I want a horror – comedy – movie: Slips-of-the-Tongue Impact Conversational Recommender System Performance

Maria Teleki, Lingfeng Shi, Chengkai Liu, James Caverlee

Department of Computer Science & Engineering, Texas A&M University, USA

{mariateleki,lingfengs111,liuchengkai,caverlee}@tamu.edu

Abstract

Disfluencies are a characteristic of speech. We focus on the impact of a specific class of disfluency – whole-word speech substitution errors (WSSE) – on LLM-based conversational recommender system performance. We develop Syn-WSSE, a psycholinguistically-grounded framework for synthetically creating genre-based WSSE at varying ratios to study their impact on conversational recommender system performance. We find that LLMs are impacted differently: llama and mixtral have improved performance in the presence of these errors, while gemini, gpt-4o, and gpt-4o-mini have deteriorated performance. We hypothesize that this difference in model resiliency is due to differences in the pre- and post-training methods and data, and that the increased performance is due to the introduced genre diversity. Our findings indicate the importance of a careful choice of LLM for these systems, and more broadly, that disfluencies must be carefully designed for as they can have unforeseen impacts.

Index Terms: conversational recommender systems, whole-word substitution speech errors, LLM, disfluency

1. Introduction

Large language models (LLMs) and recommender systems are shifting towards being used in naturalistic settings. Consider conversational systems [1, 2], including ChatGPT [3] and Gemini [4] voice modes, which encourage users to ask questions and seek recommendations. The advance of these systems is enabled in large part by efficient, high-performance ASR systems such as Whisper(X) [5, 6]. However, most datasets used for conversational search and recommendation were designed around written text [7, 8, 9, 10] and therefore lack the disfluencies that occur in natural speech – i.e. *um*, *uh*, sentence restarts, backchannels, corrections, and more [11, 12, 13, 14, 15, 16].

Current synthetic disfluency augmentation methods generally focus on disfluencies of type repetition, replacement, and restart [17, 18]. We instead focus on whole-word substitution speech errors (WSSE), a particularly critical type of error for conversational recommender systems (CRS). Consider the impact of WSSE in the following request: *"I want a horror comedy movie."* The intended genre is *"comedy,"* however, in the presence of the WSSE *"horror,"* the communicated desired genre is obscured, in that recommended movies for the genre *comedy* are expected to be very different from recommended movies for the genre *horror comedy*. In this way, this natural form of disfluency in a user's speech can impact CRS performance.

Hence, we propose Syn-WSSE, a synthetic disfluency augmentation method focusing on critical WSSE genre errors for CRS. Syn-WSSE is grounded in psycholinguistic literature [13, 11], and allows us to isolate the WSSE disfluency signal and study its impact on CRS performance over a set of widely-used LLM backbone models [19, 4, 20, 21]. We make three contributions:

1. We propose Syn-WSSE (pronounced *sin wise*), a psycholinguistic-grounded synthetic data augmentation method for semantic word speech substitution errors (WSSE), specialized for CRS in the movie recommendation domain.
2. We perform the first psycholinguistically-grounded analysis of the impact of WSSE on CRS using Syn-WSSE, finding that this class of disfluency impacts CRS performance to various degrees – depending on the amount of disfluency and the choice of LLM backbone. We find that `llama` and `mixtral` have improved performance in the presence of WSSE, while `gemini`, `gpt-4o`, and `gpt-4o-mini` have deteriorated performance in the presence of WSSE. We hypothesize that the WSSE may be introducing novel and diverse user interests [22, 23, 24, 25] which `llama` and `mixtral` are able to take advantage of due to their pre-training and post-training data and methods. This difference in performance indicates the need for a careful choice of LLM backbone in conversational recommender systems.
3. We release our code and augmented version of the INSPIRED [7] dataset at `https://github.com/mariateleki/Syn-WSSE`.

2. Preliminaries

Whole-Word Substitution Speech Errors Disfluencies – e.g., *um*, *uh*, sentence restarts, and more – are a natural part of speech. Disfluencies occur in part because speech planning has many stages, and WSSE can happen at one or more of these stages [16]. The Shriberg disfluency definition [11] breaks disfluent events into 3 key regions: the reparandum, the interregnum, and the repair. The reparandum and the interregnum are removed to form a fluent sentence from a disfluent one. For example, in the sentence, *"I want a horror comedy movie,"* the reprandum is *horror*, the interregnum is empty (often this section contains insertion-type disfluencies such as uh, um, and other tokens), and the repair is *comedy*. Hence, the fluent sentence is: *"I want a comedy movie,"* as this is what the speaker intends to say (often referred to as *intended speech*). Note that [11] does not distinguish between the subtypes of semantic repairs (detailed in Section 2.3.3.3 of [11]): i.e. error repairs (when the speaker corrects an error, such as in the *hot/cold* example) and appropriateness repairs (e.g., "up" versus "straight up" to help clarify what is meant by "up"). WSSE are a specific class of disfluency, understood as pairs of an error word and a target word: (e_i, t_i) [13]. For example, in *It was hot cold today*,

