Disfluency Detection and Removal in Speech Transcriptions via Large Language Models

Pedro L. S. de Lima[1] **, Cláudio E. C. Campelo**[1]

[1]Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba - Brasil

pedro.lima@ccc.ufcg.edu.br

campelo@dsc.ufcg.edu.br

Abstract. *The field of Automatic Speech Recognition (ASR) has significantly expanded within the technological landscape due to its extensive use in sectors such as education, healthcare, and customer service. Many modern applications depend on analyzing spoken content through Speech-to-Text (STT) conversion models. However, transcriptions produced by these systems often contain undesirable elements, such as word repetitions and the prolongation of certain sounds, known as disfluencies or linguistic crutches. These elements can negatively affect the quality of automatic content analysis by Natural Language Processing (NLP) models, including those for named entity recognition, emotion detection, or sentiment analysis. Therefore, this study aims to evaluate the feasibility of identifying and eliminating linguistic disfluencies using Large Language Models (LLMs), such as GPT-4, LLaMA, Claude, and Gemini, through Prompt Engineering techniques. The approach was tested using a corpus of debate transcriptions with manually annotated disfluency occurrences, yielding promising results.*

1. Introduction

Automatic Speech Recognition (ASR) has become essential in modern society, enabling the conversion of human speech into written text. This technology facilitates a range of applications through Speech-to-Text (STT) models, including virtual assistants, meeting transcription, automatic captioning, and customer service. Despite significant advances in speech recognition accuracy, a constant feature in transcriptions generated by these systems is the presence of linguistic disfluencies. During human speech production, it is common to generate various sounds within speech, known as disfluencies.

Disfluencies have been extensively studied and are primarily classified into three types: hesitations, repetitions, and corrections [Corley and Stewart 2008]. When a speech model transcribes voice into text, it often overlooks the context of the spoken words, focusing instead on achieving an accurate transcription. As a result, these disfluencies are common and appear in caption transcriptions, meeting notes, and any text derived from spontaneous human speech. Various studies explore different techniques for disfluency detection, ranging from unimodal to multimodal approaches, some even use Transformer-based methods, but none thoroughly investigate the utility of modern and widely accessible Large Language Models (LLMs) for the detection and removal of linguistic crutches.

LLMs based on Transformers present a promising alternative. Due to their ability to capture complex contexts and understand linguistic nuances, such as differentiating between disfluent and fluent text, they present a promising alternative. LLMs can be easily manipulated using Prompt Engineering techniques, which involve creating instructions to guide their behavior toward a specific goal. This work aims to fill a gap in the study of disfluency detection and removal in text transcriptions by evaluating the capabilities of the most advanced LLMs available today, such as OpenAI's GPT-4o [OpenAI et al. 2024], Gemini 1.5 Pro Experimental 0827 [Team et al. 2024], Claude 3.5 [Anthropic 2024] and LLaMMa 3 (70B parameters) [Meta 2024] to assess the extent of their applicability to this task.

The main contributions of this paper include:

- An analysis of LLMs' ability to remove particular text excerpts while preserving other relevant information.
- A comparative analysis of available models and their effectiveness in handling transcribed spontaneous human speech.
- An assessment of the feasibility, in terms of computational cost, of cleaning transcriptions of natural human speech.
- A dataset with annotated disfluencies in Brazilian Portuguese.

The following sections of this paper are organized as follows: Section 2 presents a literature review, covering foundational and relevant research on disfluency detection and removal, leading up to the current state-of-the-art. Section 3 details the research methodology, explaining data collection and handling processes, as well as the construction of prompts and an exploratory data analysis, followed by Section 4, which presents the results. Finally, Section 5 offers the conclusion.

2. Related Work

Research on the detection and removal of disfluencies in speech encompasses a variety of techniques, each contributing to the advancement of the state-of-the-art in this field. Studies in this domain typically utilize one of three types of input: textual transcription, audio signal, or a combination of the two. Unimodal solutions rely on a single source of information, whereas multimodal solutions integrate multiple sources, such as audio and text, to perform the task of disfluency detection/removal. The next subsections present research carried out using the unimodal text approach, followed by the unimodal audio approach, a comparison between the two approaches, and finally the conclusion of this section.

