

# O Obstáculo das Ambiguidades Estruturais da Língua para Modelos de Linguagem Linguisticamente Motivados

João Pedro Gonçalves Munhoz<sup>1</sup>, Oto Araújo Vale<sup>1</sup>

<sup>1</sup>Departamento de Letras – Universidade Federal de São Carlos (UFSCar)  
13565-905 – São Carlos – SP – Brasil

{jmunhoz@estudante.ufscar.br, otovale@ufscar.br}

**Abstract.** This article presents UDCode, a convention for encoding morphosyntactic and dependency information from Universal Dependencies. The objective is to evaluate how the granularity of UD annotations for Brazilian Portuguese impacts the recognition of named entities of time. Exploratory tests revealed that the underspecification in adverb categorization compromises precision, generating a high rate of false positives. The result shows that the effectiveness of a linguistically motivated model depends on the level of detail in the annotations. It is concluded that future work should focus on revising the annotation guidelines to include more refined adverbial categories or on methods that compensate for this lack of specificity.

**Resumo.** Este artigo apresenta o UDCode, uma convenção para a codificação de informações morfossintáticas e de dependência da Universal Dependencies. O objetivo é avaliar como a granularidade das anotações da UD para o português brasileiro impacta o reconhecimento de entidades mencionadas de tempo. Testes exploratórios revelaram que a subdeterminação na categorização de advérbios compromete a precisão, gerando alta taxa de falsos positivos. O resultado evidencia que a eficácia de um modelo linguisticamente motivado depende do nível de detalhe das anotações. Conclui-se que trabalhos futuros devem focar na revisão das diretrizes de anotação para incluir categorias adverbiais mais refinadas ou em métodos que compensem essa falta de especificidade.

## 1. Introdução

Apesar de bastante desenvolvida, a tarefa do Reconhecimento de Entidades Mencionadas (REM)<sup>1</sup> permanece como um desafio não resolvido [Marrero et al. 2013]. Nos últimos anos, a prolífica pesquisa em inteligência artificial permitiu aumentos significativos na precisão de ferramentas voltadas ao REM, inclusive para o português brasileiro [Souza et al. 2020]. Mesmo os Grandes Modelos de Linguagem de propósito geral (isto é, aqueles não especificamente projetados para o uso em tarefas linguísticas) têm sido capazes de anotar Entidades Mencionadas em textos diversos de maneira produtiva para análises linguísticas, ainda que não isentos de erros [Tsitseklis et al. 2024] [Hu et al. 2023].

<sup>1</sup>Optou-se por utilizar o termo *Entidades Mencionadas* em detrimento de *Entidades Nomeadas*. Considera-se que *mencionadas* é uma tradução mais precisa e representativa da tarefa de *Named Entity Recognition* (NER). O próprio uso da expressão “*to name names*” em inglês, no sentido de “mencionar/citar nomes”, corrobora essa interpretação.

Dentro do vasto escopo do REM, a identificação e categorização de entidades mencionadas de tempo (como datas, horários e períodos) representam um desafio particular e de grande relevância para diversas aplicações, desde a extração de informações em textos históricos até documentos do domínio clínico [Zhong and Cambria 2021].

As recentes dinâmicas comerciais no mercado da tecnologia propiciaram a geração de um fluxo massivo de modelos poderosos de linguagem. Por serem quase sempre disponibilizados de maneira gratuita, a incorporação destes modelos como ferramentas de anotação ou análise se torna cada vez mais atrativa para pesquisadores das áreas de Linguística.

Modelos comerciais como ChatGPT, Gemini ou Qwen, ainda que gratuitos, nem sempre são totalmente abertos. A utilização destes modelos suscita questões ético-metodológicas, pois a suscetibilidade dos *outputs* às mais diminutas nuances de *prompt* [Blackwell et al. 2024] torna a reproduzibilidade dos resultados mais difícil [Liao and Vaughan 2023]. Além disso, o funcionamento interno e as múltiplas inferências que tornam os Grandes Modelos de Linguagem tão poderosos ainda não puderam ser completamente explicados [Rai 2020]. Sendo assim, mesmo com o uso produtivo desses modelos na análise linguística, existe um grande espaço para o desenvolvimento de modelos de Inteligência Artificial mais explicáveis [Panigutti et al. 2023].

