.74, while Albertina PTPT and BERTugues performed very close.

Despite being the most complex and resource-intensive model, GlórIA exhibited one of the worst performances in the pun location task. Unlike BERT models, which are encoder-based and are better suited for representing the meaning of each token in a sentence, GlórIA's decoder architecture is designed for tasks involving generation or sequential prediction. This architectural difference might have made it less effective at the token classification required for pun location.

Table 5.1: Performance of models fine-tuned with positive examples only

Model	Precision	Recall	F1
Albertina PTBR	0.70	0.64	0.67
Albertina PTPT	0.72	0.73	0.73
BERTimbau base	**0.74**	**0.76**	**0.75**
BERTimbau large	**0.74**	0.75	0.74
BERTugues	0.71	0.72	0.72
RoBERTa PTBR	0.64	0.64	0.64
GlórIA	0.65	0.64	0.64

[1] `https://huggingface.co/collections/LendeaViva/pun-location-in-portuguese-66b71f8fd9398df21fb45297` (Accessed 16 September 2024)

[2] `https://github.com/NLP-CISUC/Seq-labelling-puns` (Accessed 16 September 2024)

Although these initial results might not be too impressive, it should be noted that there are many more non-pun than pun words, which creates an unbalanced scenario that increases the challenge.

Moreover, analyzing the predictions closely shows that the models are not that far from correctly identifying the pun triggers. For example, in the text of Figure 5.1, annotated with the pun word "estado", the model correctly identifies the second instance of this word, while also classifying "Mississipi" and "líquido" as pun words. Despite not being ambiguous in the context of this text, these terms do contribute to explaining the pun and its humorous effect.

While there is no direct baseline for comparison, as no other work on pun location in Portuguese is currently available, these results can be contextualized within recent research on other Latin languages. For instance, a token classification approach, similar to the one proposed in this study, achieved an accuracy of 0.56 in Spanish and 0.41 in French for the task of pun location, as presented at JOKER CLEF 2023 (Dsilva, 2023). While a direct comparison is not possible, these results highlight the challenging nature of this task across Latin languages.

	Em	que	estado	se	encontra	o	rio	Mississipi	?	No	estado	líquido	.
Gold	0	0	0	0	0	0	0	0	0	0	**1**	0	0
Prediction	0	0	0	0	0	0	0	**1**	0	0	**1**	**1**	0

Figure 5.1: Example of pun location prediction with BERTimbau Large.

Table 5.2 displays the results in pun location, but for the scenario including both positive and negative examples in Puntuguese. All models consistently perform worse, which was expected due to the increased imbalance in word labels, now with even more non-pun than pun words. Additionally, in opposition to the first scenario, these models cannot assume that there is at least one pun word in each text. Despite the more variable performance across models, with an F1 score of 0.57, BERTimbau large was the best performing in this second scenario.

Table 5.2: Performance of models fine-tuned with the entire dataset.

Model	**Precision**	**Recall**	**F1**
Albertina PTBR	**0.59**	0.44	0.51
Albertina PTPT	0.57	0.44	0.50
BERTimbau base	0.54	0.52	0.53
BERTimbau large	**0.59**	**0.56**	**0.57**
BERTugues	0.55	0.51	0.53
RoBERTa PTBR	0.47	0.42	0.44
GlórIA	0.40	0.40	0.40

The best performing model for pun location in Portuguese (BERTimbau base with pos-

itive examples) has been deployed on a Hugging Face Space[3], with an interactive interface created using *gradio*(Abid et al., 2019). This application allows users to input text and assess the model's performance in locating pun words. Upon entering a Portuguese pun, the application highlights the pun words in a different color.

5.2 Labeling post-processing

Several studies (Huang et al., 2017; Zou and Lu, 2019) take advantage of the fact that the pun word tends to be at the end of the humorous text (Attardo, 2009). With this in mind, we try to improve performance by further post-processing the prediction of the fine-tuned models.

The first method, Last Word, (hereafter, LW) analyzes each text backwards, in order to consider the last positively-labeled word as the only pun, while disregarding any other words possibly labeled as such. This primarily aims to minimize false positives, which occur when non-pun words are mistakenly identified as puns. By limiting the recognition to the first encountered pun word, the method effectively reduces the likelihood of incorrectly labeling regular words as puns, enhancing the precision of the location prediction. For instance, in the pun of Figure 5.2[4] there is only one trigger word (*"calcular"*), thus, cleaning all false positives before this word makes the automatic labeling more precise.

	Qual	é	a	pacifista	que	é	um	gênio	da	matemática	?	Madre	Teresa	de	Calcular	.
Gold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0
Pred	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	**1**	0
LW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0

Figure 5.2: Example of pun location prediction with BERTimbau large, and improvement with the LW post-processing method.

However, this approach can inadvertently increase the rate of false negatives, i.e., actual puns not identified. Since only the first detected pun word is considered, any preceding puns in the text that could be valid are missed. This could be particularly limiting in complex examples where multiple puns are present.

The second method, Last Sequence, (hereafter, LS) also processes the text from the end, but differs by considering a continuous sequence of identified pun words as valid, up to the point where no further puns are detected, and removing pun annotations from the rest of the text. This strategy is designed to be less punitive than the first, thereby reducing false negatives by recognizing each pun appearing before a non-pun segment.

[3] `https://huggingface.co/spaces/LendeaViva/Pun_Location_Portuguese` (Accessed 16 September 2024)

[4] In English, "What do you call a pacifist who's a math genius? Mother Teresa of Calculus." "Calculus" and "Calcutta" are pronounced similarly in Portuguese

This is illustrated in Figure 5.3, with a pun[5] where the LW method would remove more positive labels than necessary, but the LS method works.

	Qual	animal	se	dissolve	na	agua	?	Orango	Tang	.
Gold	0	0	0	0	0	0	0	**1**	**1**	0
Pred	0	0	0	0	0	0	0	**1**	**1**	0
LW	0	0	0	0	0	0	0	0	**1**	0
LS	0	0	0	0	0	0	0	**1**	**1**	0

Figure 5.3: Example of pun location prediction with BERTimbau large, followed by results after the post-processing methods LW and LS.

Table 5.3 shows the performance after applying each of the post-processing methods. LW generally increases precision but significantly reduces recall by missing valid pun words that occur before the first identified. Conversely, LS provides a more balanced approach, slightly reducing recall, but maintaining or improving F1 and precision.

With post-processing, the best-performing model was BERTimbau large with LS, achieving a Precision of 0.77, a Recall of 0.73 and a F1 of 0.75. When compared to the performance without post-processing (Table 5.1), the F1 of this model improved by 1 percentage point ($0.75\ vs\ 0.74$), especially due to the increase in precision. However, for other models, the slight decrease in recall makes the overall impact hardly noticeable. This also suggests that, in many cases, the model is already labeling a single sequence as the pun.

Table 5.3: Results of post-processing approaches applied to fine-tuned models' predictions for pun location.

Model	**Initial**			**LW**			**LS**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Albertina PTBR	0.70	0.64	0.67	0.61	0.55	0.58	0.71	0.64	0.67
Albertina PTPT	0.72	0.73	0.73	0.63	0.61	0.62	0.74	0.71	0.73
BERTimbau base	0.74	**0.76**	**0.75**	0.63	0.60	0.62	0.76	0.73	**0.75**
BERTimbau large	0.74	0.75	0.74	0.63	0.61	0.62	**0.77**	0.73	**0.75**
BERTugues	0.71	0.72	0.72	0.60	0.57	0.58	0.73	0.69	0.71
RoBERTa PTBR	0.64	0.64	0.64	0.59	0.56	0.57	0.67	0.63	0.65
GlórIA	0.65	0.64	0.64	0.59	0.52	0.55	0.70	0.61	0.65

5.3 Pun detection through pun location

We further explored the previous models for the task of pun detection. This was based on the simple assumption that a text is classified as a pun if it has at least one word labeled as pun (1). Otherwise, a text with non-pun words only (0) is classified as

[5]In English, "Which animal dissolves in water? Orangu Tang."

non-pun. For this, we use the models trained in the second scenario, which, despite the lower performance in pun location (see Table 5.2), were the ones exposed to both punning and non-punning texts.