2.1. Text-Based Approaches

In this context, [Snover et al. 2004] proposed a Transformation-Based Learning (TBL) algorithm for disfluency detection in speech transcriptions, employing lexical features (word usage and sentence structure). The system, referred to as System A, achieved results comparable to those employing prosodic features (variations in intonation, rhythm, duration, and intensity/volume of speech), demonstrating that satisfactory performance can be achieved without heavily relying on prosodic cues. The study underscores the importance of features such as the lexeme itself, Part-of-Speech (POS) tags, and word frequency for the speaker in identifying disfluencies. System A showed promising results in detecting various types of disfluencies and paved the way for future research focused on natural language processing techniques.

[Ferguson et al. 2015] proposed a conditional semi-Markovian method (semi-CRF) for disfluency detection in speech transcriptions, focusing on repairs such as repetitions and false starts. This technique utilizes lexical, structural, and prosodic features, such as pauses and word duration, extracted from alignment with the speech signal. This approach achieved an F-score of 85.4% on the Switchboard corpus (a dataset consisting of English telephone conversations collected in the United States during the 1990s), surpassing the performance of previous studies. Concurrently, [Zayats et al. 2016] introduced a novel method for disfluency detection in speech transcriptions using a Bidirectional LSTM (BLSTM) neural network. Their solution employs word embeddings (numerical representations of words), POS tags, and lexical pattern features as input. Additionally, the model incorporates an explicit repair mechanism and uses Integer Linear Programming (ILP) to enforce structural constraints on the disfluency sequence. This approach achieved an F-score of 85.9% on the Switchboard corpus. Analysis of the results indicates that this approach performs better in detecting complex disfluencies that do not involve mere repetitions of words. Despite its effectiveness, the model's reliance on predefined resources limits its adaptability to different types of disfluencies, contexts, and speaking styles.

[Bach and Huang 2019] also explored the BiLSTM technique with self-attention for disfluency detection in speech transcriptions. The authors demonstrated competitive results with BERT on the Switchboard corpus, outperforming it in terms of robustness and efficiency on out-of-domain datasets. The artificial addition of extra and incorrect words during model training proved highly effective in enhancing its robustness to various data types and transcription errors, making it a compelling alternative for disfluency detection in real-world scenarios. Furthermore, the proposed models are smaller than BERT, which results in reduced computational resource requirements overall.

2.2. Audio-Based Approaches

[Bassi et al. 2023] propose an end-to-end approach for speech transcription with disfluency removal using a large-scale pre-trained HuBERT acoustic model. The traditional two-step method, which first transcribes the audio into text and then removes disfluencies, neglects the prosodic cues present in the original audio. The proposed approach processes the audio directly and uses acoustic representations learned during pre-training to identify and remove disfluencies during transcription. The authors demonstrate that the end-to-end solution surpasses the two-step approach in terms of Word Error Rate (WER) and Character Error Rate (CER) on the Switchboard test set, achieving 12.2% WER and 7.3% CER. The study also highlights the significance of the pre-training objective: HuBERT, pre-trained with a clustering objective that groups audio representations based on similarities, significantly outperformed Wav2Vec2, which was pre-trained with a contrastive objective that maximizes similarity among similar samples and minimizes similarity among different samples. These results suggest that end-to-end models with large-scale acoustic pre-training with clustering objectives are a promising approach for accurate disfluent speech transcription.

2.3. Comparison Between Unimodal and Multimodal Models

[Romana et al. 2023] investigated the automatic detection of disfluencies in speech by comparing language-based, acoustic, and multimodal methods. Their results demonstrated that while language models such as BERT exhibited high accuracy with manual transcriptions, performance significantly declined with the use of transcriptions generated by Automatic Speech Recognition (ASR). Acoustic approaches utilizing models like Wav2Vec 2.0, HuBERT, and WavLM proved promising by avoiding reliance on transcriptions. However, the authors found that multimodal solutions combining acoustic and linguistic information through a BLSTM fusion network achieved the best results, outperforming unimodal techniques in disfluency detection and categorization. This study highlights the potential of multimodal methods for creating more robust disfluency detection systems.

The academic works presented in this section illustrate the progress made in the field, with advanced techniques in artificial intelligence, transformers, and robust multimodal methods applicable to various data types and transcription errors. These solutions have proven effective in detecting and removing disfluencies across diverse contexts. However, the use of widely available Large Language Models (LLMs) for cleaning automatic transcriptions has been insufficiently studied. Therefore, there is a need to investigate how LLMs can be leveraged for this purpose, complementing the advancements achieved in the reviewed academic works and democratizing access to these technologies.