Outro desafio pertinente é o desenvolvimento de modelos de linguagem mais eficientes. Hoje, o treinamento de modelos de inteligência artificial ainda é custoso e exige uma grande quantidade de dados de treinamento para ser produtivo, o que pode ser um desafio para a pesquisa científica [Hillier et al. 2024].

Este artigo utiliza uma abordagem baseada nas abstrações linguísticas da morfossintaxe e da gramática de dependências, conforme especificadas pela Universal Dependencies (UD) [Nivre et al. 2020], para diagnosticar potenciais desafios para o treinamento eficiente de um modelo de linguagem capaz de realizar a tarefa de Reconhecimento de Entidades Mencionadas de tempo. Esta abordagem visa enfrentar os problemas supracitados ao passo em que também busca permitir melhor rastreabilidade das decisões de anotação resultantes do processamento por um modelo de linguagem treinado sob esta abstração. Contudo, a aplicação prática da UD para a identificação de entidades mencionadas de tempo revela uma lacuna importante: a falta de especificação detalhada das classes de advérbios, um elemento crucial para a distinção de marcadores temporais.

## 2. UDCode

O UDCode é uma convenção sugerida para a codificação de relações de dependência e *features* morfossintáticas através de *tokens*. A codificação proposta é uma conversão de qualquer matriz CoNLL-U [Universal Dependencies contributors 2025a] para uma representação linear das informações linguísticas contidas nessa matriz.

A principal ideia do UDCode é servir como um comprometimento ideal entre duas necessidades cruciais no Processamento de Linguagem Natural (PLN). Primeiramente, ele oferece uma representação passível de ser processada eficientemente por modelos de linguagem, permitindo que estas ferramentas explorem a riqueza das informações morfossintáticas e de dependência. Em segundo lugar, e igualmente importante, o UDCode é recuperável por um analista humano, já que pode ser facilmente convertido de volta para uma representação arbórea, mais familiar aos linguistas.

Deste modo, sua utilidade para a análise linguística pode se equiparar — em alguma medida — ao uso de um CoNLL-U, que já é uma ferramenta padrão para dados anotados em Universal Dependencies. O benefício inerente ao UDCode reside na possibilidade de permitir que um modelo de linguagem compartilhe da mesma superfície de análise linguística cognoscível por um linguista humano. Espera-se que isso possa resultar em um aumento significativo da explicabilidade das decisões de anotação e da rastreabilidade do processo de inferência do modelo.

Em termos de representação, o UDCode concebe uma sequência de *tokens* onde cada *token* encapsula informações sobre a *part-of-speech*, *features* morfológicas e suas relações de dependência com outros *tokens* na sentença, utilizando marcadores ou um formato predefinido que o torne “legível” tanto por máquinas quanto por humanos. Ele deve ser capaz de permitir localizar as relações de dependência entre quaisquer *parts-of-speech* em um corpus de anotação UD convertido para essa codificação, bem como pelas especificidades das *features* de cada *token*.

Além disso, espera-se que o fato de a codificação proposta não incorporar nenhuma informação lexical faça com que um modelo de linguagem treinado sobre essa abstração realize toda e qualquer inferência exclusivamente sobre as estruturas linguísticas, nunca sobre o léxico.

### **2.1. Especificações do UDCode**

**Figura 1. Exemplo de conversão de um CoNLL-U para UDCode**

A Figura 1 exemplifica uma possível conversão de um arquivo CoNLL-U para uma sequência de *tokens* codificados em UDCode. Os *tokens* são uma combinação alfanumérica de caracteres, onde cada posição de caractere representa uma informação linguística sobre a palavra. O primeiro caractere é o indicador da *part-of-speech* e define a extensão total de cada *token* já que, para cada *part-of-speech*, haverá um conjunto possível de *features* correspondentes. Um *NOUN*, por exemplo, não codificará uma *feature* como *VerbForm*, mas um *VERB* ou um *AUX* sim. Dessa maneira, o primeiro caractere é o responsável por informar quais *features* estão codificadas no *token*. Na Tabela 1 estão relacionadas as *parts-of-speech* propostas pela Universal Dependencies e seus índices UDCode correspondentes, ambos em ordem alfabética.