Table 5.4 reports on the performances achieved in this task by each trained model. For reference, we also include performances by models fine-tuned directly for pun detection (Inácio and Oliveira, 2024) as a whole sequence classification task, here replicated in the most recent version of the dataset.

As expected, models trained to locate pun words, a different task, generally perform below those trained for pun detection. For instance, when tasked with locating pun words first, the performance of the Albertina models decreases significantly when compared to the same models trained directly for pun detection (F1 of 0.51 vs 0.69 and 0.71). For BERTimbau, this behavior is not so clear. When fine-tuned for pun location, the base model is better (F1 of 0.65 vs 0.51), whereas the large model has a lower precision (0.71 vs 0.73), but a higher recall (0.63 vs 0.59).

Table 5.4: Pun detection through sequence labeling versus directly (Inácio and Oliveira, 2024)

	Model	**Precision**	**Recall**	**F1**
Seq. Labeling	Albertina PTBR	0.76	0.39	0.51
	Albertina PTPT	0.74	0.39	0.51
	BERTimbau large	0.71	0.63	0.67
	BERTimbau base	0.67	0.63	0.65
	BERTugues	0.67	0.57	0.61
	RoBERTa PTBR	0.61	0.47	0.53
	GlórIA	0.53	0.55	0.54
Inacio and Oliveira, 2024	Albertina PTBR	**0.77**	0.64	0.69
	Albertina PTPT	0.72	**0.69**	**0.71**
	BERTimbau large	0.73	0.59	0.65
	BERTimbau base	0.65	0.42	0.51
	Bertugues	0.69	0.54	0.61
	RoBERTa PTBR	0.57	0.59	0.58
	GlórIA	0.57	0.61	0.59

While models trained for pun word location do not always outperform those by Inácio and Oliveira (2024), they still offer an advantage. Specifically, by identifying pun words, these models provide an explanation of why a text is classified as a pun. This is particularly beneficial for applications requiring not just classification, but also a support for the decision, enhancing the output explainability.

5.4 Generalization across Portuguese Variants

Since the Puntuguese dataset contains puns from two variants of the Portuguese language, European Portuguese and Brazilian Portuguese, this can be leveraged to assess the models' performance across these variants, and provide a pathway for exploring the models' understanding of linguistic and cultural differences between them. For this task, we chose the Albertina PTPT [6] and Albertina PTBR [7] models. Due to the exploratory nature of the task, smaller versions of these models were used, with 100 million parameters and 12 layers.

The experimental setup was designed to ensure fair conditions for each language variant. Since the Puntuguese dataset contains more Brazilian Portuguese puns than European Portuguese puns, the Brazilian puns were undersampled across the training, validation, and test subsets. Therefore, for this task, the training subset contains 558 Brazilian Portuguese puns, the validation subset has 79, and the test subset has 150, matching the number of European Portuguese puns in each subset.

Next, Albertina PTPT and Albertina PTBR were fine-tuned with puns from both language variants to establish a performance baseline for both models. Afterwards, Albertina PTPT was fine-tuned exclusively with European Portuguese puns, while Albertina PTBR was fine-tuned exclusively with Brazilian Portuguese puns. The goal here is to assess how a model pre-trained and fine-tuned on a specific language variant will generalize across other variants of the same language. All models were fine-tuned using the same approach and hyperparameters for the pun location task.

To compare performance across languages, these four fine-tuned models were evaluated using puns from both language variants, as well as from European Portuguese and Brazilian Portuguese separately.

The results, displayed on Table 5.5, are somewhat inconclusive regarding the models' capacity to generalize across different language variants and cultural differences.

Analyzing the results of the Albertina PTBR model, we notice that when fine-tuned on puns from both variants, the model performs best when evaluated exclusively on Brazilian Portuguese puns, achieving an F1 score of 0.68 in this scenario. Conversely, its worst performance is observed when evaluated on European Portuguese puns. Similarly, when fine-tuned solely on Brazilian Portuguese puns, the model achieves its highest performance when evaluated on puns from this variant, with an F1 score of 0.71, even surpassing the model fine-tuned on the mixed dataset. This outcome is expected, as a model pre-trained on a specific language variant is anticipated to perform better on that variant, especially when further fine-tuned with data from the same variant.

[6] `https://huggingface.co/PORTULAN/albertina-100m-portuguese-ptpt-encoder` (Accessed 16 September 2024)

[7] `https://huggingface.co/PORTULAN/albertina-100m-portuguese-ptbr-encoder` (Accessed 16 September 2024)

Table 5.5: Albertina models generalization across Portuguese variants.

	Train data	Test data	Precision	Recall	F1
Albertina PTPT	All	All	**0.70**	0.60	0.64
	All	PT-PT	0.69	0.58	0.63
	All	PT-BR	**0.70**	**0.62**	**0.66**
	PT-PT	PT-PT	0.71	0.66	0.68
	PT-PT	PT-BR	**0.72**	**0.71**	**0.71**
Albertina PTBR	All	All	0.69	0.61	0.65
	All	PT-PT	0.68	0.58	0.63
	All	PT-BR	**0.71**	**0.65**	**0.68**
	PT-BR	PT-BR	**0.73**	**0.70**	**0.71**
	PT-BR	PT-PT	0.67	0.64	0.66

However, despite the consistent under-performance of Albertina PTBR in locating puns in European Portuguese across different fine-tuning scenarios, there is an intriguing contradiction: the model actually performs better on European Portuguese puns when fine-tuned exclusively on Brazilian Portuguese data (F1 score of 0.66) compared to when it was fine-tuned on the mixed dataset (F1 score of 0.63).

The results obtained with the Albertina PTPT model further highlight this contradiction. When fine-tuned on both language variants, Albertina PTPT shows its best performance when evaluated on Brazilian Portuguese puns (F1 of 0.66), followed by mixed data (F1 of 0.64), with its weakest performance occurring when tested on European Portuguese puns (F1 of 0.63). Even more surprisingly, when fine-tuned exclusively on European Portuguese puns, the model performs best when locating Brazilian Portuguese puns (F1 of 0.71), achieving results comparable to the Albertina PTBR model fine-tuned and evaluated on Brazilian Portuguese data. This is entirely unexpected, given that Albertina PTPT was pre-trained on European Portuguese, and one would expect it to perform better with texts from this variant.

Explaining these results is challenging without a thorough analysis of the Puntuguese dataset to understand why the European Portuguese puns appear more difficult for the models to process compared to the Brazilian ones. Several factors could contribute to these outcomes. It is possible that the European Portuguese puns in the dataset are less representative of the full spectrum of linguistic particularities found in this variant, leading to a mismatch between the training data and the evaluation set. If these puns are more complex, subtle, or context-dependent, the models might struggle to generalize effectively.

Although Albertina PTPT was pre-trained on European Portuguese, the fine-tuning process may have introduced new patterns that skewed the model's performance towards Brazilian Portuguese. The fine-tuning dataset may not have been large or diverse enough to adequately reinforce the characteristics of European Portuguese, resulting

in better performance on Brazilian Portuguese data.