3. Methodology

This section contains information about the methodology used in the research, including how the data was obtained and organized, and the construction of the prompts.

3.1. The Dataset

The dataset for this study consists of text extracted from four debate sessions held at Federal University of Campina Grande to analyze debater performance. It includes 114 minutes of transcribed audio in Portuguese, providing insights into the dynamics and effectiveness of various debating techniques. The debates were moderated, with each session involving 4 to 5 debaters discussing topics related to Artificial Intelligence. Each debater had a chance to speak following questions posed by the moderator, and interruptions were not allowed, resulting in a free-flowing and spontaneous discourse. After the debates, the audio recordings were transcribed using Microsoft's Azure model, with the transcripts stored in a JSON file. This file was then converted into Excel tables containing all the transcribed data. The data underwent human review to correct major transcription errors, such as non-existent or meaningless words. Additionally, each table was annotated for disfluencies. Disfluencies were categorized into three types: hesitations, repetitions, and corrections. Four HTML-style tags were created to mark these disfluencies in the text:

- `<hes {content}/>`, which marks hesitations
- `<rep {content}/>`, which marks repetitions

- `<erro {content}/>`, which marks errors
- `<corr {content}/>`, which marks corrections

This marking and correction process resulted in four Excel files with the transcriptions of the respective debates. These files were then subjected to an exploratory data analysis.

3.2. The Prompts

To perform the task of disfluency detection and removal, four different prompts were developed. To determine which prompt technique is most effective, three types of prompt engineering methods were tested:

- Zero-Shot Prompting
- Few-Shot Prompting
- Chain-of-Thought Prompting

These three types of prompts differ significantly in how they present information to the language model (LLM), and the study aims to understand the extent of the LLMs' knowledge about disfluencies. In the Zero-Shot case, the prompt provides little or no context about the task, so it was divided into two prompts. The first prompt is a direct command to the LLM to remove repetitions, hesitations, and corrections from the text, while keeping it otherwise unchanged. The second prompt adds a description of what disfluencies are and how the three targeted types are characterized. The Few-Shot prompt includes all the information from the first two prompts, as well as an example of disfluent text in three stages: the original disfluent text, the text with disfluency tags, and the cleaned text. Finally, the Chain-of-Thought prompt is designed to help the LLM adopt a step-by-step approach to detecting and removing disfluencies from the text. These four prompts were executed with each of the LLMs. The average number of tokens processed by the LLMs in Group 14, the smallest group, ranged from 4,273 tokens (with the smallest prompt) to 5,041 tokens (with the largest prompt). In contrast, Group 1, the largest group, processed between 5,250 tokens (smallest prompt) and 6,004 tokens (largest prompt). This calculation was estimated using the Tokenizer from the OpenAI Platform.

Prompting Technique	**Context**
Zero-Shot Prompting	None
Zero-Shot Prompting	Definition of disfluencies
Few-Shot Prompting	Definition of disfluencies and a three-stage snapshot of the text during the disfluency cleaning process
Chain-of-Thought Prompting	Definition of disfluencies plus a guide on how to recognize and remove each type of disfluencies

Table 1. Prompts Created For the Task

3.3. Exploratory Data Analysis

Data from the tagged transcriptions in Excel files were analyzed to gain an overview of how each disfluent text is characterized. The initial analysis focused on the quantity of disfluencies per group. For this purpose, disfluencies were tallied in each file employing the markers described in the Dataset section. These counts were aggregated for each

