

Detecção de sarcamos e ambiguidades textuais com enriquecimento de prompts para transformação linguística

Beatriz P. Santos, Carlos H. F. Marques, Luis F. C. Souza, Paula C. Silva, Pedro H. A. A. Silva

Departamento de Computação (DComp)

Universidade Federal de São Carlos (UFSCar)

18052-780, Sorocaba, São Paulo, Brasil

beatrizpatricio@estudante.ufscar.br carlos.franca@estudante.ufscar.br

luisfelipicruzsouza@estudante.ufscar.br paulacaires@estudante.ufscar.br

pedrohaas@estudante.ufscar.br

Abstract—A interpretação de sarcamos e ambiguidades em textos representa um desafio significativo para leitores em geral. Essa dificuldade é ainda mais acentuada em pessoas com Transtorno do Espectro Autista (TEA), que frequentemente apresentam limitações na compreensão de linguagem figurada, o que pode comprometer sua inclusão comunicativa. Este projeto propõe uma solução baseada em Processamento de Linguagem Natural (PLN) para detectar sarcamos e possíveis ambiguidades lexicais em textos e reescrevê-los de forma mais clara e objetiva. Para esta finalidade, a abordagem utiliza um conjunto de dados de notícias classificadas em irônicas ou não, um léxico, modelos Transformers e modelos de linguagem de grande escala (LLMs).

Keywords—sarcasmo; ambiguidade; acessibilidade; transformers; PLN; TEA;

I. INTRODUÇÃO

O autismo é um transtorno que afeta o núcleo de socialização, incluindo a compreensão de significados e uso da linguagem [16]. Estudos mostram que indivíduos autistas apresentam as dificuldades:

- 1) Dificuldade de comunicação em geral;
- 2) Dificuldade com a linguagem pragmática, que se refere ao uso da linguagem em situações sociais, como entender sarcasmos e metáforas. De acordo com o artigo "Compreensão de ambiguidade em crianças com Transtorno Específico de Linguagem e Fala e Transtorno do Espectro Autista" [14]:

"Pesquisas realizadas confirmam que autistas de alto funcionamento falham na interpretação de estímulos metafóricos, expressões idiomáticas, comentários humorísticos ou sarcásticos, além de não compreender dicas não-verbais no desenrolar de uma conversação."

- 3) Dificuldades com a linguagem estrutural, que se refere a forma e significado de palavras e frases;
- 4) Dificuldades de interpretabilidade;

A dificuldade de comunicação de pessoas autistas está enraizada nos desafios de desenvolvimento social e afetivo [16], sendo uma questão central nesse transtorno.

Sabendo disso, o objetivo do presente trabalho é elaborar uma tarefa de Processamento de Linguagem Natural (PLN) com foco em inclusão, principalmente nas dificuldades de compreensão de sarcasmo e ambiguidade. Adicionalmente, considerando que pessoas autistas podem levar mais tempo para processar a linguagem, será feita a reescrita de frases para tornar a linguagem mais adequada e inclusiva.

II. TAREFA

A tarefa de Processamento de Linguagem Natural (PLN) proposta tem como objetivo principal a detecção e eliminação de ironias e ambiguidades lexicais em textos, visando torná-los mais claros e objetivos. Para alcançar esse objetivo, serão aplicadas 3 etapas. A primeira, detectando frases possivelmente irônicas no texto. A segunda, detectando o sentido de palavras possivelmente ambíguas no texto. A terceira e última etapa consiste em utilizar as informações obtidas pelas duas etapas anteriores para enriquecer um *prompt* para reescrever o texto.

A. Dataset

Para a etapa da detecção de sarcasmo, foi utilizado um corpus que contém dados de notícias de 3 sites de notícias [5] classificados como sarcasmos ou não. Este corpus inclui os seguintes atributos para cada notícia: título da notícia (texto), corpo da notícia (texto), *link* para a notícia (texto) e uma categoria que indica se o conteúdo é sarcástico ou não (booleano). A tabela a seguir indica o tamanho do corpus, mostrando também a quantidade de notícias categorizadas de cada *site*.