Additionally, the models might be overfitting to specific patterns in Brazilian Portuguese during fine-tuning, particularly if these patterns are more consistent or prevalent in the training data. Conversely, the models may be underfitting to the more varied or complex patterns found in European Portuguese, leading to poorer performance on this variant.

These findings highlight the need for a more careful approach when developing models that handle multiple language variants. The results suggest that models could benefit from more sophisticated mechanisms to better capture and interpret the linguistic and cultural subtleties that influence pun construction and comprehension across different variants of the same language.

6 Conclusion and Future Research Directions

This study addressed the challenging task of pun location in Portuguese, a significant contribution given the lack of computational humor research focused on this language.

A sequence-labeling approach was adopted for classifying words as either punning or non-punning, i.e., pun location. Several pretrained BERT-family models, including BERTimbau and Albertina, as well as a GPTNeo-based model, were fine-tuned and evaluated for this task in Puntuguese, a dataset of punning texts in Portuguese. The best-performing model was BERTimbau base, fine-tuned with positive examples, which achieved a Precision of 0.74, a Recall of 0.76 and an F1 of 0.75. We further attempted to refine the results by exploring post-processing, based on the assumption that pun words tend to be located at the end of the text. In this scenario, most of the models improved their Precision and F1, with some trade-off in Recall. With post-processing, the best performing model was BERTimbau large, with an F1 of 0.75 and a Precision of 0.77.

Since there is no baseline for comparison for this task in Portuguese, it is difficult to extract definitive conclusions from these results. If we analyze them in context with recent results for other Latin languages at JOKER CLEF 2023 (Ermakova et al., 2023), it is possible to conclude that these models show promising capabilities in localizing pun words. However, these results are not entirely comparable, since both the datasets and languages used are different.

When considering also non-punning texts, F1 scores decrease significantly. However, we show that some of the models fine-tuned on this data for pun location are also apt for pun detection, where they perform close to models fine-tuned specifically for the latter task. This dual functionality may enhance the utility of the models and contribute to the explainability of predictions by pinpointing specific trigger words within pun sentences.

This work also explored cultural differences and linguistic variations within the same language and how they are handled by language models. Although the results are somewhat inconclusive, they highlight the necessity of both quantity and variety in data to achieve good results across language variants in pun location tasks. Humor is highly subjective and culturally influenced; thus, language models may require more advanced mechanisms to address linguistic and cultural differences in humor processing tasks.

Despite the promising results, there is room for improvements, both in pun location

and pun detection for Portuguese. Future work could explore different fine-tuning strategies to enhance the results achieved so far or incorporate different techniques such as sentiment analysis or anomaly detection.

Reported performances may also be compared with zero- and few-shot learning techniques in Large Language Models (LLMs). By simple prompting with a task description and, for the few-shot, with minimal training data, this would enable to evaluate both the flexibility and effectiveness of LLMs in the target tasks.

In addition to improvements in the pun detection and pun location it would be interesting to explore the task of pun interpretation in Portuguese, as it might enhance the models' comprehension of complex language features. The current work attempted to explore this area by detecting and locating puns in the same approach, however the results still require improvement.

References

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., and Zou, J. (2019). Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*.

Attardo, S. (2008). *A primer for the linguistics of humor*, pages 101–156. De Gruyter Mouton, Berlin, New York.

Attardo, S. (2009). *Linguistic theories of humor*. Walter de Gruyter.

Black, S., Leo, G., Wang, P., Leahy, C., and Biderman, S. (2021). Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. `https://doi.org/10.5281/zenodo.4280483`.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cai, Y., Li, Y., and Wan, X. (2018). Sense-aware neural models for pun location in texts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 546–551.

Castro, S., Cubero, M., Garat, D., and Moncecchi, G. (2016). Is this a joke? detecting humor in spanish tweets. In *Advances in Artificial Intelligence-IBERAMIA 2016: 15th Ibero-American Conference on AI, San José, Costa Rica, November 23-25, 2016, Proceedings 15*, pages 139–150. Springer.

Chen, M., Du, F., Lan, G., and Lobanov, V. S. (2020). Using pre-trained transformer deep learning models to identify named entities and syntactic relations for clinical protocol analysis. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*, pages 1–8.

Chopra, A., Prashar, A., and Sain, C. (2013). Natural language processing. *International journal of technology enhancements and emerging engineering research*, 1(4):131–134.

Clemêncio, A., Alves, A., and Gonçalo Oliveira, H. (2019). Recognizing humor in portuguese: First steps. In *EPIA Conference on Artificial Intelligence*, pages 744–756. Springer.

Das, D. and Pramanick, A. (2017). Ju_cse_nlp at semeval 2017 task 7: Employing rules to detect and interpret english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pages 432–435.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diao, Y., Fan, X., Lin, H., Wu, D., Yang, L., Zhang, D., and Xu, K. (2019). Heterographic pun recognition via pronunciation and spelling understanding gated attention network. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pages 363–371.

Diao, Y., Lin, H., Wu, D., Yang, L., Xu, K., Yang, Z., Wang, J., Zhang, S., Xu, B., and Zhang, D. (2018). Weca: A wordnet-encoded collocation-attention network for homographic pun recognition. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 2507–2516.

Diao, Y., Lin, H., Yang, L., Fan, X., Wu, D., and Xu, K. (2020). Homographic pun location using multi-dimensional semantic relationships. *Soft Computing*, 24:12163–12173.

Doogan, S., Ghosh, A., Chen, H., and Veale, T. (2017). Idiom savant at semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 103–108.

Dsilva, R. R. (2023). Akranlu@ clef joker 2023: using sentence embeddings and multilingual models to detect and interpret wordplay. *Proceedings of the Working Notes of CLEF*.

Ermakova, L., Miller, T., Bosser, A.-G., Palma Preciado, V. M., Sidorov, G., and Jatowt, A. (2023). Overview of joker–clef-2023 track on automatic wordplay analysis. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 397–415. Springer.

Faraj, D. and Abdullah, M. (2021). Sarcasmdet at semeval-2021 task 7: Detect humor and offensive based on demographic factors using roberta pre-trained model. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*, pages 527–533.

Feng, J., Sevgili, Ö., Remus, S., Ruppert, E., and Biemann, C. (2020). Supervised pun detection and location with feature engineering and logistic regression. In *SwissText/KONVENS*.

Galeano, L. J. G. (2022). Ljgg@ clef joker task 3: An improved solution joining with dataset from task 1. In *CLEF (Working Notes)*, pages 1818–1827.

Gameiro, P., Inácio, M., Gonçalo Oliveira, H., and Alves, A. (2024). Sequence labeling for pun location and detection in Portuguese. In *Proceedings of 23rd EPIA Conference on Artificial Intelligence, EPIA 2024*, page In press, Viana do Castelo, Portugal.

Garcıa-Dıaz, J. A. and Valencia-Garcıa, R. (2021). Umuteam at haha 2021: Linguistic

features and transformers for analysing spanish humor. the what, the how, and to whom. In *Proceedings of the Iberian Languages Evaluation Forum (Iber-LEF 2021), CEUR Workshop Proceedings, Málaga, Spain*, volume 9.

Grover, K. and Goel, T. (2021). Haha@ iberlef2021: Humor analysis using ensembles of simple transformers. In *IberLEF@ SEPLN*, pages 883–890.

Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Huang, Y. H., Huang, H. H., and Chen, H. H. (2017). Identification of homographic pun location for pun understanding. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 797–798. International World Wide Web Conferences Steering Committee.

Inácio, M. and Oliveira, H. G. (2024). Exploring multimodal models for humor recognition in portuguese. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese*, pages 568–574.

