

GIORDANO HENRIQUE SILVEIRA

AINDA ESTOU PENSANDO EM UM TÍTULO LEGAL PARA MINHA DISSERTAÇÃO

(versão pré-defesa, compilada em 4 de dezembro de 2025)

Trabalho apresentado como requisito parcial à conclusão
do Curso de Bacharelado em Ciência da Computação,
Setor de Ciências Exatas, da Universidade Federal do
Paraná.

Área de concentração: *Computação*.

Orientador: Fabiano Silva.

CURITIBA PR

2025

RESUMO

O sarcasmo, presente no cotidiano, é frequentemente empregado para criticar, muitas vezes com humor, pessoas, instituições ou eventos. Por depender de contexto, conhecimento de mundo, tom e implicaturas pragmáticas, pode ser difícil de compreender até mesmo para seres humanos, já que o sentido pretendido geralmente contraria o literal. Consequentemente, a detecção automática de sarcasmo constitui um desafio relevante em Processamento de Linguagem Natural (PLN), pois erros na sua identificação podem distorcer tarefas a jusante, como análise de sentimento e mineração de opiniões. Nos últimos anos, o tema ganhou destaque justamente por seu impacto na qualidade dessas aplicações e pela necessidade de modelos capazes de captar nuances semânticas e contextuais. Neste trabalho, foram abordados dois modelos para a detecção de sarcasmo. O primeiro modelo é baseado em BiLSTM (Bidirectional Long Short-Term Memory), já o segundo modelo é baseado em BERT (Bidirectional Encoder Representations from Transformers), juntamente com BiLSTM. Como conjunto de dados, foram utilizadas manchetes sarcásticas retiradas do site The Onion e traduzidas para o português utilizando a API do ChatGPT. Ao final do trabalho, é feita uma comparação entre os dois modelos. Conclui-se que o modelo BERT com BiLSTM foi superior, alcançando F1 de 0.85.

Palavras-chave: Sarcasmo. BERT. LSTM.

ABSTRACT

The abstract should be the English translation of the “resumo”, no more, no less.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Keywords: Sarcasm. BERT. LSTM.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

DINF	Departamento de Informática
UFPR	Universidade Federal do Paraná

LISTA DE SÍMBOLOS

α	alfa, primeira letra do alfabeto grego
β	beta, segunda letra do alfabeto grego
γ	gama, terceira letra do alfabeto grego
ω	ômega, última letra do alfabeto grego
π	pi
τ	Tempo de resposta do sistema
θ	Ângulo de incidência do raio luminoso

SUMÁRIO

1 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos teóricos fundamentais para o desenvolvimento deste trabalho. Serão abordados tópicos relacionados ao aprendizado de máquina, processamento de linguagem natural, redes neurais recorrentes e Long Short-Term Memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), a base de dados utilizada para o treinamento e avaliação dos modelos propostos, bem como as métricas de avaliação empregadas. Esses conceitos são essenciais para compreender as técnicas e metodologias aplicadas na detecção de sarcasmo.

1.1 PROCESSAMENTO DE LINGUAGEM NATURAL

Processamento de Linguagem Natural (PLN) é um campo de pesquisa que tem como objetivo desenvolver sistemas e métodos para permitir que computadores compreendam, interpretem e gerem linguagem humana de forma automática (?). É "natural" no sentido de que a linguagem é humana, em oposição a linguagens de programação ou outras formas de comunicação estruturada.

Este campo de pesquisa divide-se em duas áreas principais (?): a interpretação da linguagem natural (ILN) e a geração de linguagem natural (GLN). A ILN concentra-se na compreensão, ou análise, e interpretação da linguagem humana, permitindo que os computadores extraíam significado e informações úteis a partir de textos ou fala. Uma aplicação comum da ILN é um *chatbot*, que pode entender perguntas feitas em linguagem natural e fornecer respostas relevantes. Já a GLN foca na criação de textos ou fala em linguagem humana de forma automática. Isso envolve a geração de textos coerentes e contextualmente apropriados, como resumos automáticos de documentos, geração de relatórios ou, usando novamente o exemplo do *chatbot*, a geração de respostas em linguagem natural para as perguntas dos usuários.

Neste trabalho, o foco está na ILN, especificamente na detecção de sarcasmo em títulos de notícias. Essa tarefa envolve a análise e interpretação de textos para identificar expressões sarcásticas, o que requer uma compreensão entre as palavras, o contexto e as nuances da linguagem utilizada. Para tal, modelos de aprendizado de máquina são treinados para capturar essas relações e realizar a classificação dos títulos como sarcásticos ou não sarcásticos. Modelos podem ser treinados do zero, ou podem ser utilizados modelos pré-treinados, que já possuem um conhecimento prévio sobre a linguagem, e são ajustados para a tarefa específica de detecção de sarcasmo. Treinar um modelo é atualizá-lo e uma das formas de fazer isso é o treinamento continuado.