**Tabela 1.** Índice alfabético de *parts-of-speech* da Universal Dependencies

ADJ	A	INTJ	G	PUNCT	M
ADP	B	NOUN	H	SCONJ	N
ADV	C	NUM	I	SYM	O
AUX	D	PART	J	VERB	P
CCONJ	E	PRON	K	X	Q
DET	F	PROPN	L		

Os caracteres subsequentes ao primeiro são as *features* codificadas. Se uma *feature* pode possuir apenas valor nulo ou verdadeiro, é representada por 0 ou 1 (como a *feature Abbr*, por exemplo). Se possui outros valores possíveis (como a *feature Gender*, que pode ser *female* ou *male*) sua posição é preenchida com um valor numérico — a partir de 1 — correspondente aos valores possíveis da *feature* em ordem alfabética. Numa primeira implementação do UDCode, as posições e valores possíveis para cada *feature* foram baseados nas estatísticas de ocorrência de *feature* por *part-of-speech* do *treebank UD Portuguese Bosque* [Universal Dependencies contributors 2025b].

O antepenúltimo e o penúltimo caractere de cada *token*, juntos, correspondem à relação de dependência do *token* com seu *head*. O *head*, por sua vez, é identificado pelo número que encerra o *token*, o qual corresponde diretamente ao índice do *token* governante na matriz CoNLL-U. A implementação atual codifica as 43 relações de dependência encontradas no *treebank UD Portuguese Bosque* e vão de AA à BQ.

## 2.2. Potencial do UDCode para o treinamento de modelos sob um corpus de tamanho reduzido

A abordagem convencional para o treinamento de Grandes Modelos de Linguagem baseia-se predominantemente na exposição massiva ao léxico em sua forma bruta. Modelos são treinados em grandes quantidades de texto, aprendendo padrões linguísticos e semânticos a partir da co-ocorrência e da distribuição lexical. Embora eficaz, essa estratégia demanda volumes de dados computacionalmente custosos e carece de uma compreensão linguística explícita.

Nesse contexto, a hipótese que motiva o UDCode é que o treinamento sobre uma representação que já encapsula abstrações linguísticas ricas, como *features* morfológicas e relações de dependência, poderia ser ainda mais eficiente, especialmente em cenários de domínios mais específicos, onde corpora de grande escala não são uma opção viável. Ao converter dados textuais para uma representação linear que encapsula informações morfossintáticas e relações de dependência pelo *framework* da Universal Dependencies, o UDCode fornece uma camada rica de abstrações linguísticas.

Espera-se que um modelo treinado sobre o UDCode possa desenvolver uma compreensão mais estruturada da gramática, resultando em maior robustez para tarefas que exigem raciocínio linguístico apurado, como o Reconhecimento de Entidades Mencionadas. Isso sugere que o modelo não precisaria “descobrir” ou inferir essas relações complexas a partir de padrões estatísticos massivos, mas as receberia como parte inerente de sua entrada. A representação explícita das *features* linguísticas poderia também oferecer

maior rastreabilidade sobre o funcionamento do modelo, alinhando-se ao objetivo de uma IA mais explicável.

Embora a avaliação empírica desta hipótese seja um passo para trabalhos futuros, a premissa de que a injeção direta de informações linguísticas estruturadas no processo de treinamento pode otimizar a aprendizagem de modelos de linguagem pode ser um caminho promissor para o futuro do PLN.

### **3. O desafio da subdeterminação na UD para entidades mencionadas de tempo**

A Universal Dependencies é um *framework* amplamente adotado para a anotação de dependências sintáticas e características morfológicas em diversos idiomas. No entanto, ao tentarmos operar exclusivamente com a descrição morfossintática das sentenças, o *framework* da Universal Dependencies se mostra insuficiente para a abstração das Entidades Mencionadas. Sua baixa granularidade na categorização de advérbios resulta em uma subdeterminação da anotação morfossintática.