Inácio, M., Wick-pedro, G., Ramisch, R., Espírito Santo, L., Chacon, X. S. Q., Santos, R., Sousa, R., Anchiêta, R., and Gonçalo Oliveira, H. (2024). Puntuguese: A corpus of puns in Portuguese with micro-edits. In *Proceedings of The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, page In press. ELRA and ICCL.

Inácio, M. L., Gonçalo Oliveira, H., and Wick-Pedro, G. (2023). What do humor classifiers learn? an attempt to explain humor recognition models. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 88–98. Association for Computational Linguistics (ACL).

Ismailov, A. (2019). Humor analysis based on human annotation challenge at iberlef 2019: First-place solution. In *IberLEF@ SEPLN*, pages 160–164.

Jaech, A., Koncel-Kedziorski, R., and Ostendorf, M. (2016). Phonological punderstanding. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 654–663.

Jaiswal, A. and Monika, M. (2019). Pun detection using soft computing techniques. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, pages 5–9.

Jurafsky, D., , and Martin, J. H. (2024). *Speech & language processing. Draft of February 3rd, 2024*. Accessed in 2024.

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

Kostić, M., Batanović, V., and Nikolić, B. (2023). Monolingual, multilingual and cross-lingual code comment classification. *Engineering Applications of Artificial Intelligence*, 124:106485.

Liu, L., Zhang, D., and Song, W. (2018). Exploiting syntactic structures for humor recognition. In *Proceedings of the 27th international conference on computational linguistics*.

Liu, S., Ma, M., Yuan, H., Zhu, J., Wu, Y., and Lan, M. (2021). A dual-attention neural network for pun location and using pun-gloss pairs for interpretation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13028 LNAI:688–699.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Lopes, R., Magalhaes, J., and Semedo, D. (2024). GlórIA: A generative and open large language model for Portuguese. In Gamallo, P., Claro, D., Teixeira, A., Real, L., Garcia, M., Oliveira, H. G., and Amaro, R., editors, *Proceedings of the 16th International Conference on Computational Processing of Portuguese*, pages 441–453, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.

Mao, J., Wang, R., Huang, X., and Chen, Z. (2020). Compositional semantics network with multi-task learning for pun location. *IEEE Access*, 8:44976–44982.

Martin, R. A., Puhlik-Doris, P., Larsen, G., Gray, J., and Weir, K. (2003). Individual differences in uses of humor and their relation to psychological well-being: Development of the humor styles questionnaire. *Journal of research in personality*, 37(1):48–75.

Mikhalkova, E. and Karyakin, Y. (2017). Punfields at semeval-2017 task 7: Employing roget's thesaurus in automatic pun recognition and interpretation. *arXiv preprint arXiv:1707.05479*. Redundante.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, T. and Gurevych, I. (2015). Automatic disambiguation of english puns. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 1:719–729.

Miller, T., Hempelmann, C. F., and Gurevych, I. (2017). Semeval-2017 task 7: Detection and interpretation of english puns. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 58–68.

Nakayama, H. (2018). seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Oele, D. and Evang, K. (2017). Buzzsaw at semeval-2017 task 7: Global vs. local context for interpreting and locating homographic english puns with sense embeddings. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 444–448. Association for Computational Linguistics (ACL).

Ohnesorge, F., Gutiérrez, M. Á., and Plichta, J. (2023). Clef 2023 joker tasks 2 and 3: using nlp models for pun location, interpretation and translation. In *CEUR Workshop Proceedings*, volume 3497.

Oliveira, H. G., Clemêncio, A., and Alves, A. (2020). Corpora and baselines for humour recognition in portuguese. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1278–1285.

Ortega-Bueno, R., Muñiz-Cuza, C. E., Pagola, J. E. M., and Rosso, P. (2018). Uo upv: Deep linguistic humor detection in spanish social media. In *Proceedings of the third workshop on evaluation of human language technologies for Iberian languages (IberEval 2018) co-located with 34th conference of the Spanish society for natural language processing (SEPLN 2018)*, volume 2150.

Pan, C., Song, B., Wang, S., and Luo, Z. (2021). Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584.

Pang, C., Fan, X., Su, W., Chen, X., Wang, S., Liu, J., Ouyang, X., Feng, S., and Sun, Y. (2021). abcbpc at semeval-2021 task 7: Ernie-based multi-task model for detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 286–289.

Pedersen, T. (2017). Duluth at semeval-2017 task 7: Puns upon a midnight dreary, lexical semantics for the weak and weary. *arXiv preprint arXiv:1704.08388*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peyrard, M., Borges, B., Gligorić, K., and West, R. (2021). Laughing heads: Can transformers detect what makes a sentence funny? *arXiv preprint arXiv:2105.09142*.

Popova, O. and Dadić, P. (2023). Does ai have a sense of humor? clef 2023 joker tasks 1, 2 and 3: using bloom, gpt, simplet5, and more for pun detection, location, interpretation and translation. *Proceedings of the Working Notes of CLEF*, 3.

Princeton University (2010). About wordnet. Accessed: 2024-07-21.

Pérez, A. R. (2012). *Linguistic-based Patterns for Figurative Language Processing: The Case of Humor Recognition and Irony Detection*. PhD thesis, Universitat Politècnica de València.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Raskin, V. (1979). Semantic mechanisms of humor. In *Annual Meeting of the Berkeley Linguistics Society*, pages 325–335.

Ren, L., Xu, B., Lin, H., and Yang, L. (2021). Abml: attention-based multi-task learning for jointly humor recognition and pun detection. *Soft Computing*, 25.

Rodrigues, J., Gomes, L., Silva, J., Branco, A., Santos, R., Cardoso, H. L., and Osório, T. (2023). Advancing neural encoding of portuguese with transformer albertina pt-*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Scholer, K. (2015). Tesla stock moves on april fools' joke - wsj.

Shahaf, D., Horvitz, E., and Mankoff, R. (2015). Inside jokes: Identifying humorous cartoon captions. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1065–1074.

Souza, F., Nogueira, R., and Lotufo, R. (2020a). BERTimbau: Pretrained BERT models for Brazilian Portuguese. In *Proceedings of Brazilian Conf on Intelligent Systems (BRACIS 2020)*, volume 12319 of *LNCS*, pages 403–417. Springer.

Souza, F., Nogueira, R., and Lotufo, R. (2020b). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Tagnin, S. E. (2005). O humor como quebra da convencionalidade. *Revista brasileira de linguística aplicada*, 5:247–257.

Tsai, P.-H., Chen, H.-C., Hung, Y.-C., Chang, J.-H., and Huang, S.-Y. (2021). What type of humor style do older adults tend to prefer? a comparative study of humor style tendencies among individuals of different ages and genders. *Current Psychology*, pages 1–12.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Vechtomova, O. (2017). Uwaterloo at semeval-2017 task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 421–425.

Wang, L., Lin, X., Lin, N., Fu, Y., Wu, K., and Wu, J. (2021). Humor analysis in spanish tweets with multiple strategies. *IberLEF@ SEPLN*, 2943.

Weller, O. and Seppi, K. (2019). Humor detection: A transformer gets the last laugh. *arXiv preprint arXiv:1909.00252*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T.,

Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiong, S., Wang, R., Huang, X., and Chen, Z. (2022). Multidimensional latent semantic networks for text humor recognition. *Sensors*, 22.

Yang, D., Lavie, A., Dyer, C., and Hovy, E. (2015). Humor recognition and humor anchor extraction. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376. Pun of the Day
16000 One Liners.

Zago, R. (2023). Bertugues base (aka "bertugues-base-portuguese-cased").

Zhou, Y., Jiang, J. Y., Zhao, J., Chang, K. W., and Wang, W. (2020). "the boating store had its best sail ever": Pronunciation-attentive contextualized pun recognition. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 813–822.