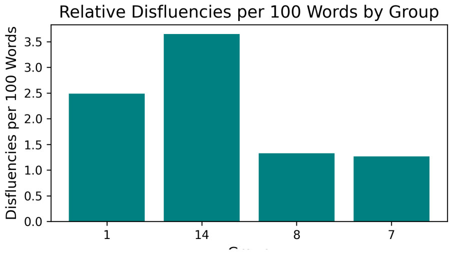

Figure 1. Total Relative Disfluencies per 100 Words

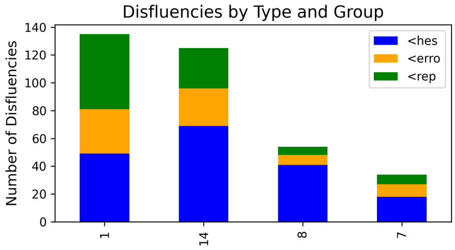

Figure 2. Disfluencies by Type and Group

group, and the totals were visualized using graphs to aid interpretation. Figure 1 displays the comparison of disfluency rates across four groups, labeled 1, 14, 8, and 7 on the X-axis. This figure presents the proportion of disfluencies calculated per 100 words for each group, facilitating a comparison of the relative frequency of disfluencies between the groups. Figure 2, using the same group labels (1, 14, 8, and 7) on the X-axis, depicts the number of disfluencies broken down by type (hesitation, error, repetition). This figure illustrates the distribution of different disfluency types within each group.

3.4. Configuration and Execution of LLMs

The execution of data in Large Language Models (LLMs) was carried out through specific Application Programming Interfaces (APIs). The Google Gemini 1.5 Pro Experimental 0827, Anthropic's Claude 3.5 Sonnet, and OpenAI's ChatGPT-4o were accessed via the official APIs provided by their respective companies. The LLaMa 3 72B was used through the Groq platform. The implementation was structured into 16 Python notebooks in the Google Colaboratory environment, with four notebooks assigned to each LLM, corresponding to debate groups. Each notebook was initialized with the configuration of the corresponding LLM, followed by the result extraction codes detailed in this methodological section, and then executed using the pre-established prompts. The results obtained were recorded at the end of each notebook, later compiled into tables for this work, and analyzed for the research objectives. In a separate notebook, an exploratory data analysis was conducted using Excel files from the groups, with the procedures and results described in detail in this work.

4. Results

Table 2. Zero-Shot (No Context) - Group 14

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	14.06%	97.95%	85.69
GPT-4o	62.50%	96.83%	49.77
LLaMa	53.12%	95.24%	21.86
Claude	57.81%	32.98%	18.47

The data presented in Tables 2 and 3 clearly show that when using Zero-Shot prompts in Group 14 (the most disfluent group), GPT-4o, Gemini, and LLaMa maintained a relatively good textual structure, as indicated by the Levenshtein similarity value. None of the LLMs successfully balanced the removal of disfluencies while preserving the

Table 3. Zero-Shot (With Context) - Group 14

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	10.94%	97.96%	86.23
GPT-4o	60.94%	74.96%	49.77
LLaMa	62.50%	51.16%	8.79
Claude	60.94%	33.37%	18.65

original text's quality with these two prompts, but text maintenance results for Claude and LLaMa fell significantly below expectations for most tested prompts, making them currently unreliable for this type of task. Therefore, the following analysis focuses solely on GPT-4o and Gemini 1.5.

Table 4. Test Results for GPT-4o and Gemini - Few Shot - Group 14

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	51.56%	98.18%	82.57
GPT-4o	67.19%	96.83%	74.73

With Few-Shot prompting (Table 4), GPT-4o achieved a 67.19% disfluency removal rate while maintaining 96.83% of the original text. It also surpassed Gemini in response time. Although Gemini had a slightly higher text maintenance rate, it performed poorly in removing disfluencies.

Table 5. Test Results for GPT-4o and Gemini - Chain of Thought - Group 14

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	26.56%	98.09%	84.61
GPT-4o	68.75%	97.85%	45.76

Using Chain-of-Thought prompting (Table 5), GPT-4o was the only one among the four LLMs to produce a minimally favorable result. When compared to Gemini, GPT-4o achieved a 68.75% total disfluency removal rate, despite a similar text maintenance rate, while Gemini, though maintaining text quality, failed in removing disfluencies.

Table 6. Test Results for GPT-4o and Gemini - Few Shot - Group 8

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	62.96%	95.80%	102.58
GPT-4o	48.15%	88.22%	47.05

Table 7. Test Results for GPT-4o and Gemini - Chain of Thought - Group 8

Model	Total Removal Rate %	Levenshtein Similarity %	Time (seconds)
Gemini	33.33%	98.19%	104.26
GPT-4o	40.74%	88.51%	45.25

In Group 8, one of the least disfluent groups, GPT-4o's effectiveness declined in both disfluency removal and maintaining text fluency, as shown in Tables 6 and 7.