O treinamento continuado é treinar o modelo pré-treinado em uma nova base de dados, que é específica para a tarefa desejada que é diferente da base de dados utilizada no pré-treinamento, porém mantendo a mesma tarefa geral. A detecção de sarcasmo em títulos de notícias é uma tarefa específica dentro do campo mais amplo do processamento de linguagem

natural: a análise de sentimentos. Portanto, o treinamento continuado é uma abordagem adequada para adaptar modelos pré-treinados para essa tarefa específica. Existem dois tipos de treinamento continuado: O treinamento continuado com foco na tarefa (TAPT - Task Adaptive Pre-Training) e o treinamento continuado com foco na adaptação ao domínio (DAPT - Domain Adaptive Pre-Training) (?).

O TAPT envolve a adaptação de um modelo pré-treinado para uma tarefa específica, utilizando uma base de dados não rotulada relacionada à tarefa. No caso da detecção de sarcasmo em títulos de notícias, o TAPT poderia ser realizado utilizando uma grande coleção de títulos de notícias, sem rótulos, para ajustar o modelo pré-treinado. Isso permitiria que o modelo aprendesse as características específicas dos títulos de notícias, melhorando sua capacidade de detectar sarcasmo nessa tarefa específica. Já o DAPT está preocupado com o domínio. Neste caso, o modelo é treinado por mais algum tempo em um domínio específico. Detecção de sarcasmo em títulos de notícias é uma tarefa que envolve o domínio das notícias, portanto, a abordagem utilizada neste trabalho foi o DAPT, onde o modelo pré-treinado é ajustado utilizando uma base de dados de títulos de notícias, tanto sarcásticos quanto não sarcásticos.

Treinar um modelo é uma tarefa de aprendizado de máquina, tópico este que será abordado na próxima seção. Nas seções seguintes, serão apresentados os modelos de aprendizado de máquina utilizados neste trabalho, Redes Neurais Recorrentes (RNNs) e Long Short-Term Memory (LSTM), e o modelo BERT.

1.2 APRENDIZADO DE MÁQUINA

De acordo com (?):

"Um programa de computador aprende com a experiência E em relação a uma tarefa T e uma medida de desempenho P, se seu desempenho em T, medido por P, melhora com a experiência E."

Isto significa que um sistema de aprendizado de máquina é capaz de melhorar seu desempenho em uma tarefa específica à medida que é exposto a mais dados ou experiências relacionadas a essa tarefa.

O aprendizado de máquina pode ser dividido em três categorias principais (?), aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. No aprendizado supervisionado, o modelo é treinado com um conjunto de dados rotulado, onde cada exemplo de entrada está associado a uma saída desejada. O objetivo do modelo é aprender a mapear entradas para saídas corretas, de forma que possa fazer previsões precisas em novos dados. Já no aprendizado não supervisionado, o modelo é treinado com um conjunto de dados não rotulado, onde o objetivo é descobrir padrões ou estruturas subjacentes nos dados. Isso pode incluir tarefas como agrupamento (clustering) ou redução de dimensionalidade. Por fim, no aprendizado por reforço, o modelo aprende a tomar decisões sequenciais em um ambiente, recebendo recompensas

ou punições com base em suas ações. O objetivo é aprender uma política que maximize a recompensa acumulada ao longo do tempo.

O Deep Learning (aprendizado profundo) (?) é uma subárea do aprendizado de máquina que se concentra no uso de redes neurais profundas, ou seja, redes neurais com múltiplas camadas ocultas. Essas redes são capazes de aprender representações hierárquicas dos dados, permitindo que o modelo capture padrões complexos e relações não lineares. O Deep Learning tem sido particularmente bem-sucedido em tarefas como reconhecimento de imagem, processamento de linguagem natural e jogos, onde grandes quantidades de dados e poder computacional estão disponíveis.

Neste trabalho, o foco está no aprendizado supervisionado, onde modelos de Deep Learning são treinados para detectar sarcasmo em títulos de notícias. Os modelos utilizados incluem redes neurais recorrentes (LSTM) e o modelo BERT, ambos capazes de capturar o contexto e as nuances presentes na linguagem natural.

1.3 REDES NEURAIS RECORRENTES E LSTM

Redes Neurais Recorrentes (RNNs) são uma classe de redes neurais projetadas para lidar com dados sequenciais, como texto ou séries temporais. Ela funciona mantendo um estado interno que captura informações sobre elementos anteriores na sequência, permitindo que a rede aprenda dependências de longo prazo (?). No entanto, as RNNs tradicionais enfrentam desafios como o problema do desvanecimento do gradiente, que dificulta o aprendizado de dependências de longo prazo.

Para superar essas limitações, foram desenvolvidas variantes das RNNs, como as Long Short-Term Memory (LSTM) (?). As LSTMs introduzem uma arquitetura de célula especial que inclui portas de entrada, esquecimento e saída, permitindo que a rede controle o fluxo de informações e mantenha informações relevantes por períodos mais longos. Isso é particularmente útil para tarefas de processamento de linguagem natural, onde o contexto e a ordem das palavras são importantes para a compreensão do significado.

