Z-SCORES: A METRIC FOR LINGUISTICALLY ASSESSING DISFLUENCY REMOVAL

Maria Teleki, Sai Janjur, Haoran Liu, Oliver Grabner, Ketan Verma, Thomas Docog, Xiangjue Dong, Lingfeng Shi, Cong Wang, Stephanie Birkelbach, Jason Kim, Yin Zhang, James Caverlee

Texas A&M University

ABSTRACT

Evaluating disfluency removal in speech requires more than aggregate token-level scores. Traditional word-based metrics such as precision, recall, and F1 ($\mathcal{E}$-Scores) capture overall performance but cannot reveal why models succeed or fail. We introduce $\mathcal{Z}$-Scores, a span-level linguistically-grounded evaluation metric that categorizes system behavior across distinct disfluency types (EDITED, INTJ, PRN). Our deterministic alignment module enables robust mapping between generated text and disfluent transcripts, allowing $\mathcal{Z}$-Scores to expose systematic weaknesses that word-level metrics obscure. By providing category-specific diagnostics, $\mathcal{Z}$-Scores enable researchers to identify model failure modes and design targeted interventions – such as tailored prompts or data augmentation – yielding measurable performance improvements. A case study with LLMs shows that Z-scores uncover challenges with INTJ and PRN disfluencies hidden in aggregate F1, directly informing model refinement strategies.

Index Terms— Disfluency removal, fluency restoration, evaluation, LLM, SLM

1. INTRODUCTION

Spontaneous speech is filled with disfluencies systematically categorized in Shriberg's linguistic framework [1], which defines **interjections (INTJ)** – e.g., *um*, *uh*, *uh-huh* – signaling hesitation or backchanneling; **parentheticals (PRN)** – e.g., *you know*, *I mean* – functioning as meta-commentary or discourse markers; and **edited nodes (EDITED)** – e.g., *Where did I put my keys – sorry, phone?* – capturing false starts, repairs, and restarts. As illustrated in Figure 1, interactions with smart speakers or wearable devices often contain such disfluencies, which have been shown to degrade downstream task performance in transcription, translation, and conversational recommendation [2, 3].

Existing approaches for evaluating disfluency removal systems rely primarily on word-level precision, recall, and F1 scores. While these metrics are useful for coarse performance measurement, they provide only a limited view. Crucially, they cannot explain why a system succeeds or fails. For example, a model's overall F1 may look strong, yet it may consistently fail to remove parentheticals or interjections — weaknesses that remain hidden in aggregate scores.

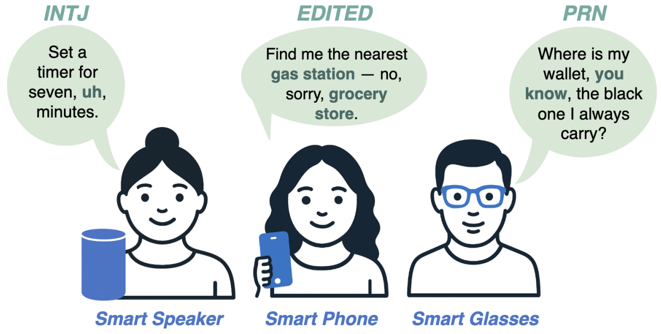

INTJ
Set a timer for seven, uh, minutes.
EDITED
Find me the nearest gas station — no, sorry, grocery store.
PRN
Where is my wallet, you know, the black one I always carry?
Smart Speaker
Smart Phone
Smart Glasses

Fig. 1. Removing disfluencies such as INTJ (*uh*), EDITED (*gas station* is replaced with *grocery store*), and PRN (*you know*) ensures clean text input for downstream tasks. Our proposed $\mathcal{Z}$-Score metric uncovers categorical errors, which can drive targeted model improvements.