Essa falta de especificidade reside na forma como a UD é aplicada ao português brasileiro. Embora o *framework* geral da UD preveja mecanismos para uma diferenciação fina, como a sub-relação *advmod:tmod* ou a feature *AdvType=Time*, a aplicação dessas especificações é escassa e inconsistente entre línguas. Como resultado, um advérbio de tempo (e.g., “ontem”), de modo (e.g., “rapidamente”) ou de lugar (e.g., “aqui”) recebem a mesma tag *ADV* e a mesma relação *advmod*. Para a tarefa de REM temporal, essa indistinção é um obstáculo significativo, pois impede a diferenciação de marcadores temporais com base apenas na estrutura sintática. Como resultado, os *parsers* e ferramentas que dependem desses recursos não anotam o tipo semântico dos advérbios. Além disso, é preciso notar que a complexidade da referência temporal transcende a anotação, envolvendo aspectos semântico-discursivos como a resolução de anáforas e a própria ontologia do tempo [Kamp and Reyle 1993] [Ilari et al. 2014].

#### **3.1. Teste do UDCODE para o reconhecimento de Entidades Mencionadas em uma entrevista**

Para investigar o potencial do UDCODE na identificação de entidades mencionadas de tempo, realizou-se um experimento inicial. O objetivo primordial foi examinar como a representação estruturada do UDCODE poderia facilitar a localização de marcadores temporais, com base em suas propriedades morfossintáticas e suas relações de dependência. O processo de teste foi sistematicamente dividido nas etapas descritas a seguir.

##### **3.1.1. Preparação dos dados de referência a partir do HAREM**

A etapa inicial consistiu na criação de um conjunto de dados de referência contendo entidades mencionadas de tempo conhecidas. Para isso, utilizamos como base a Coleção Dourada do Segundo HAREM (Avaliação de Reconhecimento de Entidades Mencionadas) [Mota and Santos 2008]. Desse corpus, todas as ocorrências de entidades mencionadas de tempo previamente anotadas foram isoladas.

Em seguida, cada uma dessas entidades temporais foi processada em uma *pipeline* automática: primeiro, o PortTokenizer [Lopes 2024] foi utilizado para a tokenização

e, na sequência, o PortParser [Lopes and Pardo 2024] realizou a análise morfossintática e de dependências, gerando um arquivo CoNLL-U. Finalmente, esses dados CoNLL-U processados foram convertidos para a representação linear do UDCode. Este conjunto de dados UDCode resultante funcionou como um “catálogo” de padrões morfossintáticos e sintáticos característicos de entidades mencionadas de tempo, conforme o *framework* UD e a convenção UDCode, aqui proposta.

### **3.1.2. Localização de potenciais entidades mencionadas de tempo em uma entrevista do programa Roda Viva**

Para o corpus de teste, selecionamos uma entrevista do programa Roda Viva que teve como entrevistado o artista Mano Brown. Um ponto crucial aqui é que este corpus já possuía anotações linguísticas completas e de alta qualidade no formato CoNLL-U, seguindo as diretrizes da Universal Dependencies [Barros and Vale 2024]. Isso garantiu que o corpus de teste já estivesse em um formato compatível e com uma análise linguística prévia. A partir dessas anotações CoNLL-U, realizamos uma conversão direta para a representação UDCode, assegurando consistência na codificação entre os dados de referência e os dados de teste.

Com ambos os conjuntos de dados — as entidades mencionadas de tempo do HAREM e a entrevista do Roda Viva — devidamente convertidos para o formato UDCode, o passo final envolveu a tentativa de identificar potenciais entidades mencionadas de tempo no corpus da entrevista. O processo consistiu em uma busca por correspondência exata: cada sequência de UDCode gerada para as entidades de tempo do HAREM foi usada como uma consulta direta sobre o corpus da entrevista, também convertido para UDCode. O objetivo era verificar se os mesmos padrões morfossintáticos e de dependência se repetiam. É importante ressaltar que, nesta fase exploratória, não houve o treinamento de um modelo de linguagem formal. O objetivo foi avaliar a capacidade do UDCode, por si só, de revelar e destacar segmentos que linguisticamente corresponderiam a expressões temporais.

Este teste, embora de natureza exploratória, permitiu uma avaliação inicial da viabilidade e da granularidade do UDCode para revelar padrões relevantes para o reconhecimento de entidades mencionadas de tempo. Ao mesmo tempo, ele se tornou uma ferramenta para evidenciar como as características e as limitações do *framework* UD influenciam diretamente a identificação dessas entidades.