O modelo LSTM funciona da seguinte maneira: a porta de entrada decide quais informações da entrada atual devem ser armazenadas na célula de memória, a porta de esquecimento determina quais informações devem ser descartadas da célula de memória, e a porta de saída controla quais informações da célula de memória devem ser usadas para gerar a saída da rede. Essa arquitetura permite que as LSTMs aprendam dependências de longo prazo de forma mais eficaz do que as RNNs tradicionais. Isso porque as LSTMs são capazes de manter informações relevantes na célula de memória por períodos mais longos, enquanto as RNNs tradicionais tendem a esquecer informações importantes devido ao problema do desvanecimento do gradiente.

Visto que este trabalho visa detectar sarcasmo em títulos de notícias, é de suma importância capturar o contexto e as nuances presentes na linguagem e compreender como as palavras se relacionam entre si. Logo, para melhor capturar essas relações, foi utilizado o

BiLSTM (Bidirectional LSTM). O BiLSTM é uma Bidirectional Recurrent Neural Network (BRNN) (?), isto é, uma rede neural recorrente que processa a sequência de dados em ambas as direções, para frente e para trás. Isso permite que o modelo tenha acesso a informações contextuais tanto do passado quanto do futuro, melhorando sua capacidade de compreender o significado das palavras em um determinado contexto.

O exemplo a seguir ilustra o funcionamento do BiLSTM. Considere a frase "Ele não gostou do filme porque era muito longo". Ao processar essa frase, o LSTM unidirecional pode ter dificuldade em entender o significado da palavra "porque", pois ela depende do contexto fornecido pelas palavras que a seguem. No entanto, o BiLSTM pode capturar essa dependência, pois processa a frase em ambas as direções, permitindo que o modelo compreenda o significado completo da frase. Como este trabalho envolve a detecção de sarcasmo em títulos de notícias, o uso do BiLSTM é particularmente vantajoso, pois o sarcasmo muitas vezes depende do contexto e das nuances presentes na linguagem. Ao utilizar o BiLSTM, o modelo pode capturar melhor essas relações e melhorar sua capacidade de detectar sarcasmo.

O modelo base de BiLSTM utilizado neste trabalho (?) é composto por uma camada de embeddings, que converte as palavras em vetores densos de tamanho fixo. Para isto, cada headline foi tokenizado e convertida em uma sequência de índices inteiros. Cada token é mapeado para um vetor de embeddings, utilizando o FastText (?) como embeddings pré-treinados. Os embeddings são necessários para que o modelo possa trabalhar com representações numéricas das palavras, permitindo que ele aprenda padrões e relações entre as palavras.

O FastText é um modelo de aprendizado de máquina desenvolvido pelo Facebook AI Research (FAIR) para gerar representações vetoriais de palavras (word embeddings) de maneira eficiente. Diferente de métodos tradicionais como Word2Vec, o FastText representa cada palavra como um conjunto de subpalavras (n-grams de caracteres). Isso permite que o modelo capture informações morfológicas e produza vetores mais robustos, especialmente para línguas ricas em flexão ou palavras raras. Além disso, o FastText disponibiliza (?) vetores pré-treinados em grandes corpora, como Wikipedia e Common Crawl, cobrindo mais de 150 idiomas. Esses embeddings podem ser reutilizados diretamente em tarefas de PLN, reduzindo o custo de treinamento e melhorando a qualidade do modelo em cenários com poucos dados.

Feitos os embeddings, as sequências de vetores são alimentadas em uma camada BiLSTM, que processa a sequência em ambas as direções (para frente e para trás). As saídas das duas direções são concatenadas, fornecendo uma representação rica do contexto de cada palavra na sequência. Em seguida, as saídas processadas passam por uma camada de Global Max Pooling que extrai as características mais relevantes e fortes ao longo de todas as etapas da sequência, condensando-as em um vetor de tamanho fixo. A representação resultante passa por camadas Dense, que interpretam, refinam e comprimem os recursos aprendidos. Para melhorar a generalização e reduzir o overfitting, camadas de Dropout desativam neurônios aleatoriamente durante o treinamento. Por fim, uma camada de saída Sigmoid produz uma probabilidade entre

0 e 1, permitindo que o modelo realize uma classificação binária, no caso, sarcasmo ou não sarcasmo.

1.4 BERT

O BERT (Bidirectional Encoder Representations from Transformers) (?) é um modelo de linguagem baseado na arquitetura Transformer (?), que foi pré-treinado em uma grande quantidade de dados textuais. Uma arquitetura Transformer é composta por camadas de atenção, que são camadas que permitem que o modelo foque em diferentes partes da entrada ao processar uma sequência, e camadas feed-forward, que são camadas totalmente conectadas que processam as representações intermediárias geradas pelas camadas de atenção. Essa arquitetura permite que o modelo capture o contexto bidirecional das palavras, o que o torna muito eficaz para tarefas de processamento de linguagem natural. O BERT consiste em camadas empilhadas de codificadores Transformer (Transformer Encoders), que são responsáveis por processar a entrada e gerar representações contextuais das palavras.