To address this gap, we introduce $\mathcal{Z}$-Scores, a span-level diagnostic metric for disfluency removal that can be used to drive modeling improvements. $\mathcal{Z}$-Scores quantify how models handle distinct categories of disfluencies (EDITED, INTJ, PRN) by leveraging a deterministic alignment module that ensures reliable mapping between generated outputs and disfluent transcripts. This enables evaluation at the level of linguistic phenomena, not just tokens, and makes it possible to uncover systematic behaviors that traditional metrics obscure.

$\mathcal{Z}$-Scores complement existing word-level metrics with linguistic interpretability, as they let us evaluate how well models remove these distinct disfluency types rather than collapsing them into a single F1 score. $\mathcal{Z}$-Scores empower researchers and practitioners to ask category-aware questions such as: *Do generative models remove interjections as effectively as parentheticals? Do models fail to remove parentheticals when they occur near certain linguistic phenomena? How do prompting strategies shift performance across disfluency types?* **By providing a diagnostic lens, $\mathcal{Z}$-Scores serve as a bridge between computational evaluation and linguistic analysis, empowering researchers to uncover hidden weaknesses and guide modeling improvements in disfluency removal. We release an open-source Python package** at `https://github.com/mariateleki/zscore`.

2. RELATION TO PRIOR WORK

Previous work evaluates performance in terms of word-based precision, recall, and F-score – we denote these as $\mathcal{E}$-Scores – for the disfluency removal task as shown in Figure 2. To calculate $\mathcal{E}$-Scores, an alignment between the original text and the model-generated text is required, as shown in Table 1. Hence, previous work in this task has primarily been formulated as sequence classification, because obtaining the alignment between the ground-truth disfluent text and the GM-generated text is difficult. In our work, we provide a method for obtaining this alignment (§3.1), enabling the use of generative language models for the disfluency removal task.

2.1. Generative Models (GMs)

Compared to prior specialized models, generative models (GMs) like LLMs and SLMs offer two distinct advantages for this task: First, their extensive pretraining on diverse internet-scale text corpora provides rich world knowledge and a deep semantic understanding that smaller, task-specific models lack. This allows them to better interpret domain-specific vocabulary and contextual nuances within disfluent speech. Second, their development cycle allows for continuous updates to the base models, enabling them to adapt to evolving language – including new slang and concepts – thereby addressing the challenge of temporal language shift. As shown in Figure 2, previous work has focused on using generative models as part of the classification model pipeline for the disfluency removal task, either (i) using GM as encoders, or (ii) using GM to generate synthetic disfluency data for fine-tuning; we detail the use of GMs in these ways next. The use of generative models for disfluency removal has been underexplored due to a lack of methodology to align generated text with disfluent text. To enable this alignment, we develop a new alignment module, $\mathcal{A}$ (§3.1).

2.2. Classification Models

Classification models perform token-by-token classification of disfluent text, labeling each token in the sequence as either disfluent, I ("inside" the disfluent region) or fluent, O ("outside") as shown in column t_{CLS} of Table 1. We detail the major approaches: (i) Some works propose joint encoding. [4] use a joint LLM-ASR architecture. [5] and [6] propose transformer variants with joint encoding of labeled and unlabeled input sequences. (ii) Many works perform synthetic disfluency insertion on the original text before fine-tuning. [7] and [8] generate synthetic disfluency data from OpenAI LLMs and fine-tune small BERT models. [9] and [10] rely on rule-based methods for generating synthetic disfluency data. (iii) [11] proposes a span classifier for building parse trees. (iv)

$t_{disfluent}$	t_{tag}	t_{Φ}	t_{CLS}	$\mathbb{1}_{gt}$	$\mathbb{1}_{pred}$	$\mathbb{1}_{tp}$	$\mathbb{1}_{tn}$	$\mathbb{1}_{fp}$	$\mathbb{1}_{fn}$
i	PRN	i	I	1	0	0	0	0	1
mean	PRN	mean	I	1	0	0	0	0	1
but	NONE	but	O	0	0	0	1	0	0
she	EDITED		I	1	1	1	0	0	0
was	EDITED		I	1	1	1	0	0	0
truly	EDITED		I	1	1	1	0	0	0
she	NONE		I	0	1	0	0	1	0
-	-	Luna	O	*	*	*	*	*	*
was	NONE	was	O	0	0	0	1	0	0
truly	NONE	truly	O	0	0	0	1	0	0
aware	NONE	aware	O	0	0	0	1	0	0