Arquivo	Registros
estadao.json	11.272
sensacionalista.json	5.006
the_piaui_herald.json	2.216
Total	18.494

Table I
TAMANHO DOS ARQUIVOS DO CORPUS DE NOTÍCIAS [5]

Além disso, para a avaliação da etapa de seleção de sentido em palavras possivelmente ambíguas, foi necessário a criação de um corpus específico de maneira semi automática (utilizando LLMs e revisão humana). Cada registro do corpus continha uma frase, qual a palavra ambígua presente e os seus sentidos no contexto da frase.

B. Medidas de avaliação

Para a detecção de sarcasmo e sentidos de possíveis ambiguidades léxicas, foram utilizadas métricas clássicas de classificação: acurácia, precisão, recall e F1-score, permitindo avaliar o desempenho tanto em termos de acertos gerais quanto na capacidade de identificar corretamente os casos positivos.

Já para a etapa de transformação de texto com remoção de sarcasmo e ambiguidade lexical, foi feita uma avaliação baseada em similaridade do cosseno, para avaliar se o significado dos textos foi mantido após a reescrita e também foi realizada uma avaliação manual. Nessa fase, nosso grupo analisou os textos gerados com base nos critérios de clareza, fidelidade ao conteúdo original e efetividade na remoção de elementos ambíguos ou irônicos. Essa avaliação utilizou a escala Likert, variando de 1 a 7, onde 1 representa a menor concordância com o critério avaliado e 7 representa a maior concordância. Complementando a avaliação manual e a similaridade de cosseno, foi utilizada a pontuação de Flesch [11] adaptada para o português para verificar se os textos ficaram mais "claros" e "simples".

III. ESTRATÉGIA

A. Ambiguidade

1) *Recursos em língua portuguesa usados:* Para consultar os sentidos das palavras, foi utilizado o léxico OpenWordnet-PT, que é a versão em português do léxico semântico de Princeton (Wordnet) [13].

Também, para avaliar o desempenho do processo de seleção de sentido, foi criado um corpus com cerca de 500 registros de maneira semi-automática. Durante esse processo, identificamos a necessidade de atribuir mais de um sentido por palavra em certos contextos, pois o léxico apresentava definições parecidas (algumas mais específicas e outras mais gerais) que se aplicavam à mesma situação. Dessa forma, o corpus foi anotado considerando múltiplos sentidos quando ambos se mostravam coerentes com o uso da palavra na frase. Portanto, cada registro possui uma frase, uma palavra alvo, e os sentidos da palavra alvo naquela frase (seguindo os sentidos da OpenWordnet-PT).

Para a abordagem mais avançada, usando transformers, utilizamos um modelo BERT pré-treinado para português (neuralmind/bert-base-portuguese-cased) para gerar embeddings contextualizadas das palavras e dos sentidos.

2) *Abordagem com algoritmo de Lesk - Etapa de pré-processamento:* Para este algoritmo, além do pré-processamento básico que remove pontuações e converte para minúsculo, identificamos a necessidade de remover stopwords, pois o algoritmo de Lesk calcula a sobreposição entre as palavras do contexto e das definições. Como as stopwords são termos muito frequentes e pouco informativos, elas aumentam erroneamente o overlap e podem levar à escolha de sentidos incorretos.

3) *Abordagem com algoritmo de Lesk - Etapa de processamento:* Primeiramente, os possíveis sentidos da palavra alvo são consultados no léxico OpenWordnet-PT. Caso tenha mais de um sentido possível, a palavra possivelmente é ambígua.

Em seguida, cada possível sentido é comparado com as palavras do contexto, no caso, as outras palavras da frase que não a alvo. O sentido mais provável é então aquele cuja definição possui o maior número de palavras em comum com o contexto.