O BERT é um modelo baseado em Transformers pré-treinado em uma grande quantidade de dados textuais utilizando duas tarefas principais: Masked Language Modeling (MLM) e Next Sentence Prediction (NSP). No MLM, o modelo é treinado para prever palavras mascaradas em uma sequência de texto, isto é, algumas palavras são substituídas por um token especial [MASK], e o modelo deve prever quais palavras foram mascaradas com base no contexto fornecido pelas palavras restantes. No NSP, o modelo é treinado para prever se uma sentença B segue uma sentença A na sequência original do texto. Essas tarefas permitem que o BERT aprenda representações ricas das palavras e do contexto em que elas aparecem, o que o torna muito eficaz para uma ampla variedade de tarefas de processamento de linguagem natural.

O BERT utilizado neste trabalho é o BERTimbau (?), que é uma versão do BERT treinada em português. O BERTimbau foi treinado usando dados de BrWaC, uma grande coleção de textos em português da web. Como resultado, o BERTimbau é capaz de capturar as nuances e particularidades da língua portuguesa, o que o torna especialmente adequado para tarefas de processamento de linguagem natural nessa língua.

A função base do BERT é gerar representações contextuais das palavras em uma sequência de texto. Essas representações são vetores densos que capturam o significado e o contexto das palavras, levando em consideração as palavras que as cercam. Essas representações podem ser utilizadas como entradas para outros modelos de aprendizado de máquina, como redes neurais, para realizar tarefas específicas de processamento de linguagem natural, função essa que foi utilizada neste trabalho e descrita nos capítulos seguintes.

1.5 BASE DE DADOS

A base de dados utilizada neste trabalho foi criada por (?) e é composta por títulos de notícias extraídos do site TheOnion.com¹, conhecido por seu conteúdo satírico e sarcástico, e do site Huffington Post², que apresenta notícias sérias e informativas. Esta base de dados está disponível publicamente e pode ser acessada através do repositório no GitHub³. O propósito do site The Onion é entreter os leitores com notícias fictícias que utilizam o sarcasmo como principal recurso humorístico, isto prove um ambiente ideal para coletar exemplos de sarcasmo em títulos de notícias.

Muitos estudos anteriores, que serão referenciados logo mais, sobre detecção de sarcasmo utilizavam bases de dados compostas por tweets, que são mensagens curtas postadas na plataforma Twitter. No entanto, tweets podem conter abreviações, gírias, erros gramaticais e outros elementos que dificultam a análise linguística. Além disso, muitas dessas bases foram coletadas de forma automática, utilizando hashtags como #sarcasm para identificar mensagens sarcásticas, porém isso de nada garante que todas as mensagens com essa hashtag sejam realmente sarcásticas, ou que todas as mensagens sarcásticas estejam marcadas com essa hashtag. Por outro lado, títulos de notícias são geralmente mais formais e estruturados, o que facilita a análise linguística e a detecção de sarcasmo. Portanto, a utilização de títulos de notícias como base de dados para detecção de sarcasmo pode proporcionar resultados mais precisos e confiáveis.

A base de dados contém 28.503 títulos de notícias, sendo 14.951 títulos sarcásticos do The Onion e 13.552 títulos não sarcásticos do Huffington Post, ou seja, a base de dados está balanceada entre as duas classes. Cada título é acompanhado por um rótulo binário, onde 1 indica que o título é sarcástico e 0 indica que o título não é sarcástico. A seguir, são apresentados alguns exemplos de títulos sarcásticos e não sarcásticos presentes na base de dados:

- Título sarcástico: "ford develops new suv that runs purely on gasoline"
- Título não sarcástico: "what to know regarding current treatments for ebola"
- Título sarcástico: "mother comes pretty close to using word 'streaming' correctly"
- Título não sarcástico: "5 ways to file your taxes with less stress"

Esses exemplos ilustram a diferença entre títulos sarcásticos, que utilizam humor e ironia para transmitir uma mensagem, e títulos não sarcásticos, que são informativos e diretos.

Este trabalho é focado na detecção de sarcasmo em títulos de notícias em português. Portanto, a base de dados original, que está em inglês, foi traduzida para o português utilizando a API do ChatGPT⁴. O modelo escolhido para a tradução foi o GPT-4.1 Mini, que é capaz de

¹<https://www.theonion.com/>

²<https://www.huffpost.com/>

³https://raw.githubusercontent.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection/refs/heads/master/Sarcasm_Headlines_Dataset.json

⁴<https://openai.com/blog/chatgpt>

realizar traduções de boa qualidade, entretanto, é importante ressaltar que a tradução automática pode introduzir erros ou nuances que não estavam presentes no texto original. Todos os cuidados foram tomados para garantir que a tradução mantivesse o significado e o contexto dos títulos originais. O prompt utilizado para a tradução foi elaborado para solicitar uma tradução literal, sem tentar interpretar o sarcasmo, de forma a preservar o sentido original dos títulos. O prompt utilizado foi o seguinte:

"You are a professional translator. Translate the following English headlines into Brazilian Portuguese. Rules: 1. Preserve the original meaning, tone, and style. 2. If the sentence contains sarcasm, irony, exaggeration, or humor, keep it intact in Portuguese. 3. Do not explain or interpret sarcasm — just translate it naturally. 4. Keep proper names, places, and organizations unchanged. 5. Return only the translations, numbered correspondingly."