Table 1. **Alignment Example**: Our method, $\mathcal{A}$, is able to align $t_{disfluent}$ and t_{Φ} for $\mathcal{E}$- and $\mathcal{Z}$-scoring, in contrast to previous methods which simply perform sequence classification (t_{CLS}) to allow scoring. Hallucinated tokens (*) are filtered before scoring, ensuring they do not affect results, allowing continuity with previous evaluation methodology.

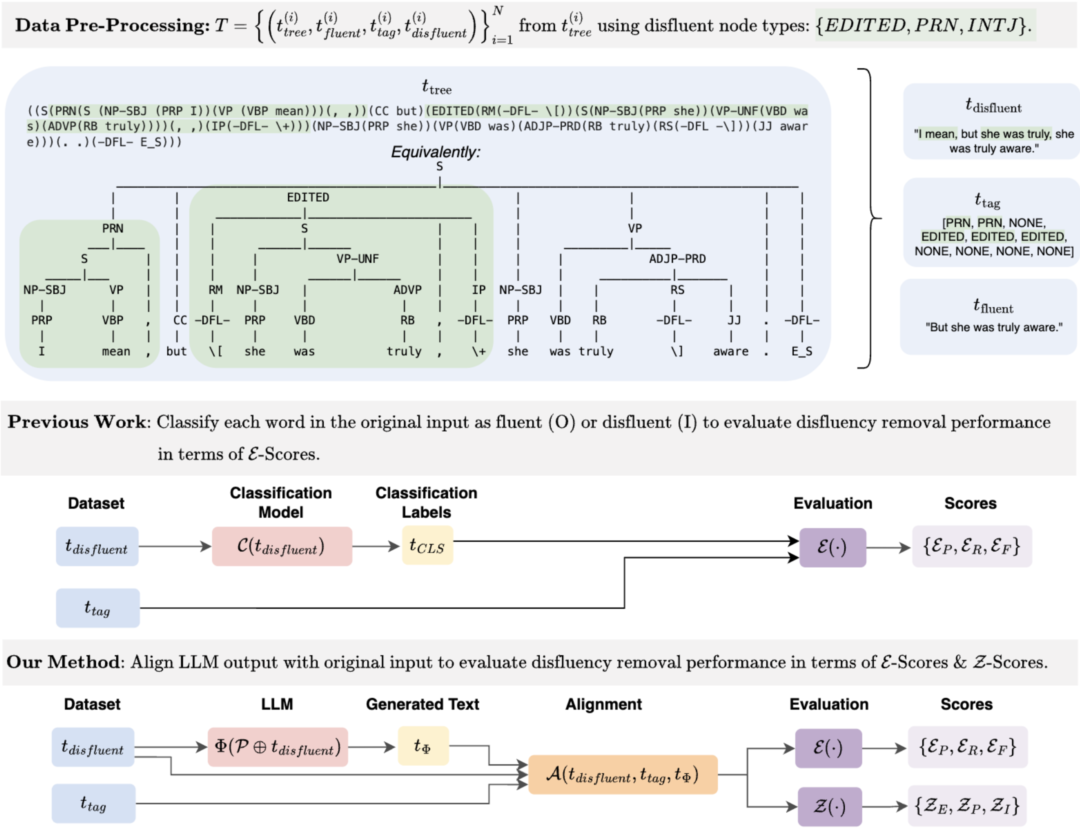
Earlier works use methods such as LSTM, transition modeling, and noisy channel modeling for sequence classification [12, 13, 14, 15, 16, 17, 18, 19, 20]. Additionally, some of these models force alignment with the original sequence via constrained decoding. While this method enables calculation of $\mathcal{E}$-Scores, it can also reduce the expressive power of the GM by reducing the search space during decoding.