4) *Abordagem com transformers - Etapa de pré-processamento:* As frases de entrada passam por um pré-processamento básico, que inclui a conversão para minúsculas e a remoção de pontuações. Isso ajuda a padronizar o texto para a geração de embeddings.

5) *Abordagem com transformers - Etapa de processamento:* Primeiramente, os possíveis sentidos da palavra alvo são consultados no léxico OpenWordnet-PT. Caso tenha mais de um sentido possível, a palavra possivelmente é ambígua.

Em seguida, o modelo BERT pré-treinado é usado para gerar os embeddings tanto da palavra alvo da frase quanto dos possíveis sentidos que foram consultados no léxico. Para a palavra alvo, os embeddings são dados pela média dos embeddings dos tokens correspondentes a palavra. Já para os embeddings dos sentidos, é utilizado uma ponderação entre os embeddings do token CLS, e a média dos outros embeddings que não são do token CLS. Os pesos que obtiveram os melhores resultados foram 80% para o token CLS, e 20% para a média dos embeddings dos outros tokens. Cabe ressaltar que para a geração dos embeddings, foram testadas diversas estratégias, sendo uma delas sugerida no livro de Jurafsky e Martin (Speech and Language Processing) [12] de somar as representações das 4 últimas camadas do modelo. Porém, dentre todas as estratégias, a que obteve melhor resultado foi utilizando somente a última camada.

A partir então do embedding da palavra alvo, é calculada a similaridade do cosseno com os embeddings de cada sentido possível.

O sentido mais provável para a palavra alvo, então, é escolhido como sendo o de maior similaridade. Outras estratégias para selecionar esse sentido foram discutidas, como o uso do algoritmo de SVM e o uso de fine tuning em um modelo, porém pela carência de um corpus anotado para o português, a abordagem que consideramos mais adequada

foi a similaridade do cosseno.

Na implementação da função principal de desambiguação, optamos por aplicar a lematização somente na palavra alvo e não no restante do contexto, pois focamos em preservar a forma original das palavras do contexto para conseguirmos manter as características semânticas da frase, garantindo que os embeddings refletissem com maior fidelidade ao contexto.

6) *Dificuldades encontradas*: Podemos notar a carência de um corpus anotado para desambiguação de sentidos em português. Com um corpus como esse que contivesse bastante exemplos, seria possível testar outras estratégias para selecionar o sentido, que não a similaridade do cosseno, como o algoritmo de SVM e um fine-tuning em um modelo. Também, podemos perceber que o léxico carece de muitas definições. Por exemplo, para a palavra "banco", ele tem as definições de banco tanto como assento, quanto para instituição financeira, mas não tem a definição como banco de areia. Um outro exemplo é a palavra manga, que é ambígua mas não tem nenhuma definição no léxico. O léxico apresenta também palavras muito genéricas, portanto a frequência dessas palavras afetou principalmente o algoritmo de Lesk, resultando em alto overlap para sentidos errados.

Table II
RESULTADOS DAS ESTRATÉGIAS

Estratégia	Acurácia	Precisão (macro)	Revocação (macro)	F1-score (macro)
Algoritmo de Lesk	0.44	0.34	0.42	0.34
Transformers e cosseno	0.67	0.60	0.66	0.59

B. Sarcasmo

Para o processamento linguístico dos textos em português, foi empregado o modelo pré-treinado pt-core-news-sm, da biblioteca spaCy, utilizado especificamente na etapa de lemmatization para normalização morfológica das palavras.

Além disso, foi utilizado o modelo sentence-transformers/xlm-r-bert-base-nli-stsb-mean-tokens, da arquitetura Transformers, treinado em múltiplos idiomas, incluindo o português, para realização do fine-tuning voltado à detecção de sarcasmo.

1) *Etapas de pré-processamento*: O pré-processamento textual envolveu etapas clássicas de limpeza e padronização dos dados, incluindo: remoção de caracteres não textuais, remoção de stopwords e conversão para letras minúsculas.