Além disso, no momento da criação, o modelo foi configurado com uma temperatura de 0.3, o que torna o modelo mais conservador, reduzindo a criatividade na resposta.

```
response = client.chat.completions.create(
    model=model,
    messages=[ {"role": "system", "content": "You are a professional transla-
              {"role": "user", "content": prompt} },
    temperature=0.3
)
```

1.6 MÉTRICAS DE AVALIAÇÃO

Métricas de avaliação são essenciais para medir o desempenho dos modelos de aprendizado de máquina. Elas fornecem uma maneira objetiva de comparar diferentes modelos e técnicas, permitindo identificar quais abordagens são mais eficazes para uma determinada tarefa. Neste trabalho, serão utilizadas as seguintes métricas para avaliar o desempenho dos modelos propostos: acurácia, precisão, recall e F1-score, sendo esta última a métrica principal para comparação entre os modelos.

A acurácia é uma métrica que indica a proporção de previsões corretas em relação ao total de previsões feitas onde TP é o número de verdadeiros positivos, TN é o número de verdadeiros negativos, FP é o número de falsos positivos e FN é o número de falsos negativos. A acurácia é uma métrica útil quando as classes estão balanceadas, mas pode ser enganosa em casos de classes desbalanceadas.

$$Acuracia = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.1)$$

A precisão é a proporção de verdadeiros positivos em relação ao total de previsões positivas feitas pelo modelo. A precisão é importante quando o custo de falsos positivos é alto.

$$Preciso = \frac{TP}{TP + FP} \quad (1.2)$$

O recall, também conhecido como sensibilidade, é a proporção de verdadeiros positivos em relação ao total de casos positivos reais, isto é, em relação ao total de exemplos que realmente pertencem à classe positiva, ele é crucial quando o custo de falsos negativos é alto.

$$Recall = \frac{TP}{TP + FN} \quad (1.3)$$

O F1-score é a média harmônica entre precisão e recall, fornecendo uma métrica balanceada que leva em conta ambos. O F1-score é especialmente útil quando há um trade-off, ou seja, quando aumentar a precisão pode diminuir o recall e vice-versa. É desejável uma métrica que considere ambos os aspectos.

$$F1 = 2 \times \frac{Preciso \times Recall}{Preciso + Recall} \quad (1.4)$$

Esses métricas serão utilizadas para avaliar o desempenho dos modelos propostos na detecção de sarcasmo em títulos de notícias em português. Os resultados serão apresentados e discutidos no capítulo sobre avaliação experimental.

2 MODELO PROPOSTO

Neste capítulo, será apresentado o modelo proposto para a detecção de sarcasmo em títulos de notícias em português. O modelo combina a arquitetura BiLSTM com o BERT como extrator de características, visando aproveitar o poder do BERT para capturar o contexto e as nuances da linguagem, enquanto o BiLSTM é responsável por aprender as relações temporais e sequenciais dos dados.

2.1 FERRAMENTAS E BIBLIOTECAS

Os experimentos realizados neste trabalho foram feitos no Google Colab, que é uma plataforma de computação em nuvem que permite a execução de código Python em notebooks Jupyter. O Google Colab oferece acesso a GPUs, o que é essencial para o treinamento de modelos de deep learning, como o LSTM e o BERT. Além disso, o Google Colab oferece uma série de bibliotecas pré-instaladas, como TensorFlow, Keras e Transformers, que facilitam o desenvolvimento e a implementação dos modelos. Como ambiente de desenvolvimento, foi utilizado o Jupyter Notebook, que é uma ferramenta interativa que permite a criação e compartilhamento de documentos que contêm código, visualizações e texto explicativo. O Jupyter Notebook é amplamente utilizado na comunidade de ciência de dados e machine learning devido à sua facilidade de uso e flexibilidade. Para a implementação dos modelos LSTM e BERT, foram utilizadas as bibliotecas TensorFlow e Keras.

O TensorFlow é uma biblioteca de código aberto desenvolvida pelo Google para computação numérica e machine learning. Ele oferece uma ampla gama de ferramentas e recursos para a construção, treinamento e implantação de modelos de machine learning, incluindo suporte para redes neurais profundas e computação distribuída. O TensorFlow é amplamente utilizado na comunidade de machine learning devido à sua flexibilidade, escalabilidade e desempenho. Já o Keras é uma biblioteca de alto nível construída sobre o TensorFlow, que fornece uma interface simples e intuitiva para a construção e treinamento de modelos de deep learning. O Keras oferece uma série de camadas pré-construídas, otimizadores e funções de perda, o que facilita o desenvolvimento de modelos complexos.