2.3. The Alignment Problem

Two previous works have attempted to address the problem of aligning generated text with the original disfluent text. (i) [10] perform this alignment with *longest common subsequence (LCS)*. However, this approach is suboptimal, as we will show in §3.1. In contrast, [21] – who also use a translation-based model – avoid the alignment, instead evaluating their performance using the BLEU and ROUGE metrics. These types of n-gram-based metrics ignore alignment, meaning the fine-grained parse tree information about where specific disfluencies are removed is lost. (ii) [22] (who also point out this problem with the evaluation of [21]) propose a statistical weighting method based on Gestalt pattern matching [23] to align the generated text with the original disfluent text. However, this method is not deterministic, and therefore lacks the alignment guarantee that we are able to obtain with our alignment method (detailed in §3.1).

3. $\mathcal{Z}$-SCORE FRAMEWORK

Figure 2 shows our framework for evaluating GMs on the disfluency removal task. We introduce $\mathcal{Z}$-scoring, a span-level metric for quantifying the types (i.e. EDITED, INTJ, PRN) of disfluencies each model is able to remove. We design an alignment module to enable both traditional $\mathcal{E}$-scoring and this span-level $\mathcal{Z}$-scoring for GMs.



Data Pre-Processing: $T = \left\{\left(t^{(i)}_{tree}, t^{(i)}_{fluent}, t^{(i)}_{tag}, t^{(i)}_{disfluent}\right)\right\}_{i=1}^{N}$ from $t^{(i)}_{tree}$ using disfluent node types: $\{EDITED, PRN, INTJ\}$.
t_{tree}
((S(PRN(S (NP-SBJ (PRP I))(VP (VBP mean)))(, ,))(CC but)(EDITED(RM(-DFL- \[))(S(NP-SBJ(PRP she))(VP-UNF(VBD was)(ADVP(RB truly))))(, ,)(IP(-DFL- \+)))(NP-SBJ(PRP she))(VP(VBD was)(ADJP-PRD(RB truly)(RS(-DFL -\]))(JJ aware)))(. .)(-DFL- E_S)))
Equivalently:
S
PRN
S
NP-SBJ
VP
PRP
VBP
I
mean
,
CC
but
EDITED
S
VP-UNF
RM
NP-SBJ
-DFL-
PRP
VBD
\[
she
was
ADVP
RB
truly
,
IP
-DFL-
+
NP-SBJ
PRP
she
VP
VBD
was
ADJP-PRD
RB
truly
RS
-DFL-
\]
JJ
aware
.
-DFL-
E_S
$t_{\text{disfluent}}$
"I mean, but she was truly, she was truly aware."
t_{tag}
[PRN, PRN, NONE, EDITED, EDITED, EDITED, NONE, NONE, NONE, NONE]
t_{fluent}
"But she was truly aware."
Previous Work: Classify each word in the original input as fluent (O) or disfluent (I) to evaluate disfluency removal performance in terms of $\mathcal{E}$-Scores.
Dataset
Classification Model
Classification Labels
Evaluation
Scores
$t_{disfluent}$
$\mathcal{C}(t_{disfluent})$
t_{CLS}
$\mathcal{E}(\cdot)$
$\{\mathcal{E}_P, \mathcal{E}_R, \mathcal{E}_F\}$
t_{tag}
Our Method: Align LLM output with original input to evaluate disfluency removal performance in terms of $\mathcal{E}$-Scores & $\mathcal{Z}$-Scores.
Dataset
LLM
Generated Text
Alignment
Evaluation
Scores
$t_{disfluent}$
$\Phi(\mathcal{P} \oplus t_{disfluent})$
t_Φ
$\mathcal{A}(t_{disfluent}, t_{tag}, t_\Phi)$
$\mathcal{E}(\cdot)$
$\{\mathcal{E}_P, \mathcal{E}_R, \mathcal{E}_F\}$
t_{tag}
$\mathcal{Z}(\cdot)$
$\{\mathcal{Z}_E, \mathcal{Z}_P, \mathcal{Z}_I\}$

Fig. 2. $\mathcal{Z}$**-Score Framework**: Previous work treated the disfluency removal task as a classification task. In contrast, we design an alignment module, $\mathcal{A}$, to allow GMs to be used for the disfluency removal task. This module aligns generated text with disfluent text to enable the $\mathcal{E}$- and $\mathcal{Z}$-scoring of GMs, Φ, allowing category-specific performance analysis and supporting targeted modeling strategies.