Complementarmente, foi aplicada a técnica de lemmatization com o modelo pt-core-news-sm, a fim de reduzir as palavras às suas formas canônicas (lemmas), promovendo a uniformização lexical do corpus.

2) *Abordagem 1 - Classificação tradicional com Word2Vec e Random Forest*: Na primeira abordagem para detecção automática de sarcasmo em textos, foi

implementado um pipeline baseado em algoritmos clássicos de aprendizado de máquina (Machine Learning). O modelo que apresentou o melhor desempenho entre os avaliados foi o Random Forest Classifier, pertencente à classe dos algoritmos baseados em comitês de decisão.

O primeiro passo consistiu na vetorização dos textos — processo fundamental para transformar dados textuais em representações numéricas manipuláveis por modelos estatísticos. Para isso, foi adotada a técnica de Word Embedding, utilizando o algoritmo Word2Vec [18], que aprende representações vetoriais densas para palavras com base em seu contexto. Esse método é eficaz para capturar propriedades semânticas e sintáticas do idioma, projetando palavras semanticamente similares em regiões próximas no espaço vetorial.

O treinamento do Word2Vec foi realizado a partir dos tokens extraídos durante o pré-processamento dos textos. O modelo foi ajustado com parâmetros específicos (como dimensionalidade do vetor, tamanho da janela de contexto, estratégia Skip-gram, e número de épocas) e, uma vez treinado, foi utilizado para gerar vetores médios por frase, compondo assim a matriz de entrada para o classificador.

Com os embeddings obtidos, foram avaliados diferentes algoritmos de classificação supervisionada: Árvore de Decisão (Decision Tree), Random Forest, Regressão Logística (Logistic Regression), K-Nearest Neighbors (KNN) e Máquinas de Vetores de Suporte (SVM). Através da análise de métricas de classificação, o modelo Random Forest destacou-se como o mais eficaz.

Por fim, o classificador treinado foi integrado a um sistema de inferência capaz de realizar a predição automática de sarcasmo em textos.

3) *Abordagem 2 - Modelo Transformers + fine-tuning*: O fine-tuning de modelos do Sentence Transformers [?] foi adotado com o objetivo de adaptar as representações à tarefa de detecção de sarcasmo em notícias. Nesse sentido, embeddings genéricos como o Word2Vec podem não capturar de maneira satisfatória as nuances contextuais inerentes desse recurso de linguagem. Para isso, foi utilizado um modelo pré-treinado do Sentence Transformers e que foi ajustado com base no dataset de notícias. A função de perda foi escolhida levando em consideração a classificação supervisionada com rótulos discretos. Após o treinamento, o modelo foi avaliado com a acurácia de 0.92, apresentando resultado superior à abordagem Word2Vec. Após o fine-tuning, os embeddings foram utilizados como entrada para um classificador LogisticRegression.

C. Enriquecimento de Prompt e Reescrita de Texto

A etapa de reestruturação do texto teve como foco a remoção de ironias e ambiguidades para melhorar a acessibilidade comunicativa para pessoas com TEA, nesta etapa foi utilizado o modelo de LLM gemini-2.5-flash, por se tratar de um modelo gratuito.

Table III
RESULTADOS DAS ESTRATÉGIAS DE CLASSIFICAÇÃO

Estratégia	Acurácia	Precisão (macro)	Revocação (macro)	F1-score (macro)
Word2Vec + Random Forest	0.85	0.85	0.85	0.85
Sequence Transformers	0.92	0.92	0.92	0.92

Para otimizar a performance do LLM e garantir que as transformações fossem alinhadas aos objetivos do projeto, foram aplicados as seguintes técnicas de *prompt engineering*:

- 1) **Contexto do Assistente:** O *prompt* inicia definindo o papel do LLM como "um assistente especializado em reescrita de textos para melhorar a acessibilidade comunicativa". Este contexto é crucial para direcionar o modelo a focar na clareza e na remoção de elementos que dificultam a compreensão para pessoas com TEA.
- 2) **Instruções de Reescrita Explícitas:** Um conjunto de instruções detalhadas é fornecido, abrangendo os três principais pilares da reescrita:
 - **Remoção de sarcasmo:** Instruções claras para substituir frases irônicas por suas mensagens literais e diretas, evitando sarcasmo, duplo sentido e linguagem figurada, e utilizando um tom neutro e factual. Esta diretriz é fundamental, visto que a dificuldade na compreensão de sarcasmos é uma característica proeminente em indivíduos com TEA.
 - **Resolução de Ambiguidade:** Orientações para esclarecer palavras com múltiplos significados usando o contexto, escolhendo o sentido mais direto e comum, e substituindo expressões ambíguas por termos específicos. A ambiguidade lexical representa um desafio significativo para a compreensão em TEA, e a explicitação no *prompt* visa mitigar essa dificuldade.
 - **Clareza e Simplicidade:** Instruções para usar frases curtas e diretas, preferir a voz ativa, manter a ordem lógica das informações e evitar metáforas e analogias complexas. Essas práticas são reconhecidas por promoverem a clareza textual e facilitarem o processamento da linguagem por pessoas com TEA.
 - **Preservação do Conteúdo:** Uma instrução explícita para manter todas as informações essenciais, sem alterar fatos, dados específicos, nomes próprios, datas ou números. Isso assegura a fidelidade do texto reescrito ao conteúdo original, um critério importante na avaliação do projeto.
- 3) **Elementos Problemáticos Identificados Dinamicamente:** Uma característica chave do *prompt engineering* empregado é a inclusão dinâmica das frases irônicas e palavras ambíguas detectadas no texto origi-

nal pelas etapas anteriores do pipeline. Ao explicitar "OS SEGUINTES ELEMENTOS FORAM IDENTIFICADOS COMO PROBLEMÁTICOS E REQUEREM ATENÇÃO ESPECIAL", o *prompt* direciona o LLM a focar suas transformações especificamente nesses pontos. Isso aumenta a eficácia da reescrita, pois o modelo é informado precisamente sobre onde aplicar os princípios de clareza e desambiguação.

- 4) **Exemplos de Transformação (*Few-Shot Learning*):** O *prompt* inclui exemplos práticos de transformações de ironia para reescrita direta, ambiguidade para clareza e linguagem figurada para literal. A inclusão desses exemplos guia o LLM na aplicação das regras de reescrita, fornecendo um modelo de como a saída desejada deve ser formatada e o tom a ser adotado.
- 5) **Instrução de Formato de Saída:** Por fim, o *prompt* solicita que o LLM forneça "APENAS O TEXTO REESCRITO, SEM QUALQUER TEXTO INTRODUTÓRIO OU EXPLICATIVO". Esta instrução visa garantir que a saída do modelo seja limpa e diretamente utilizável, evitando verborragia desnecessária.

A escolha dessas técnicas de *prompt engineering* é justificada pela necessidade de fornecer ao LLM um direcionamento claro e específico, minimizando a chance de interpretações errôneas ou reescritas que não atendam aos requisitos de acessibilidade para pessoas com TEA. Ao explicitar o contexto, as instruções, os exemplos e os elementos problemáticos, o *prompt* serve como um guia detalhado, aumentando a probabilidade de o LLM gerar um texto reescrito que seja não apenas livre de sarcasmo e ambiguidade, mas também intrinsecamente mais compreensível e direto.