Além disso, para a utilização do modelo BERT, foi utilizada a biblioteca Transformers, desenvolvida pela Hugging Face. A biblioteca Transformers oferece uma ampla gama de modelos pré-treinados baseados na arquitetura Transformer, incluindo o BERT, GPT e RoBERTa. Ela fornece uma interface simples para o carregamento e utilização desses modelos, bem como ferramentas para o fine-tuning e avaliação. A biblioteca Transformers é amplamente utilizada na comunidade de processamento de linguagem natural devido à sua facilidade de uso e à qualidade dos modelos pré-treinados disponíveis. Como otimizador, foi utilizado o Adam (Adaptive

Moment Estimation), que é um dos otimizadores mais populares e eficazes para o treinamento de redes neurais profundas.

Além disso, foi também utilizado o Pandas e o NumPy. O Pandas é uma biblioteca de código aberto que fornece estruturas de dados e ferramentas de análise de dados para a linguagem Python. Ele é amplamente utilizado na comunidade de ciência de dados devido à sua facilidade de uso e eficiência no manuseio de grandes conjuntos de dados. O NumPy é uma biblioteca fundamental para computação científica em Python, fornecendo suporte para arrays multidimensionais e uma ampla gama de funções matemáticas e estatísticas. Ele é amplamente utilizado na comunidade de ciência de dados e machine learning devido à sua eficiência e desempenho em operações numéricas.

A seguir as versões das principais bibliotecas utilizadas neste trabalho:

- TensorFlow: 2.12.0
- Pandas: 1.5.3
- NumPy: 1.24.2
- Scikit-learn: 1.2.2
- Transformers: 4.30.2

2.2 BASE DE DADOS

A base de dados utilizada neste trabalho é composta por títulos de notícias sarcásticas e não sarcásticas, conforme descrito no Capítulo ???. A base de dados então dividida em conjuntos de treino, validação e teste, com proporções de 70%, 10% e 20%, respectivamente. Essa divisão permite avaliar o desempenho do modelo de forma adequada, garantindo que o modelo seja treinado em uma parte dos dados e avaliado em dados não vistos durante o treinamento. Para a divisão dos dados, foi utilizado a função `train_test_split` do Scikit-learn. A divisão foi feita de forma estratificada, garantindo que a proporção de títulos sarcásticos e não sarcásticos seja mantida em cada conjunto, além de definir um valor fixo para o parâmetro `random_state`, garantindo a reproduzibilidade dos resultados.

2.3 PRÉ-PROCESSAMENTO DOS DADOS

O pré-processamento dos dados é uma etapa fundamental para garantir a qualidade e eficácia dos modelos utilizados na classificação. Neste trabalho, foi feito uma tradução dos títulos para o português, conforme mencionado anteriormente. Para isso, foi utilizada a API do ChatGPT, que é capaz de realizar traduções de boa qualidade. Essa etapa de pré-processamento é essencial para garantir que os dados estejam na língua correta para o desenvolvimento dos modelos propostos.

2.4 MODELO BASE

Após a preparação dos dados, foi implementado o modelo BiLSTM. Primeiramente, os dados foram divididos em conjuntos de treino, validação e teste, conforme descrito anteriormente. Para avaliar o desempenho do modelo, a base de teste foi utilizada e os resultados são apresentados no Capítulo ???. O modelo BiLSTM foi implementado utilizando a API Sequencial do Keras. A arquitetura do modelo é composta por uma camada de embedding, seguida por uma camada BiLSTM com 128 unidades, seguida de uma operação de Global Max Pooling, seguidas por 2 camadas densas com funções de ativação ReLU, intercaladas com camadas de dropout para evitar overfitting. Finalmente, a camada de saída é uma camada densa com uma unidade e função de ativação sigmóide, que produz uma saída entre 0 e 1, indicando a probabilidade da manchete ser sarcástica.

A função de perda utilizada foi a binary crossentropy, adequada para problemas de classificação binária. O otimizador utilizado foi o Adam, com uma taxa de aprendizado de 0.00001. O Adam é um dos otimizadores mais populares e eficazes para o treinamento de redes neurais profundas. Um otimizador é um algoritmo que ajusta os pesos do modelo durante o treinamento, com o objetivo de minimizar a função de perda.

O modelo foi treinado por um máximo de 25 épocas, com tamanho de lote (batch size) de 32. Durante o treinamento, foi utilizado o Early Stopping, que monitora a métrica de validação (val_loss) e interrompe o treinamento se a métrica não melhorar por 3 épocas consecutivas, prevenindo overfitting.

2.5 MODELO PROPOSTO: BERT-BiLSTM

Além do modelo BiLSTM, foi implementado o modelo BERT-BiLSTM, que utiliza o modelo pré-treinado BERT como camada de embedding. Como descrito anteriormente, o BERT é um modelo de linguagem baseado na arquitetura Transformer, que foi pré-treinado em uma grande quantidade de dados textuais. O BERT é capaz de capturar o contexto bidirecional das palavras, o que o torna muito eficaz para tarefas de processamento de linguagem natural.

Primeiramente, os títulos foram tokenizados utilizando o tokenizador do BERTimbau, que converte as palavras em tokens compatíveis com o modelo. Esses tokens foram então alimentados no modelo, que gerou representações vetoriais para cada título. Essas representações foram então utilizadas como entradas para o modelo BiLSTM, que realizou a classificação dos títulos em sarcásticos ou não sarcásticos. A arquitetura do modelo BERT-BiLSTM é semelhante à do modelo BiLSTM, com a diferença de que a camada de embedding é substituída pelo modelo BERT pré-treinado. A função de perda, otimizador, tamanho de lote e estratégia de Early Stopping utilizados foram os mesmos do modelo BiLSTM.