## **3.2. Resultados e observações do teste**

A aplicação do UDCode, em caráter exploratório e qualitativo, permitiu diagnosticar como a subdeterminação das anotações da UD impacta a identificação de entidades de tempo. Por um lado, a abordagem de busca por padrões mostrou-se eficaz para estruturas diretas. Por exemplo, a sequência UDCode B00A12 I00100B00, correspondente a ”Em 1997”, encontrou correspondências precisas:

- “O grupo, formado em São Paulo **em 1988**, tem além de Mano Brown [...]”
- “**Em 1997**, com o disco Sobrevivendo no Inferno [...]”

A maior dificuldade, contudo, emergiu com os advérbios. A falta de especificidade na categorização da UD teve um impacto direto na precisão. A subdeterminação da

tag *ADV* foi a principal fonte de falsos positivos no teste. O advérbio “anualmente” (codificado como C000000B00) do HAREM, de fato, retornou algumas correspondências temporais corretas, como:

- “**Atualmente** atinge também a classe média”
- “**Antigamente**, você tinha que [...]”

Contudo, essa mesma codificação também produziu um número significativo de falsos positivos, identificando frases sem nenhuma relação temporal, como:

- “**Dentro** do que a gente vê [...]”
- “**Bem** menos do que poderia [...]”

Similarmente, o padrão para a expressão “mais tarde” (codificado como C000000AD2 C000000B00) também resultou neste problema. Embora “mais tarde” seja inequivocamente uma entidade de tempo, sua codificação UDCode, baseada na falta de especificidade dos advérbios na UD para o português brasileiro, levou a retornos como:

- “**Bem menos** do que poderia [...]”
- “**Nem lá?**”

Essas observações sugerem que a dependência das abstrações da Universal Dependencies para a anotação morfossintática e de dependências tornam o UDCode suscetível às limitações inerentes às diretrizes e implementações atuais da UD para o português brasileiro. Para o reconhecimento de entidades mencionadas de tempo, a ausência de uma categorização mais granular ou de *features* específicas para advérbios de tempo — como a *feature* *AdvType=Time* ou a sub-relação *admod:tmmod* — nas anotações do *treebank* de referência introduz uma subdeterminação da anotação que o UDCode, por si só, não é capaz de solucionar.

Essa falta de especificação é um obstáculo significativo, pois advérbios de tempo são fortes indicadores linguísticos de expressões temporais. A ausência de *corpora* ou *parsers* para o português capazes de realizar anotações que especifiquem a função temporal de um advérbio leva a uma indefinição que impede sistemas automatizados de distinguir, por exemplo, “Ele chegou tarde” (advérbio de tempo) de “Ele falou alto” (advérbio de modo). Consequentemente, qualquer abordagem que dependa exclusivamente dos *parsers* de UD atuais para o português enfrentará dificuldades para alcançar alta precisão na tarefa, visto que elementos cruciais para a desambiguação não estão indicados nos dados de treinamento.

#### 4. Conclusão

Este artigo apresentou o UDCode, uma convenção para a codificação linear de informações da Universal Dependencies (UD), e o utilizou como ferramenta de diagnóstico. A concepção do UDCode buscou um equilíbrio entre uma representação processável por máquinas e a rastreabilidade para analistas humanos. A motivação teórica é que tal abordagem poderia otimizar o treinamento de modelos de linguagem, potencialmente permitindo o uso de corpora menores e promovendo maior explicabilidade.

Para investigar os desafios práticos dessa abordagem, um teste exploratório focou no reconhecimento de entidades de tempo. Os resultados revelaram uma limitação crucial:

enquanto o UDCode foi eficaz na identificação de estruturas explícitas (e.g., “Em 1997”), a performance foi severamente comprometida pela subdeterminação na categorização de advérbios na UD para o português brasileiro. A baixa granularidade da tag *ADV* resultou em uma alta taxa de falsos positivos, exemplificada por casos onde advérbios de tempo geravam correspondências com expressões sem qualquer conotação temporal.

Os resultados mostram que a eficácia de abordagens linguisticamente motivadas está intrinsecamente ligada à granularidade das anotações subjacentes. A falta de especificidade na UD, neste caso, impede que um método baseado puramente em sua estrutura resolva a tarefa de forma satisfatória.