3.1. Alignment Module ($\mathcal{A}$)

Prior alignment strategies such as LCS, BLEU/ROUGE-based evaluation, and statistical weighting methods either produce systematic errors (e.g., LCS preserves disfluent tokens), fail to capture span-level disfluencies (BLEU/ROUGE), or lack determinism (statistical weighting). In contrast, our deterministic alignment module $\mathcal{A}$ ensures reliable alignment.

The alignment module is responsible for constructing the $t_{disfluent}$, t_{tag}, t_Φ columns which are used by $\mathcal{E}(\cdot)$ and $\mathcal{Z}(\cdot)$. Hence, we first tokenize using TreebankWordTokenizer. We then use a variation of **Gestalt matching [23],** $\mathcal{G}$, to align the tokens in the generated, t_Φ, and ground-truth, $t_{disfluent}$, sequences for comparison. t_{tag} is metadata for $t_{disfluent}$, as it contains sequence information pertaining to the set of disfluent tags: {EDITED, PRN, INTJ}.

However, $\mathcal{G}$ cannot be straightforwardly applied, as it performs *early matching* – e.g. for input tuples in the form $(t_{disfluent}, t_{tag}, t_\Phi)$, $\mathcal{G}$ **incorrectly yields**:

$$[(\text{the}, \textbf{EDITED}, \textbf{the}), (\text{the}, \text{NONE}, \emptyset), (\text{cat}, \text{NONE}, \emptyset)]$$

To fix this problem, we modify $\mathcal{G}$ to create $\mathcal{A}$ for alignment. We append a special token and the tag (e.g. *"the§EDITED"*) to disfluent tokens to form t'_Φ, before running $\mathcal{G}(t_{disfluent}, t'_\Phi)$. This step forces disfluent tokens into the *replace* case of $\mathcal{G}$, where we match to valid NONE tags before valid disfluent tags. Hence, $\mathcal{A}$ **correctly yields**:

$$[(\text{the}, \text{EDITED}, \emptyset), (\text{the}, \textbf{NONE}, \textbf{the}), (\text{cat}, \text{NONE}, \emptyset)]$$

We show an example of this alignment in Table 1, using the example from Figure 2. The character $*$ marks hallucinated tokens produced by the GM, which we treat as an artifact of GMs and explicitly remove during post-processing,[1] ensuring that this content does not propagate to downstream tasks.

3.2. $\mathcal{E}$-Scores

The $\mathcal{E}$-Score function:

$$\mathcal{E}(\mathcal{A}(t_{disfluent}, t_{tag}, t_\Phi)) \rightarrow \{\mathcal{E}_F, \mathcal{E}_P, \mathcal{E}_R\}$$

[1] Hence, $\mathcal{A}$ also facilitates a straightforward filtering step that eliminates hallucinated tokens entirely from the final system output.

returns a set of scores for the word-based F1, precision, and recall scores for the disfluency removal task using the $\mathcal{A}$ alignment. Previous work measured performance in terms of $\mathcal{E}$-Scores. Looking to Table 1, $\mathcal{E}$-Scores can only be calculated given a correct alignment between $t_{disfluent}$ and t_Φ, which we obtain with our alignment module, $\mathcal{A}$. We demonstrate calculating $\mathcal{E}$-Scores with Table 1. From the alignment $\mathcal{A}$, $\mathbb{1}_{gt}$ and $\mathbb{1}_{pred}$ are calculated by comparing t_Φ to t_{tag}:

- $\mathbb{1}_{gt}$: *Based on the ground truth parse tree, should this word be removed by* Φ*?*
- $\mathbb{1}_{pred}$: *Was this word actually removed by* Φ*?*

Then, from $\mathbb{1}_{\{tp,fn,tn,fp\}}$, the $\mathcal{E}$-Scores are calculated:

- $\mathcal{E}_P = \frac{\Sigma_{tp}}{\Sigma_{tp}+\Sigma_{fp}} = \frac{3}{3+1} \rightarrow 75.0$
- $\mathcal{E}_R = \frac{\Sigma_{tp}}{\Sigma_{tp}+\Sigma_{fn}} = \frac{3}{3+2} \rightarrow 60.0$
- $\mathcal{E}_F = \frac{2 \cdot \mathcal{E}_P \cdot \mathcal{E}_R}{\mathcal{E}_P + \mathcal{E}_R} = \frac{2 \cdot 0.75 \cdot 0.60}{0.75+0.60} \rightarrow 66.0$

$\mathcal{E}_P$ penalizes over-deletion, hence $\mathcal{E}_P$ is low when fluent tokens are incorrectly removed. Recall that fp indicates Φ should not have removed the token, but did. In contrast, $\mathcal{E}_R$ penalizes under-deletion, hence $\mathcal{E}_R$ is low when ground-truth disfluencies remain in the output. Recall that fn indicates Φ should have removed the token, but didn't. Then, $\mathcal{E}_F$ is the harmonic mean of $\mathcal{E}_P$ and $\mathcal{E}_R$ and the main indicator of overall disfluency removal performance; it is low if either $\mathcal{E}_P$ or $\mathcal{E}_R$ is low.

3.3. $\mathcal{Z}$-Scores

We propose $\mathcal{Z}$-Scores, a new span-level metric for the disfluency removal task. $\mathcal{Z}$-Scores are more informative in terms of model performance on individual disfluency types. We use this new metric to examine the type of disfluencies each model removes well, or struggles to remove. The $\mathcal{Z}$-Score function:

$$\mathcal{Z}(\mathcal{A}(t_{disfluent}, t_{tag}, t_\Phi)) \rightarrow \{\mathcal{Z}_E, \mathcal{Z}_I, \mathcal{Z}_P\}$$

returns a set of scores for the percentage of EDITED, PRN, and INTJ nodes that the model was successfully able to remove using the $\mathcal{A}$ alignment:

- $\mathcal{Z}_E = \frac{\mathbb{1}_{gt} \wedge (w_{tag}=EDITED) \wedge \mathbb{1}_{pred}}{\mathbb{1}_{gt} \wedge (w_{tag}=EDITED)} = \frac{3}{3} \rightarrow 100\%$
- $\mathcal{Z}_I = \frac{\mathbb{1}_{gt} \wedge (w_{tag}=INTJ) \wedge \mathbb{1}_{pred}}{\mathbb{1}_{gt} \wedge (w_{tag}=INTJ)} = \frac{0}{0} \rightarrow NaN$
- $\mathcal{Z}_P = \frac{\mathbb{1}_{gt} \wedge (w_{tag}=PRN) \wedge \mathbb{1}_{pred}}{\mathbb{1}_{gt} \wedge (w_{tag}=PRN)} = \frac{0}{2} \rightarrow 0\%$

Because t_{tag} is constructed using a top-down recursive approach, the tags are span-level. Hence, $\mathcal{Z}$-Scores can be considered to be a span-level metric, whilst $\mathcal{E}$-Scores can be considered to only be a word-level metric.