Zou, Y. and Lu, W. (2019). Joint detection and location of english puns. *arXiv preprint arXiv:1909.00175*.

Özge Sevgili, Ghotbi, N., and Tekir, S. (2017). N-hance at semeval-2017 task 7: A computational approach using word association for puns. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 436–439.

Appendix A - Article

"Sequence Labeling for Pun Location and Detection in Portuguese"

Published in EPIA 2024

Sequence Labeling for Pun Location and Detection in Portuguese

Patrícia Gameiro[1], Márcio Lima Inácio[2,3,4][0000−0002−0875−4574], Hugo Gonçalo Oliveira[2,3,4][0000−0002−5779−8645], and Ana Alves[1,3,4][0000−0002−3692−338X]

[1] Polytechnic Institute of Coimbra, Coimbra Institute of Engineering (ISEC)
[2] Department of Informatics Engineering, University of Coimbra, Portugal
[3] Centre for Informatics and Systems of the University of Coimbra (CISUC)
[4] Intelligent Systems Associate Laboratory (LASI)

Abstract. Detecting humor is a necessary step towards language understanding. However, work on computational humor for Portuguese is still limited. For this language, we tackle the task of pun location. With a corpus of annotated punning texts, we fine-tune available encoder models for labeling words in context as punning or not. We achieve an F1 of 0.75 with a BERT-based model and further improve precision with post-processing. Moreover, we show that a model trained for pun location can be used for pun detection as well, performing close to a model specifically trained on the latter task, but with the advantage of identifying the pun words, thus contributing to explainability.

Keywords: Computational Humor · Pun location · Pun detection · Pun disambiguation · Computational Processing of Portuguese.

1 Introduction

Humor is a complex and context-dependent aspect of human communication. Since using and understanding it denotes fluency in the target language [33], the automatic detection of humor has been tackled in the scope of Natural Language Processing (NLP), to further enhance the capabilities of current computational systems. As the use of language models becomes more widespread, enabling humor recognition can have truly impactful outcomes, enhancing interactions and decision-making processes across various sectors.

While there is diverse published work in humor recognition and pun related tasks in various languages, research in Portuguese has primarily focused on identifying humor in general [4, 17] or specifically detecting puns [15], where texts are classified as humorous or not.

This study introduces the task of pun location to Portuguese, which, to the best of our knowledge, has not been explored previously. Pun location involves identifying pun words within texts, thereby not only detecting humor but also contributing to the explainability of the results. This is a significant step beyond mere detection, as it provides insights into the specific elements within the text that trigger the humorous effect.

For this purpose, a sequence-labeling approach is employed on Puntuguese [16], a recently compiled corpus of punning texts in Portuguese, where pun words are manually annotated. Various encoder language models are fine-tuned for this task, and post-processing is applied to further enhance the model's performance. Moreover, we leverage on the resulting models for the task of pun detection, where performance was close to previous approaches exclusive for this task, with the advantage of identifying which words contribute to the pun.

The main contributions of this work are: an assessment of the performance of Portuguese encoder models in humor-related tasks; explorations on the identification of pun words in Portuguese texts, towards further advances in computational humor for this language; and a methodological framework that both detects puns and locates the pun words, allowing for outcome explainability.

The remainder of the paper is structured as follows: Section 2 reviews related works in computational humor, particularly focusing on pun detection, location and interpretation. Section 3 describes the methodology, including the dataset, models, and experimental setup. Section 4 reports the results of the models in both pun location and detection tasks. Section 5 closes with the main conclusions of the study and potential directions for future research.

2 Related Work

Computational Humor has two main subareas: humor generation and humor recognition. The latter has focused on specific types of humor, such as the pun, a form of wordplay that exploits ambiguity towards a humorous effect.

Different pun-related tasks have been tackled by the community, and three of them were part of SemEval 2017 [23] Task 7: (i) pun detection, where given contexts are classified as containing a pun or not; (ii) pun location, where pun words are identified in given contexts; (iii) pun interpretation, also known as pun disambiguation, where pun words are associated to senses corresponding to each of their meanings.

Regarding orthography, puns can be homographic, when they take advantage of words with different meanings but the same orthography (homonymy); or heterographic, when orthography is different. Concerning pronunciation, puns can additionally be homophonic, when the words have the same sound.

Given their nature, much work targeting homographic puns resorts to techniques for word sense disambiguation (WSD) [24, 31, 26, 8, 19], whereas, for homophonic puns, phonetic features can be useful [18, 10, 35].

As in many NLP tasks, studies have explored different types of embeddings, including sense [8, 2, 19], contextualized [35, 28], pronunciation [35], character [36, 7], or more traditional Word2Vec and GloVe [31, 13]. Alternatively, word associations were computed with the PMI [34, 31, 13], and rarity approximated with the IDF [34]. Parts of speech have also been explored [5, 22, 34, 36, 9, 13, 28, 31] and, assuming that most puns are located at the end of a text, some studies consider the position of words [14, 34, 36, 7].

For English, recent work has approached pun detection with a range of algorithms for supervised classification, such as bi-LSTM networks [8], convolutional neural networks [7], or an attention-based multi-task learning model [28]. For Portuguese, related work used traditional supervised methods for classifying short texts as humorous or not [4], exploiting TF-IDF and a set of handcrafted features. BERT was subsequently fine-tuned for the purpose [17] and achieved almost perfect performance. The focus on puns is very recent, with pun detection performed on Puntuguese [16], a new corpus of puns in Portuguese. Different BERT models, some of them integrating handcrafted features, were used for this purpose [15]. Since puns are manually annotated, the release of Puntuguese opens the door not only to pun detection, but also to pun location in this language.

Pun location generally requires the text to be processed as a sequence of words. Previous approaches include: a bi-LSTM that modeled the sequence of word senses produced by different WSD methods [2]; a compositional network and multi-task learning leveraging different levels of embeddings for considering long- (character, word) and short-distance (n-gram) semantic relations between words [21]; and a dual-attentive network that integrates word sense and pronunciation embeddings with context information [19]. The latter study claims to present the best results so far for this task in the English language, with F1 scores of 0.90 and 0.92, respectively for homographic and heterographic puns.

Given their interconnection, many studies addressed pun detection and location jointly, e.g., with a bi-LSTM-CRF [36], contextualized and phonological embeddings [35], or logistic regression and feature engineering, using statistic and semantic properties of puns [13].

The JOKER track in CLEF [12] is a more recent evaluation that encompassed the three aforementioned sub-tasks and, additionally, pun translation, in English, Spanish and French. Most participants relied on transformers like GPT-3, BLOOMZ, and SimpleT5, which achieved the best performance in pun location in English [25] and Spanish [27]. One approach relied on XLM-RoBERTa for locating puns via sequence labeling [11], with promising results in English and the best performance in French; even if, for both French and Spanish, accuracy remained below 0.6. This method resembles the one we propose.

In summary, a broad range of computational approaches was applied to pun detection, location and interpretation, which highlights the complexity of such tasks. However, the majority of works were for English. To the best of our knowledge, work for Portuguese is limited to the least complex task of pun detection.

3 Methodology

This section describes the methodology adopted in this work, including the description of the dataset used as well as the experimental setup, which details the approach and selected pretrained models.

The designed approach addresses pun location as a sequence-labeling task, using encoder models of the BERT family, pretrained for Portuguese. The goal

is to obtain a model capable of, given a short text with a pun, output its words correctly labeled as pun (1) or non-pun (0).

3.1 Dataset

Experimentation is performed on the Puntuguese [16] dataset, as available in the HuggingFace platform[5]. Puntuguese is a curated collection of punning texts in both Brazilian and European Portuguese with a total of 2,850 puns, 2,053 attributed to Brazilian Portuguese and 797 to European Portuguese.