Para trabalhos futuros, torna-se imperativo explorar formas de trabalhar essa lacuna. A principal via é a revisão das diretrizes de anotação da UD para o português, visando incorporar sistematicamente uma categorização adverbial mais fina (com *features* como *AdvType=Time* ou sub-relações como *admod:tmod*). Alternativamente, podem-se desenvolver métodos que compensem essa subdeterminação com informações de contexto mais amplo. Acreditamos que o enriquecimento e a máxima granularidade morfossintática são passos fundamentais para o avanço de modelos de linguagem linguisticamente motivados, mais precisos e explicáveis.

## Agradecimentos

João Pedro Gonçalves Munhoz é beneficiário de uma bolsa Univesp. Parte deste trabalho foi realizado no âmbito do Centro de Inteligência Artificial da USP (C4AI - <http://c4ai.inova.usp.br/>), que tem o apoio da IBM e da FAPESP (processo 2019/07665-4). Este projeto também foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, que tem recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-Softex, coordenado pela Softex e publicado como Residência em TIC 13, DOU 01245.010222/2022-44. Os autores agradecem o apoio financeiro concedido pelo PPGL/UFSCar de verba CAPES/PROAP para a participação neste evento.

## Referências

- Barros, C. D. and Vale, O. A. (2024). Roda viva: um corpus oral e a universal dependencies. In *Anais Eletrônicos do XVI Encontro de Linguística de Corpus e da XII Escola Brasileira de Linguística Computacional*, volume 1, pages 89–94, Brasília.
- Blackwell, R. E., Barry, J., and Cohn, A. G. (2024). Towards reproducible llm evaluation: Quantifying uncertainty in llm benchmark scores. *arXiv preprint arXiv:2410.03492*.
- Hillier, D., Guertler, L., Tan, C., Agrawal, P., Ruirui, C., and Cheng, B. (2024). Super tiny language models. *arXiv preprint arXiv:2405.14159*.
- Hu, Y., Ameer, I., Zuo, X., Peng, X., Zhou, Y., Li, Z., Li, Y., Li, J., Jiang, X., and Xu, H. (2023). Zero-shot clinical entity recognition using chatgpt. *arXiv preprint arXiv:2303.16416*.
- Ilari, R., de Castilho, A. T., and Gnero, M. B. M. (2014). *Gramática do português culto falado no Brasil: Palavras de classe aberta*.
- Kamp, H. and Reyle, U. (1993). *From discourse to logic*. Studies in Linguistics and Philosophy. Springer, Dordrecht, Netherlands, 1993 edition.

- Liao, Q. V. and Vaughan, J. W. (2023). Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 10.
- Lopes, L. (2024). portTokenizer. <https://github.com/LuceleneL/portTokenizer>.
- Lopes, L. and Pardo, T. (2024). Towards portparser - a highly accurate parsing system for Brazilian Portuguese following the Universal Dependencies framework. 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 - Vol. 1*, pages 401–410, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.
- Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., and Gómez-Berbís, J. M. (2013). Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489.
- Mota, C. and Santos, D., editors (2008). *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca.
- Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., and Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- Panigutti, C., Hamon, R., Hupont, I., Fernandez Llorca, D., Fano Yela, D., Junklewitz, H., Scalzo, S., Mazzini, G., Sanchez, I., Soler Garrido, J., and Gomez, E. (2023). The role of explainable ai in the context of the ai act. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’23, page 1139–1150, New York, NY, USA. Association for Computing Machinery.
- Rai, A. (2020). Explainable ai: From black box to glass box. *Journal of the academy of marketing science*, 48:137–141.
- Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pretrained BERT models for brazilian portuguese. In *Intelligent Systems*, Lecture notes in computer science, pages 403–417. Springer International Publishing, Cham.
- Tsitseklis, K., Stavropoulou, G., and Papavassiliou, S. (2024). Custom named entity recognition vs chatgpt prompting: a paleontology experiment. In *2024 Panhellenic Conference on Electronics & Telecommunications (PACET)*, pages 1–5. IEEE.
- Universal Dependencies contributors (2025a). *CoNLL-U Format*. Universal Dependencies.
- Universal Dependencies contributors (2025b). *UD Portuguese Bosque*. Universal Dependencies.
- Zhong, X. and Cambria, E. (2021). Literature review. In *Socio-Affective Computing*, pages 15–34. Springer International Publishing, Cham.