4. A CASE STUDY: METAPROMPTING

We conduct a small-scale study with metaprompting [24] to illustrate the utility of our proposed metric. Experiments are conducted on the Switchboard dataset [25], though our method is generalizable to other corpora. We use gpt-4o-mini as a representative GM, with results shown in Table 2.

gpt-4o-mini							
M		$\mathcal{E}_F$	$\mathcal{E}_P$	$\mathcal{E}_R$	$\mathcal{Z}_E$	$\mathcal{Z}_I$	$\mathcal{Z}_P$
s	$\mathcal{P}_0$	$72.69_{5.79}$	$75.61_{7.05}$	$70.48_{7.35}$	$85.20_{8.23}$	$61.89_{11.08}$	$65.02_{20.99}$
s	$\mathcal{P}_1$	$81.94_{3.75}$	$84.47_{4.92}$	$79.90_{5.65}$	$83.67_{9.27}$	$78.28_{8.10}$	$74.86_{22.06}$
s	$\mathcal{P}_2$	$79.86_{5.42}$	$76.88_{7.02}$	$83.52_{6.12}$	$87.45_{7.48}$	$79.60_{8.89}$	$87.09_{15.46}$

Table 2. ***Metaprompting*** *(mean*$_{\text{std. dev}}$*): Incorporating short prompts with common disfluencies (*$\mathcal{P}_1, \mathcal{P}_2$*) improves performance. While* $\mathcal{E}$*-Scores suggest modest overall gains,* $\mathcal{Z}$*-Scores reveal that these improvements are primarily driven by better INTJ and PRN removal, highlighting the* ***diagnostic value*** *of our proposed metric.*

We start with our baseline prompt for disfluency removal, $\mathcal{P}_0$. $\mathcal{P}_0$ achieves reasonable overall $\mathcal{E}$-Scores, but $\mathcal{Z}$-Scores reveal clear weaknesses: the model handles EDITED disfluencies well ($\mathcal{Z}_E$=85.20), yet performs poorly on INTJ (interjections such as *uh*, *um*, with $\mathcal{Z}_I$=61.89) and PRN (parentheticals such as *I mean*, with $\mathcal{Z}_P$=65.02). These modeling deficiencies are hidden when looking only at $\mathcal{E}$-Scores.

In comparison, metaprompts $\mathcal{P}_1$ and $\mathcal{P}_2$ include explicit examples of INTJ and PRN disfluencies. For these prompts, the $\mathcal{Z}$-Scores show marked improvements: $\mathcal{Z}_I$ increases approximately 16 points, while $\mathcal{Z}_P$ rises approximately 9 points. In contrast, $\mathcal{Z}_E$ remains stable, confirming that the gains stem specifically from better handling of INTJ and PRN.

Importantly, $\mathcal{Z}$-Scores make this diagnostic insight possible. **Whereas $\mathcal{E}$-Scores provide only an aggregate view of precision and recall, $\mathcal{Z}$-Scores reveal that performance gains are localized to the specific linguistic phenomenon of INTJ and PRN disfluencies.** This illustrates the value of $\mathcal{Z}$-Scores as a fine-grained diagnostic tool for understanding model behavior and driving model improvements.

Future directions include the development of category-specific prompting schemes, architectural innovations (e.g. disfluency category specialized adapters), and tailored augmentation pipelines, with $\mathcal{Z}$-Scores serving as the central mechanism for diagnosis-driven model refinement.

5. CONCLUSION

We introduced $\mathcal{Z}$-Scores, a span-level evaluation metric that complements traditional $\mathcal{E}$-Scores for disfluency removal with linguistic assessment based on specific disfluency categories (EDITED, INTJ, PRN). A case study with LLMs illustrates how $\mathcal{Z}$-Scores reveal differences in handling specific disfluency types (INTJ, PRN) and how $\mathcal{Z}$-Scores can be used to drive model performance improvements via targeted prompting. We release our metric as an open source Python package, providing the community with a standardized resource for future research.

6. REFERENCES

[1] Elizabeth Shriberg, *Preliminaries to a Theory of Speech Disfluencies*, Ph.D. thesis, 1994.

[2] Maria Teleki, Lingfeng Shi, Chengkai Liu, and James Caverlee, "I want a horror – comedy – movie: Slips-of-the-Tongue Impact Conversational Recommender System Performance," in *INTERSPEECH*, 2025.

[3] Fabian Retkowski, Maike Züfle, Andreas Sudmann, Dinah Pfau, Shinji Watanabe, Jan Niehues, and Alexander Waibel, "Summarizing speech: A comprehensive survey," 2025.