The dataset further categorizes puns into four distinct types: homophonic (572 puns), homographic (6 puns), a combination of both homophonic and homographic (504 puns), and those that are neither (1,827 puns). Each pun is manually annotated with its punning mechanisms, namely: the pun words (i.e., the triggers for the text to be considered a pun) and their alternative signs. In addition to the punning texts, Puntuguese includes a non-humorous counterpart for each entry, created through micro-editing. This allows to train models for pun location and for pun detection tasks.

Each example in the dataset contains the corresponding sequence of labels for the words in text: 1 for pun words and 0 for non-pun words. Figure 1 shows two examples of punning texts from the dataset and their respective labeling. In the first example[6] "*estado*" is the trigger for the pun, hence it is the only word labeled as 1, with all the others as 0. The second example[7] contains two pun triggers, "*Hobbits*" and "*Hobbyte*", therefore, the sequence includes two words labeled with 1.

Em	que	estado	se	encontra	o	rio	Mississipi	?	No	estado	líquido	.
0	0	0	0	0	0	0	0	0	0	**1**	0	0

O	que	são	oito	Hobbits	?	Um	Hobbyte	.
0	0	0	0	1	0	0	1	0

Fig. 1. Examples of punning texts in Puntuguese and their labeling.

3.2 Experimental setup

The Puntuguese dataset is split into training (70%), test (20%), and validation (10%) subsets, using a stratified sampling approach to maintain an even distribution of pun types and language varieties. This ensured that each subset mirrored the overall composition of the corpus in terms of the proportion of homophonic, homographic, mixed, and non-humorous text, as well as the balance between Brazilian and European Portuguese entries. The data was also preprocessed to convert all text to lowercase.

[5] https://huggingface.co/datasets/Superar/Puntuguese
[6] In English, translated to "In what state is the Mississippi River? In the liquid state."
[7] In English, "What are eight Hobbits? A Hobbyte."

Multiple models of the BERT [6] family, pretrained for Portuguese, were fine-tuned for pun location in the training portion of Puntuguese, and evaluated in its test portion. The hyperparameters, selected empirically for training each model, included a learning rate of 3e-5, batch sizes of 8 for both training and evaluation, a training duration of 4 epochs, and weight decay set to 0.02.

The following models, all available from the HuggingFace Hub, were considered: BERTimbau [32], base[8] [32] and large[9]; Albertina [29], PTPT[10] and PTBR[11]; BERTugues[12]; RoBERTa PT-BR[13]. BERTimbau base and BERTugues are relatively lightweight, each with 110M parameters across 12 layers. In contrast, BERTimbau large and, especially, Albertina, are significantly more resource demanding. BERTimbau large has 335M parameters and 24 layers, while each Albertina model has 887M parameters and 24 layers. RoBERTa PT-BR offers a slight increase in complexity over the lighter models, with 125M parameters. All models are pretrained in Brazilian Portuguese text, with the exception of Albertina PTPT, which is pretrained for European Portuguese.

Regarding pun location, models were evaluated with Precision, Recall and F1, considering the sequence labeling of pun words (i.e., words labeled as 1), computed with the HuggingFace *evaluate* library[14]. Fine-tuned models were assessed in two distinct scenarios: (i) considering exclusively the positive examples in Puntuguese, i.e., texts with at least a pun word, enabling the model to focus its attention on identifying exactly where the puns are, making it more effective for pun location tasks; (ii) considering both positive and negative examples, i.e., not only the punning texts, but also the texts without pun words. In this latter case, the model is expected to distinguish further whether a text contains a pun, making it also suited for pun detection.

Each example was tokenized with the respective tokenizer of each model, ensuring that labels were properly aligned. All subwords and special tokens were assigned a label of -100, which was ignored during model training and evaluation.

4 Results

This section describes the results achieved for the tasks of pun location, followed by the exploration of post-processing, aiming at improving performance. Finally, pun detection is addressed through pun location and its results are compared with those of models fine-tuned for pun detection directly.

[8] https://huggingface.co/neuralmind/bert-base-portuguese-cased
[9] https://huggingface.co/neuralmind/bert-large-portuguese-cased
[10] https://huggingface.co/PORTULAN/albertina-900m-portuguese-ptpt-encoder
[11] https://huggingface.co/PORTULAN/albertina-900m-portuguese-ptbr-encoder
[12] https://huggingface.co/ricardoz/BERTugues-base-portuguese-cased
[13] https://huggingface.co/josu/roberta-pt-br
[14] https://huggingface.co/docs/evaluate/index

4.1 Pun location

Table 1 has the models' performances in the pun location task, using only positive examples in the fine-tuning process. In this scenario, BERTimbau base achieved the best overall performance, with a Precision of 0.74, a Recall of 0.76, and an F1 of 0.75. BERTimbau large also achieved solid results, with an F1 of 0.74, while Albertina PTPT and BERTugues performed very close.

Table 1. Performance of models fine-tuned with positive examples only.

Model	**Precision**	**Recall**	**F1**
Albertina PTBR	0.70	0.64	0.67
Albertina PTPT	0.72	0.73	0.73
BERTimbau base	**0.74**	**0.76**	**0.75**
BERTimbau large	**0.74**	0.75	0.74
BERTugues	0.71	0.72	0.72
RoBERTa PTBR	0.64	0.64	0.64

Although these initial results might not be too impressive, we stress that there are many more non-pun than pun words. Moreover, analyzing the predictions closely shows that the models are not that far from correctly identifying the pun triggers. For example, in the text of Figure 2, annotated with the pun word "estado", the model correctly identifies the second instance of this word, while also classifying "Mississipi" and "líquido" as pun words. Despite not being ambiguous in the context of this text, these terms do contribute to explaining the pun and its humorous effect.

	Em	que	estado	se	encontra	o	rio	Mississipi	?	No	estado	líquido	.
Labels	0	0	0	0	0	0	0	0	0	0	**1**	0	0
Prediction	0	0	0	0	0	0	0	**1**	0	0	**1**	**1**	0

Fig. 2. Example of pun location prediction with BERTimbau Large.

Table 2 has the same results, but for the more challenging scenario, with both positive and negative examples in Puntuguese. All models consistently perform worse, which was expected due to the increased imbalance in word labels, now with even more non-pun than pun words. Additionally, in opposition to the first scenario, these models cannot assume that there is at least one pun word in each text. Despite the more variable performance across models, with an F1 score of 0.57, BERTimbau large was the best performing in this second scenario.

4.2 Labeling post-processing

Several studies [14, 36] take advantage of the fact that pun word tends to be at the end of the humorous text [1]. With this in mind, we try to improve performance by further post-processing the prediction of the fine-tuned models.

Table 2. Performance of models fine-tuned with the entire dataset.

Model	Precision	Recall	F1
Albertina PTBR	**0.59**	0.44	0.51
Albertina PTPT	0.57	0.44	0.50
BERTimbau base	0.54	0.52	0.53
BERTimbau large	**0.59**	**0.56**	**0.57**
BERTugues	0.55	0.51	0.53
RoBERTa PTBR	0.47	0.42	0.44

The first method (hereafter, LW) analyzes each text backwards, in order to consider the last positively-labeled word as the only pun, while disregarding any other words possibly labeled as such. This primarily aims to minimize false positives, which occur when non-pun words are mistakenly identified as puns. By limiting the recognition to the first encountered pun word, the method effectively reduces the likelihood of incorrectly labeling regular words as puns, enhancing the precision of the location prediction. For instance, in the pun of Figure 3[15] there is only one trigger word (*"calcular"*), thus, cleaning all false positives before this word makes the automatic labeling more precise.