IV. AVALIAÇÃO

A. Avaliação quantitativa

Para a avaliação quantitativa, gerou-se 40 textos de maneira semi-supervisionada (usando LLMs e validação manual), com o intuito de obter trechos com ampla abrangência de casos de sarcasmos e ambiguidades. Cada um deles foi processado pela aplicação e seu resultado foi anotado. Então, cada membro da pesquisa ficou responsável por classificar de acordo com 3 critérios, usando uma escala Likert com valores de 1 a 7 (onde 1 representa a menor concordância com o critério avaliado e 7 representa a maior concordância) os resultados da aplicação. Os critérios foram os seguintes:

- 1) Melhora na clareza
- 2) Fidelidade ao conteúdo original
- 3) Efetividade na remoção de elementos ambíguos ou irônicos

Com base nas classificações do grupo, obteve-se os seguintes resultados:

Pode-se observar que a média dos 3 critérios ficou entre 5 e 6. O que indica uma concordância parcial/intermediária

Table IV
ESTATÍSTICAS POR CRITÉRIO

Crítério	Média	Moda	Mediana
Melhora na clareza	5,565	5	6
Fidelidade ao conteúdo original	5,715	6	6
Efetividade na remoção de ambiguidade e sarcasmo	5,69	7	6

com o atendimento aos critérios, considerando que o valor 7 representa a concordância total.

B. Avaliação qualitativa

Os mesmos 40 textos utilizados na avaliação quantitativa foram usadas na avaliação qualitativa. Cada um deles foi processado pela aplicação. Com base no texto original e no texto gerado pela aplicação, foram calculados a similaridade do cosseno, e a diferença no índice Flesch dos dois.

Por meio dessa avaliação, obtemos os seguintes resultados:

- 1) Para 50% dos textos, comparando o texto original e o texto de saída, o índice Flesch aumentou, significando que o texto ficou mais "claro" e "simples".
- 2) Para 100% dos textos, a similaridade do cosseno entre o texto original e o texto de saída ficou acima de 80%, o que é um indicativo de que o sentido do texto se manteve.

V. CONSIDERAÇÕES FINAIS

A partir dos experimentos realizados, foi possível observar que a abordagem baseada em modelos Transformers, utilizando a última camada de embeddings e similaridade de cosseno, obteve o melhor desempenho na tarefa de desambiguação, com destaque para a acurácia de 0.67 e F1-score de 0.59, superior à abordagem clássica do algoritmo de Lesk.

Da mesma forma, o fine-tuning de um modelo Transformers se mostrou mais eficaz que o aprendizado tradicional para a tarefa de detecção de sarcasmo, principalmente devido à técnica de representação vetorial contextualizada, o que se mostra superior a vetores estáticos para captura do significado das palavras e textos. Se destaca a métrica de acurácia (0.93 versus 0.85).

A etapa de reescrita do texto utilizando técnicas de engenharia de *prompt* demonstrou potencial na clareza e acessibilidade dos textos, conforme avaliado pela métrica Flesch e pelo critério da avaliação quantitativa. Contudo, este processo pode evoluir ao se utilizar modelos de LLMs mais robustos, o que não foi possível abordar nesta pesquisa devido às limitações de hardware e de obtenção de tais modelos. Ainda, seria possível aprimorar a *engenharia de prompt* ao utilizar mais técnicas, como por exemplo, *self-refine*.

Por fim, os resultados indicam potenciais de melhoria na tarefa. Em relação ao sarcasmo, a base de dados poderia ser

mais diversa e considerar recursos textuais mais presentes e modernas no contexto da comunicação, como redes sociais. No que concerne à tarefa de ambiguidade, ela poderia envolver uma análise semântica e sintática mais profunda, indo além do nível lexical abordado no projeto. Além disso, um léxico mais estruturado pode potencializar os resultados, pois há palavras que não estão com todos os seus sentidos no léxico ou apresentam sentidos genéricos.