Table 8. Test Results for GPT-4o and Gemini - Few Shot - Group 1

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	63.89%	96.68%	126.40
GPT-4o	68.06%	78.18%	52.48

Table 9. GPT-4o and Gemini - Chain of Thought - Group 1

Model	Total Removal Rate	Levenshtein Similarity	Time (seconds)
Gemini	25.00%	97.85%	127.22
GPT-4o	68.06%	77.55%	53.69

Gemini achieved a 62.96% removal rate with good text maintenance, albeit taking more than twice as long. This trend, where GPT-4o did not match Gemini in text maintenance, was also observed in Group 1, as shown in Tables 8 and 9, which is the largest group but not as disfluent as Group 14. The models demonstrated high effectiveness in removing repetitions, achieving 91.30% removal in Group 14 for GPT-4o, compared to Gemini's 56.52% in the Few-Shot prompt. In the Chain-of-Thought prompt, GPT-4o maintained a consistent removal rate of 91.30% while also outpacing Gemini in processing time. Although GPT-4o showed strong performance in Group 14, it struggled in Group 1, where Gemini achieved 87.18% removal with superior text preservation (96.68%). These results suggest that while GPT-4o excels in specific contexts, Gemini may be more robust when handling larger, more complex texts. [1].

5. Conclusion and Future Work

This study explored the efficacy of Large Language Models (LLMs) in detecting and eliminating linguistic disfluencies from transcriptions of academic debates. By leveraging advanced prompt engineering techniques, such as Zero-Shot, Few-Shot, and Chain-of-Thought prompting, we assessed the performance of leading LLMs — GPT-4, Gemini 1.5, Claude 3.5, and LLaMa 3 — in this task. The results revealed several key insights into the capabilities and limitations of these models. GPT-4o demonstrated the highest overall performance in disfluency removal, achieving an optimal balance between removing disfluencies and maintaining text coherence, particularly under Few-Shot and Chain-of-Thought prompting conditions. Gemini 1.5 also performed well but showed variability depending on the prompt type and the specific debate group analyzed. It excelled in text maintenance but had lower removal rates compared to GPT-4o in some cases. Claude 3.5 and LLaMa 3 produced weaker results, struggling to maintain text coherence while removing disfluencies. GPT-4o demonstrated more efficient processing times compared to the other models, which is crucial for practical, real-world applications. In conclusion, while LLMs like GPT-4o and Gemini 1.5 show promise for improving transcription quality by removing disfluencies, further advancements—such as fine-tuning, employing more advanced prompt engineering techniques, integrating widely used LLMs with multimodal systems, or developing future models—are necessary to fully enhance their capabilities.

[1] Repository: https://github.com/pedrosqra/STIL

References

Anthropic (2024). Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`. Accessed: 2024-08-27.

Bach, N. and Huang, F. (2019). Noisy bilstm-based models for disfluency detection. In *Interspeech*, pages 4230–4234.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477.

Bassi, S., Duregon, G., Jalagam, S., and Roth, D. (2023). End-to-end speech recognition and disfluency removal with acoustic language model pretraining. *arXiv preprint arXiv:2309.04516*.

Corley, M. and Stewart, O. W. (2008). Hesitation disfluencies in spontaneous speech: The meaning of um. *Language and Linguistics Compass*, 2(4):589–602.

Ferguson, J., Durrett, G., and Klein, D. (2015). Disfluency detection with a semi-markov model and prosodic features. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 257–262.

Hsu, W., Bolte, B., Tsai, Y. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, abs/2106.07447.

Meta, A. (2024). Introducing llama 3: Advancements in large language models. Accessed: 2024-08-27.

OpenAI (2024). Openai tokenizer. Accessed: 2024-10-08.

OpenAI, Achiam, J., and et al., S. A. (2024). Gpt-4 technical report.

Romana, A., Koishida, K., and Provost, E. M. (2023). Automatic disfluency detection from untranscribed speech. *arXiv preprint arXiv:2311.00867*.

Snover, M., Dorr, B., and Schwartz, R. (2004). A lexically-driven algorithm for disfluency detection. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 157–160.

Team, G., Georgiev, P., and et al., V. I. L. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Zayats, V., Ostendorf, M., and Hajishirzi, H. (2016). Disfluency detection using a bidirectional lstm. *arXiv preprint arXiv:1604.03209*.