	Qual	é	a	pacifista	que	é	um	gênio	da	matemática	?	Madre	Teresa	de	Calcular	.
Labels	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0
Pred	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	**1**	0
LW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0

Fig. 3. Example of pun location prediction with BERTimbau large, and improvement with the LW post-processing method.

However, this approach can inadvertently increase the rate of false negatives, i.e., actual puns not identified. Since only the first detected pun word is considered, any preceding puns in the text that could be valid are missed. This could be particularly limiting in complex examples where multiple puns are present.

The second method (hereafter, LS) also processes the text from the end. Still, it differs by considering a continuous sequence of identified pun words as valid, up to the point where no further puns are detected, and removing pun annotations from the rest of the text. This strategy tends to be less punitive than the former, thus reducing false negatives by recognizing each pun appearing before a non-pun segment. This is illustrated in Figure 4, with a pun[16] where the LW method would remove more positive labels than necessary, but the LS method works.

Table 3 shows the performance after applying each of the pre-processing methods. LW generally increases precision but significantly reduces recall by

[15] In English, "What do you call a pacifist who's a math genius? Mother Teresa of Calculus." "Calculus" and "Calcutta" are pronounced similarly in Portuguese

[16] In English, "Which animal dissolves in water? Orangu Tang."

	Qual	animal	se	dissolve	na	água	?	Orango	Tang	.
Labels	0	0	0	0	0	0	0	**1**	**1**	0
Pred	0	0	0	0	0	0	0	**1**	**1**	0
LW	0	0	0	0	0	0	0	0	**1**	0
LS	0	0	0	0	0	0	0	**1**	**1**	0

Fig. 4. Example of pun location prediction with BERTimbau large, followed by results after the post-processing methods LW and LS.

missing valid pun words that occur before the first identified. Conversely, LS provides a more balanced approach, slightly reducing recall, but maintaining or improving F1 and precision.

With post-processing, the best-performing model was BERTimbau large with LS, achieving a Precision of 0.77, a Recall of 0.73 and a F1 of 0.75. When compared to the performance without post-processing (Table 1), the F1 of this model improved by 1 percentage point (0.74 *vs* 0.75), especially due to the increase in precision. However, for other models, the slight decrease in recall makes the overall impact hardly noticeable. This also suggests that, in many cases, the model is already labeling a single sequence as the pun.

Table 3. Results of post-processing approaches applied to fine-tuned models' predictions for pun location.

Model	**LW**			**LS**		
	P	**R**	**F1**	**P**	**R**	**F1**
Albertina PTBR	0.61	0.55	0.58	0.71	0.64	0.67
Albertina PTPT	0.63	0.61	0.62	0.74	0.71	0.73
BERTimbau base	0.63	0.60	0.62	0.76	**0.73**	**0.75**
BERTimbau large	0.63	0.61	0.62	**0.77**	**0.73**	**0.75**
BERTugues	0.60	0.57	0.58	0.73	0.69	0.71
RoBERTa PTBR	0.59	0.56	0.57	0.67	0.63	0.65

4.3 Pun detection through pun location

We further explored the previous models for the task of pun detection. This was based on the simple assumption that a text is classified as a pun if it has at least one word labeled as pun (1). Otherwise, a text with non-pun words only (0) is classified as non-pun. For this, we use the models trained in the second scenario, which, despite the lower performance in pun location (see Table 2), were the ones exposed to both punning and non-punning texts.

Table 4 reports on the performances achieved in this task by each trained model. For reference, we also include performances by models fine-tuned directly for pun detection [15] as a whole sequence classification task, here replicated in the most recent version of the dataset.

As expected, models trained to locate pun words, a different task, generally perform below those trained for pun detection. For instance, if tasked for locating

pun words first, the performance of the Albertina models decreases significantly when compared to the same models trained directly for pun detection (F1 of 0.51 *vs* 0.69 and 0.71). For BERTimbau, this behavior is not so clear. When fine-tuned for pun location, the base model is better (F1 of 0.65 *vs* 0.51), whereas the large model has a lower precision (0.71 *vs* 0.73), but a higher recall (0.63 *vs* 0.59).

Table 4. Pun detection through sequence labeling versus directly (Inácio et al. [15])

	Model	**Precision**	**Recall**	**F1**
Seq. Labeling	Albertina PTBR	0.76	0.39	0.51
	Albertina PTPT	0.74	0.39	0.51
	BERTimbau large	0.71	0.63	0.67
	BERTimbau base	0.67	0.63	0.65
	BERTugues	0.67	0.57	0.61
	RoBERTa PTBR	0.61	0.47	0.53
Inácio et al.	Albertina PTBR	**0.77**	0.64	0.69
	Albertina PTPT	0.72	**0.69**	**0.71**
	BERTimbau large	0.73	0.59	0.65
	BERTimbau base	0.65	0.42	0.51
	Bertugues	0.69	0.54	0.61
	RoBERTa PTBR	0.57	0.59	0.58

While models trained for pun word location do not always outperform those by Inácio et al.'s [15], they still offer an advantage. Specifically, by identifying pun words, these models provide an explanation of why a text is classified as a pun. This is particularly beneficial for applications requiring not just classification, but also a reason for the decision, enhancing the output explainability.

5 Conclusion and Future Research

We addressed the challenging task of pun location in Portuguese, a significant contribution due to the lack of computational humor research on this language.

A sequence-labeling approach was adopted for classifying words as either punning or non-punning, i.e., pun location. Several pretrained BERT-family models, including BERTimbau and Albertina, were fine-tuned and evaluated for this task in Puntuguese, a dataset of punning texts in Portuguese. The best-performing model was BERTimbau base, fine-tuned with positive examples, which achieved an F1 of 0.75. We further attempted to refine the results by exploring post-processing, based on the assumption that pun words tend to be located at the end of the text. In this scenario, most of the models improved their Precision and F1, with some trade-off in Recall.

When considering also non-punning texts, F1 scores decrease significantly. However, we show that some of the models fine-tuned on this data for pun location are also apt for pun detection, where they perform close to models

fine-tuned specifically for the latter task. This dual functionality may enhance the utility of the models and contribute to the explainability of predictions by pinpointing specific trigger words within pun sentences. The source code used in our experimentation was released in a public repository[17].

Despite the promising results, there is room for improvements, both in pun location and pun detection for Portuguese. Future research for Portuguese should also consider fine-tuning encoder-decoder models, as PTT5 [3], or decoder-only models, as GlórIA [20] or Gervásio [30]. Reported performances may also be compared with zero- and few-shot learning techniques in Large Language Models (LLMs). By simple prompting with a task description and, for the few-shot, with minimal training data, this would enable to evaluate both the flexibility and effectiveness of LLMs in the target tasks.

In addition to improvements in the pun detection and pun location it would be interesting to explore the task of pun interpretation in Portuguese, as it might enhance the models' comprehension of complex language features. Moreover, the Puntuguese dataset could be leveraged for training models for pun-related tasks exclusively in European Portuguese and in Brazilian Portuguese. This would not only assess the models' performance across these variants, but also provide a pathway for exploring the models' understanding of linguistic and cultural differences between them.

Acknowledgements: This work was partially 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), within the scope of the project CISUC (UID/CEC/00326/2020).