Espera-se que a incorporação do BERT como camada de embedding traga melhorias no desempenho do modelo, especialmente na captura do contexto e das nuances da linguagem presentes nos títulos de notícias.

3 AVALIAÇÃO EXPERIMENTAL

Neste capítulo, serão apresentados os experimentos realizados para avaliar o desempenho do modelo proposto na detecção de sarcasmo em títulos de notícias em português. Serão descritos os dados utilizados, as métricas de avaliação adotadas, os resultados obtidos e uma análise comparativa com outros modelos existentes na literatura.

3.1 MÉTRICAS DE AVALIAÇÃO

Como descrito no Capítulo 2, para avaliar o desempenho do modelo proposto, foram utilizadas as seguintes métricas:

- Acurácia: Mede a proporção de previsões corretas em relação ao total de previsões realizadas.
- Precisão: Mede a proporção de verdadeiros positivos em relação ao total de positivos previstos pelo modelo.
- Recall: Mede a proporção de verdadeiros positivos em relação ao total de positivos reais no conjunto de dados.
- F1-Score: É a média harmônica entre precisão e recall, fornecendo uma medida balanceada do desempenho do modelo.

Essas métricas foram escolhidas devido à sua capacidade de fornecer uma visão abrangente do desempenho do modelo, considerando tanto a capacidade de identificar corretamente os exemplos positivos quanto a capacidade de evitar falsos positivos. A principal métrica utilizada para comparar o desempenho dos modelos foi o F1-Score, pois ele leva em conta tanto a precisão quanto o recall, sendo especialmente útil em situações onde há um desequilíbrio entre as classes. Um F1-Score mais alto indica tanto uma alta precisão quanto um alto recall, refletindo um desempenho geral melhor do modelo na tarefa de detecção de sarcasmo.

3.2 RESULTADOS OBTIDOS

Nesta seção, serão apresentados os resultados obtidos pelos experimentos realizados com o modelo proposto, bem como uma comparação com outros modelos existentes na literatura. Os resultados serão apresentados em termos das métricas de avaliação descritas anteriormente. A Tabela ?? apresenta um resumo dos resultados obtidos pelos diferentes modelos na tarefa de detecção de sarcasmo.

Os resultados apresentados na Tabela ?? indicam que o modelo proposto, BERT + BiLSTM, superou o modelo BiLSTM simples em todas as métricas avaliadas. A incorporação

Tabela 3.1: Resultados dos Experimentos de Detecção de Sarcasmo

	Base de Dados	Acurácia	Precisão	Recall	F1-Score
BiLSTM	Headlines	82%	79%	87%	80%
BERT + BiLSTM	Headlines	•	82%	84%	86%
MHA-BiLSTM (?)	Headlines	•	72%	83%	77%
BiLSTM (?)	Headlines	86%	84%	86%	85%
BERT+BiLSTM (?)	Headlines	89%	89%	89%	89%
MTL-BiLSTM (?)	Twitter	•	93%	93%	93%
CNN-LR (?)	Twitter	94%	95%	94%	94%
BERT (?)	Twitter	•	74%	74%	74%
BERT (?)	Reddit	74%	74%	76%	77%
RoBERTa (?)	Reddit	76%	76%	78%	77%
RoBERTa+Bi-LSTM+MHA (?)	Headlines	93%	93%	93%	93%

do BERT como extrator de características permitiu capturar melhor o contexto e as nuances da linguagem. No entanto, o modelo base foi superado por pouco, indicando que apesar das melhorias trazidas pelo BERT, o BiLSTM ainda é um modelo competitivo para a tarefa de detecção de sarcasmo.

Em comparação com outros modelos existentes na literatura, o modelo proposto apresentou um desempenho competitivo, embora alguns modelos, como o MTL-BiLSTM (?) e o CNN-LR (?), tenham alcançado métricas superiores. É importante notar que nenhum dos modelos comparados foi testado na mesma base de dados, os trabalhos acima utilizaram uma abordagem para detecção de sarcasmo em textos em inglês, enquanto o presente trabalho focou em títulos de notícias em português. A comparação direta dos resultados deve ser feita com cautela, considerando as diferenças nas bases de dados e nas abordagens utilizadas.

Os resultados obtidos indicam que o modelo proposto é eficaz na detecção de sarcasmo em títulos de notícias em português, mas há espaço para melhorias. Futuras pesquisas podem explorar a incorporação de outras técnicas de pré-processamento, arquiteturas de modelos mais complexas e a utilização de bases de dados maiores e mais diversificadas para aprimorar ainda mais o desempenho na tarefa de detecção de sarcasmo.