REFERENCES

- [1] Instituto Brasileiro de Geografia e Estatística (IBGE), "Censo 2022 identifica 2,4 milhões de pessoas diagnosticadas com autismo no Brasil," 2023. [Online]. Available: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/43464-censo-2022-identifica-2-4-milhoes-de-pessoas-diagnosticadas-com-au>
- [2] A. Villavicencio et al., "Livro de PLN – Capítulo 23: Detecção de Ironia," 1ª ed., 2021. [Online]. Available: <https://brasileiraspln.com/livro-pln/1a-edicao/parte9/cap23/cap23.html>
- [3] J. N. Tavares, "Vamos conhecer mais sobre o TEA: guia audiovisual psicoeducativo para pais e cuidadores," Revista FACIT, 2023. [Online]. Available: <https://revistas.faculdefacit.edu.br/index.php/JNT/article/view/2563/1728>
- [4] A. Reyes et al., "Detecção de Ironia em Redes Sociais," RUA - Revista de Universidades de Alicante, 2021. [Online]. Available: https://rua.ua.es/dspace/bitstream/10045/117496/1/PLN_67_23.pdf
- [5] Schuberty, "PLNCrawler: Coleta de dados de notícias satíricas e reais," GitHub, 2021. [Online]. Available: <https://github.com/schuberty/PLNCrawler/tree/master>
- [6] F. A. A. Nóbrega, "Desambiguação lexical de sentidos para o português por meio de uma abordagem multilíngue mono e multidocumento," Instituto de Ciências Matemáticas e de Computação, 2013. [Online]. Available: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-28082013-145948/publico/FernandoNobrega_revisaada.pdf
- [7] L. C. Moraes, I. C. Silvério, R. A. S. Marques, B. C. Anaia, D. F. de Paula, M. C. S. de Faria, I. Cleveston, A. S. Correia, e R. M. K. Freitag, "Análise de ambiguidade linguística em modelos de linguagem de grande escala (LLMs)," *Texto Livre: Linguagem e Tecnologia*, vol. 18, e53181, 2025. [Online]. Available: <https://www.scielo.br/j/tl/a/Z6nCFZLTHX6XVmmhJsKvWxd/>
- [8] I. F. Januario, "Um Método para Desambiguação de Sentido e Substituição Lexical Apoiado em Dicionários e Embeddings," 2019. [Online]. Available: <https://www.repositorio.ufop.br/server/api/core/bitstreams/fa6e40bd-4194-491e-b9f1-c9cbf39ec39e/content>
- [9] P. Vandenbussche, T. Scerri, R. Daniel Jr., "Word Sense Disambiguation with Transformer Models," 2020. [Online]. Available: <https://aclanthology.org/2021.semdeep-1.2.pdf>

- [10] M. Lesk “Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone” 1987 [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/318723.318728>
- [11] J. R. Goldim “Índices de Legibilidade de Flesch-Kincaid e de Facilidade de Leitura de Flesch” 2003 [Online]. Available: <https://www.ufrgs.br/bioetica/ilfk.htm>
- [12] D. Jurafsky, J. H. Martin “Speech and Language Processing” 2024 [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>
- [13] V. de Paiva, A. Rademaker, and G. de Melo, “OpenWordNet-PT: An Open Brazilian Wordnet for Reasoning,” in *Proceedings of COLING 2012: Demonstration Papers*, Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 353–360. [Online]. Available: <http://www.aclweb.org/anthology/C12-3044>
- [14] ISHIHARA, M. K.; TAMANAHA, A. C.; PERISSINOTO, J. Compreensão de ambiguidade em crianças com Transtorno Específico de Linguagem e Fala e Transtorno do Espectro Autista. CoDAS, v. 28, n. 6, p. 753–757, 12 dez. 2016.
- [15] Scholten I, Hartman CA and Hendriks P (2021) Prediction Impairment May Explain Communication Difficulties in Autism. Front. Psychol. 12:734024. doi: 10.3389/fpsyg.2021.734024
- [16] VOLKMAR, F. R. Handbook of Autism and Pervasive Developmental Disorders, Assessment, Interventions, and Policy. [s.l: s.n.].
- [17] Training Overview — Sentence Transformers documentation. Disponível em: https://sbert.net/docs/sentence_transformer/training_overview.html.
- [18] word2vec — Text. Disponível em: <https://www.tensorflow.org/text/tutorials/word2vec>.