References

1. Attardo, S.: Linguistic theories of humor. Walter de Gruyter (2009)
2. Cai, Y., Li, Y., Wan, X.: Sense-aware neural models for pun location in texts. In: Procs of 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 546–551 (2018)
3. Carmo, D., Piau, M., Campiotti, I., Nogueira, R., Lotufo, R.: PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data. arXiv preprint arXiv:2008.09144 (2020)
4. Clemêncio, A., Alves, A., Gonçalo Oliveira, H.: Recognizing humor in Portuguese: First steps. In: EPIA Conference on Artificial Intelligence. pp. 744–756. Springer (2019)
5. Das, D., Pramanick, A.: Ju_cse_nlp at semeval 2017 task 7: Employing rules to detect and interpret english puns. In: Procs of 11th International Workshop on Semantic Evaluations (SemEval-2017). pp. 432–435 (2017)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Procs 2019 Conference of the North American Chapter of the Association for Computational Linguistics:

[17] https://github.com/NLP-CISUC/Seq-labelling-puns

Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. ACL Press (Jun 2019)
7. Diao, Y., Fan, X., Lin, H., Wu, D., Yang, L., Zhang, D., Xu, K.: Heterographic pun recognition via pronunciation and spelling understanding gated attention network. The Web Conference 2019 - Procs of World Wide Web Conference, WWW 2019 pp. 363–371 (2019)
8. Diao, Y., Lin, H., Wu, D., Yang, L., Xu, K., Yang, Z., Wang, J., Zhang, S., Xu, B., Zhang, D.: Weca: A wordnet-encoded collocation-attention network for homographic pun recognition. Procs of 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP pp. 2507–2516 (2018)
9. Diao, Y., Lin, H., Yang, L., Fan, X., Wu, D., Xu, K.: Homographic pun location using multi-dimensional semantic relationships. Soft Computing **24**, 12163–12173 (2020)
10. Doogan, S., Ghosh, A., Chen, H., Veale, T.: Idiom savant at SemEval-2017 Task 7: Detection and interpretation of english puns. In: Procs of 11th international workshop on semantic evaluation (SemEval-2017). pp. 103–108 (2017)
11. Dsilva, R.R.: AKRaNLU@ CLEF JOKER 2023: using sentence embeddings and multilingual models to detect and interpret wordplay. In: CLEF (Working Notes). pp. 1846–1853 (2023)
12. Ermakova, L., Miller, T., Bosser, A.G., Palma Preciado, V.M., Sidorov, G., Jatowt, A.: Overview of JOKER–CLEF-2023 track on automatic wordplay analysis. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 397–415. Springer (2023)
13. Feng, J., Sevgili, Ö., Remus, S., Ruppert, E., Biemann, C.: Supervised pun detection and location with feature engineering and logistic regression. In: SwissText/KONVENS (2020)
14. Huang, Y.H., Huang, H.H., Chen, H.H.: Identification of homographic pun location for pun understanding. In: Procs of 26th International Conference on World Wide Web Companion. pp. 797–798. International World Wide Web Conferences Steering Committee (2017)
15. Inácio, M., Gonçalo Oliveira, H.: Exploring multimodal models for humor recognition in Portuguese. In: Procs of 16th International Conference on Computational Processing of Portuguese (PROPOR). pp. 568–574 (2024)
16. Inácio, M., Wick-pedro, G., Ramisch, R., Espírito Santo, L., Chacon, X.S.Q., Santos, R., Sousa, R., Anchiêta, R., Gonçalo Oliveira, H.: Puntuguese: A corpus of puns in Portuguese with micro-editions. In: Procs of 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING). p. In press. ELRA and ICCL (2024)
17. Inácio, M.L., Gonçalo Oliveira, H., Wick-Pedro, G.: What do humor classifiers learn? an attempt to explain humor recognition models. In: Procs of 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. pp. 88–98. ACL (ACL) (9 2023)
18. Jaech, A., Koncel-Kedziorski, R., Ostendorf, M.: Phonological pun-derstanding. In: Procs of 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 654–663. ACL (2016)
19. Liu, S., Ma, M., Yuan, H., Zhu, J., Wu, Y., Lan, M.: A dual-attention neural network for pun location and using pun-gloss pairs for interpretation **13028**, 688–699 (2021)

20. Lopes, R., Magalhaes, J., Semedo, D.: GlórIA: A generative and open large language model for Portuguese. In: Procs of 16th International Conference on Computational Processing of Portuguese. pp. 441–453. ACL, Santiago de Compostela, Galicia/Spain (Mar 2024)
21. Mao, J., Wang, R., Huang, X., Chen, Z.: Compositional semantics network with multi-task learning for pun location. IEEE Access **8**, 44976–44982 (2020)
22. Mikhalkova, E., Karyakin, Y.: Punfields at SemEval-2017 Task 7: Employing Roget's Thesaurus in automatic pun recognition and interpretation. arXiv preprint arXiv:1707.05479 (2017)
23. Miller, T., Hempelmann, C.F., Gurevych, I.: SemEval-2017 Task 7: Detection and Interpretation of English Puns. Procs of Annual Meeting of the Association for Computational Linguistics pp. 58–68 (2017)
24. Oele, D., Evang, K.: Buzzsaw at SemEval-2017 Task 7: Global vs. local context for interpreting and locating homographic english puns with sense embeddings. In: Procs of 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 444–448. ACL (2017)
25. Ohnesorge, F., Gutiérrez, M.Á., Plichta, J.: CLEF 2023 JOKER tasks 2 and 3: using NLP models for pun location, interpretation and translation. In: CEUR Workshop Proceedings. vol. 3497 (2023)
26. Pedersen, T.: Duluth at SemEval-2017 Task 7 : Puns upon a midnight dreary, lexical semantics for the weak and weary. In: Procs of 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 416–420. ACL, Vancouver, Canada (Aug 2017)
27. Popova, O., Dadić, P.: Does ai have a sense of humor? clef 2023 joker tasks 1, 2 and 3: using bloom, gpt, simplet5, and more for pun detection, location, interpretation and translation. Procs of Working Notes of CLEF **3** (2023)
28. Ren, L., Xu, B., Lin, H., Yang, L.: Abml: attention-based multi-task learning for jointly humor recognition and pun detection. Soft Computing **25** (2021)
29. Rodrigues, J., Gomes, L., Silva, J., Branco, A., Santos, R., Cardoso, H.L., Osório, T.: Advancing neural encoding of portuguese with transformer albertina pt. In: EPIA Conference on Artificial Intelligence. pp. 441–453. Springer (2023)
30. Santos, R., Silva, J., Gomes, L., Rodrigues, J., Branco, A.: Advancing generative AI for Portuguese with open decoder Gervásio PT-* (2024)
31. Özge Sevgili, Ghotbi, N., Tekir, S.: N-hance at SemEval-2017 Task 7: A computational approach using word association for puns. Procs of Annual Meeting of the Association for Computational Linguistics pp. 436–439 (2017)
32. Souza, F., Nogueira, R., Lotufo, R.: BERTimbau: Pretrained BERT models for Brazilian Portuguese. In: Procs of Brazilian Conf on Intelligent Systems (BRACIS 2020). LNCS, vol. 12319, pp. 403–417. Springer (2020)
33. Tagnin, S.E.: O humor como quebra da convencionalidade. Revista brasileira de linguística aplicada **5**, 247–257 (2005)
34. Vechtomova, O.: Uwaterloo at SemEval-2017 Task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In: Procs of 11th international workshop on semantic evaluation (SemEval-2017). pp. 421–425 (2017)
35. Zhou, Y., Jiang, J.Y., Zhao, J., Chang, K.W., Wang, W.: "The Boating Store Had its Best Sail Ever": Pronunciation-attentive contextualized pun recognition. In: Procs of 58th Annual Meeting of the Association for Computational Linguistics. pp. 813–822. ACL, Online (Jul 2020)
36. Zou, Y., Lu, W.: Joint detection and location of English puns. In: Procs of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). ACL (2019)

Instituto Superior de Engenharia
Politécnico de Coimbra