[4] Dominik Wagner, Sebastian P Bayerl, Ilja Baumann, Korbinian Riedhammer, Elmar Noth, and Tobias Bocklet, "Large language models for dysfluency detection in stuttered speech," *arXiv preprint arXiv:2406.11025*, 2024.

[5] Shaolei Wang, Zhongyuan Wang, Wanxiang Che, Sendong Zhao, and Ting Liu, "Combining self-supervised learning and active learning for disfluency detection," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 3, pp. 1–25, 2021.

[6] Qianqian Dong, Feng Wang, Zhen Yang, Wei Chen, Shuang Xu, and Bo Xu, "Adapting translation models for transcript disfluency detection," in *AAAI and IAAI and AAAI Symposium on Educational Advances in Artificial Intelligence*, 2019.

[7] Jianbang Ding, Suiyun Zhang, and Dandan Tu, "Disfluency detection for real-world scenarios," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–9.

[8] Zhenrong Cheng, Jiayan Guo, Hao Sun, and Yan Zhang, "Boosting disfluency detection with large language model as disfluency generator," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.

[9] Shaolei Wang, Zhongyuan Wang, et al., "Combining self-training and self-supervised learning for unsupervised disfluency detection," in *EMNLP*, 2020.

[10] Rohan Chaudhury, Maria Teleki, Xiangjue Dong, and James Caverlee, "DACL: Disfluency augmented curriculum learning for fluent text generation," in *LREC-COLING*, May 2024, pp. 4311–4321.

[11] Paria Jamshid Lou and Mark Johnson, "Improving disfluency detection by self-training a self-attentive model," in *ACL*, 2020, pp. 3754–3763.

[12] Jingfeng Yang, Diyi Yang, and Zhaoran Ma, "Planning and generating natural and diverse disfluent texts as augmentation for disfluency detection," *EMNLP*, p. 1450–1460, 2020.

[13] Paria Jamshid Lou, Yufei Wang, and Mark Johnson, "Neural constituency parsing of speech transcripts," in *NAACL*, 2019.

[14] Nguyen Bach and Fei Huang, "Noisy bilstm-based models for disfluency detection.," in *INTERSPEECH*, 2019, pp. 4230–4234.

[15] Feng Wang, Wei Chen, Zhen Yang, Qianqian Dong, Shuang Xu, and Bo Xu, "Semi-supervised disfluency detection," in *International Conference on Computational Linguistics*, 2018, pp. 3529–3538.

[16] Paria Jamshid Lou and Mark Johnson, "Disfluency detection using a noisy channel model and a deep neural language model," *ACL (Short Paper)*, p. 547–553, 2017.

[17] Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi, "Disfluency detection using a bidirectional LSTM," in *INTERSPEECH*, 2016.

[18] Mark Johnson and Eugene Charniak, "A TAG-based noisy-channel model of speech repairs," in *ACL*, 2004, pp. 33–39.

[19] Eugene Charniak and Mark Johnson, "Edit Detection and Parsing for Transcribed Speech," *NAACL*, 2001.

[20] Donald Hindle, "Deterministic parsing of syntactic nonfluencies," *ACL*, p. 123–128, 1983.

[21] Elizabeth Salesky, Matthias Sperber, and Alex Waibel, "Fluent translations from disfluent speech in end-to-end speech translation," *arXiv*, 2019.

[22] Paria Jamshid Lou and Mark Johnson, "End-to-end speech recognition and disfluency removal," in *EMNLP Findings*, 2020, pp. 2051–2061.

[23] John W. Ratcliff and David E. Metzener, "Pattern Matching: The Gestalt Approach," 1988.

[24] Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao, "Meta prompting for ai systems," *arXiv preprint arXiv:2311.11482*, 2023.

[25] John J Godfrey, Edward C Holliman, and Jane McDaniel, "Switchboard: Telephone speech corpus for research and development," in *ICASSP*, 1992, vol. 1, pp. 517–520.