REFERÊNCIAS

- Baruah, A., Das, K., Barbhuiya, F. e Dey, K. (2020). Context-aware sarcasm detection using BERT. Em Klebanov, B. B., Shutova, E., Lichtenstein, P., Muresan, S., Wee, C., Feldman, A. e Ghosh, D., editores, *Proceedings of the Second Workshop on Figurative Language Processing*, páginas 83–87, Online. Association for Computational Linguistics.
- de Medeiros Caseli, H. e das Graças Volpe Nunes, M. (2023). *Processamento de Linguagem Natural*. Brasileiras em PLN.
- de Souza, F. C. (2020). Bertimbau = pretrained bert models for brazilian portuguese. Dissertação de Mestrado, Mestrado em Ciência da Computação - Universidade Federal de Campinas, Campinas - SP.
- Devlin, J., Chang, M., Lee, K. e Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. páginas 4171–4186.
- Grave, E., Bojanowski, P., Gupta, P., Joulin, A. e Mikolov, T. (2018). Learning word vectors for 157 languages. Em *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.
- Hochreiter, S. e Kepler, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Joulin, A., Grave, E., Bojanowski, P. e Mikolov, T. (2017). Bag of tricks for efficient text classification. 2:427–431.
- Khan, A., Majumdar, D. e Mondal, B. (2025). A hybrid transformer based model for sarcasm detection from news headlines. *Journal of Intelligent Information Systems*, 63:1339–1359.
- Kumar, A., Narapareddy, V. T., Aditya Srikanth, V., Malapati, A. e Neti, L. B. M. (2020). Sarcasm detection using multi-head attention based bidirectional lstm. *IEEE Access*, 8:6388–6397.
- Misra, R. e Arora, P. (2023). Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Nayak, D. e Bolla, B. (2022). *Efficient Deep Learning Methods for Sarcasm Detection of News Headlines*, páginas 371–382.
- Razali, M. S., Halin, A. A., Ye, L., Doraisamy, S. e Norowi, N. M. (2021). Sarcasm detection using deep learning with contextual features. *IEEE Access*, 9:68609–68618.

- Schuster, M. e Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Scola, E. e Segura-Bedmar, I. (2021). Sarcasm detection with bert. *Procesamiento del Lenguaje Natural*, 67:13–25.
- Shrikhande, P., Setty, V. e Sahani, D. A. (2020). Sarcasm detection in newspaper headlines. Em *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, páginas 483–487.
- Tan, Y., Chow, C. O., Kanesan, J., Chuah, J. H. e Lim, Y. (2023). Sentiment analysis and sarcasm detection using deep multi-task learning. *Wireless Personal Communications*, 129:2213–2237.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. e Polosukhin, I. (2017). Attention is all you need. páginas 6000–6010.

APÊNDICE A – EXEMPLO DE ANEXO

Os apêndices são uma extensão do texto, destacados deste para evitar descontinuidade na sequência lógica ou alongamento excessivo de determinado assunto ou tópico secundário dentro dos capítulos da dissertação ou da tese. São contribuições que servem para esclarecer, complementar, provar ou confirmar as ideias apresentadas no texto dos capítulos e que são importantes para a compreensão dos mesmos.

Todos os apêndices devem vir após as referências bibliográficas e devem ser enumerados por letras maiúsculas (A, B, C, ...).

A.1 UMA SEÇÃO

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetuer quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit.

Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

A.1.1 Uma Subseção

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.

A.1.1.1 Uma Subsubseção

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper.

Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.

APÊNDICE B – TESTES DE ALINHAMENTO DE LISTAS

B.1 OUTRA SEÇÃO

Exemplo de lista simples com dois níveis: Exemplo de lista simples com dois níveis:
Exemplo de lista simples com dois níveis: Exemplo de lista simples com dois níveis: Exemplo
de lista simples com dois níveis: Exemplo de lista simples com dois níveis: Exemplo de lista
simples com dois níveis: Exemplo de lista simples com dois níveis: Exemplo de lista simples
com dois níveis.

- (a) Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra,
Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra, Caturra,
Caturra.

(b) da Terra, da
Terra, da Terra, da Terra, da Terra, da Terra, da Terra, da Terra, da Terra, da Terra,
da Terra.

2. Laranja, Laranja, Laranja, Laranja, Laranja, Laranja, Laranja, Laranja, Laranja,
Laranja, Laranja, Laranja, Laranja, Laranja, Laranja.

(a) Bahia,
Bahia, Bahia, Bahia, Bahia, Bahia.

(b) Lima,
Lima, Lima, Lima, Lima, Lima, Lima, Lima, Lima, Lima, Lima, Lima, Lima, Lima,

Exemplo de lista descritiva com dois níveis: Exemplo de lista descritiva com dois níveis:
Exemplo de lista descritiva com dois níveis: Exemplo de lista descritiva com dois níveis: Exemplo
de lista descritiva com dois níveis: Exemplo de lista descritiva com dois níveis: Exemplo de lista
descritiva com dois níveis: Exemplo de lista descritiva com dois níveis: Exemplo de lista
descritiva com dois níveis.

Laranja : Laranja, Laranja, Laranja, Laranja, Laranja, Laranja, Laranja, Laranja, Laranja,
Laranja, Laranja, Laranja, Laranja, Laranja, Laranja.