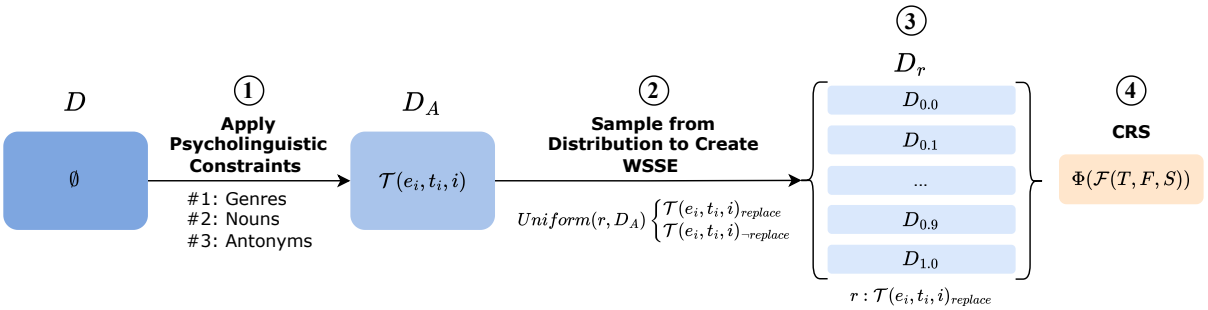

Figure 1: *Syn-WSSE Framework:* ① *We apply psycholinguistic constraints to create the Candidate-WSSEs – i.e.,* $\mathcal{T}(e_i, t_i, i)$ *– in* D_A*.* ② *We draw* r *percent of samples uniformly from the list of Candidate-WSSEs tokens to create WSSE.* ③ D_r *are created and Syn-WSSE is complete.* ④ *We evaluate the performance of the CRS (§3.2) on each* D_r*, shown in Tables 1, 2, and 3.*

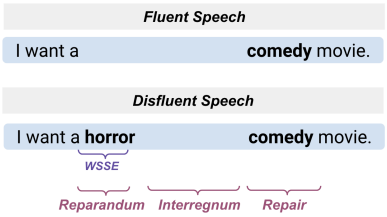

Figure 2: *WSSE in the Shriberg disfluency definition [11].*

e_i = *hot* and t_i = *cold*, because the speaker intended to say *cold*, but accidentally said *hot*, and thus followed up their error with the correct term. We focus on semantic WSSE as opposed to phonological WSSE – i.e. WSSE which are related phonetically. As detailed in Section 3.1, we construct our Syn-WSSE method for synthetic WSSE augmentation based on the findings of [13].

Conversational Recommender Systems Since the advent of the transformer era [26, 27], LLMs have become capable and popularized. Consider models such as OpenAI's ChatGPT [28, 19], Google's Gemini [4], and Meta's Llama [20]. Conversational search and recommendation are considered downstream tasks for a base LLM. Recent recommender system approaches for incorporating LLMs include: aligning the LLM latent space with the collaborative latent space [29, 30] or the sequential latent space [31], or taking a zero-shot approach [32, 33]. We focus in this work on the zero-shot setting [32], as it most exposes the inherent capabilities of the LLM with respect to joint disfluency understanding and recommendation.

3. Methodology

In this section, we introduce our methodology for Syn-WSSE (§3.1) and for evaluating the impact of WSSE on CRS with a set of widely-used base-LLMs (§3.2).

3.1. Syn-WSSE: A Framework for Synthetic Generation of Whole-Word Substitution Speech Errors

In order to understand the impact of semantic WSSE [13] on conversational recommender systems, we construct a *psycholinguistically-grounded method* – Syn-WSSE – to construct datasets D_r with varying ratios, r, of synthetic WSSE as shown in Figure 1. For $r \in \{0.0, 0.1, ..., 1.0\}$, we uniformly randomly select r percent of all Candidate-WSSEs which meet the below constraints to synthetically place in the dataset, leaving us with 11 datasets of varying WSSE concentration to conduct our experiments. This allows us to stress-test the CRS' performance in the presence of WSSE. To identify Candidate-WSSEs in our dataset, we follow a few key constraints based on the findings of [13]:

Constraint #1: WSSE Are Genres. We impose the constraint that WSSE should occur only on *genres* for two reasons: (i) [13] find that 82% of WSSE in their analysis were *shared-feature substitutions*, e.g., e_i = *horror* and t_i = *comedy*, versus *associative substitutions*, like e_i = *JLaw* (short for Jennifer Lawrence, a famous actress) and t_i = *comedy*. (ii) We hypothesize that genre errors are most likely to impact the recommendation quality [24]. Using $\mathcal{F}$ =`gpt-4o-mini`, we obtain the genres for a given text using $p_{GetGenres}$ as shown in Figure 3. We notice that sometimes `gpt-4o-mini` hallucinates genres when none are directly present in the text – e.g. *User: I like the movie Die Hard. Response: Action*. Hence, we place a rule to ensure that genres are present in the text.

Constraint #2: WSSE Are Nouns. [13] find that 99.6% of semantic WSSE errors, e_i, are broadly the same part-of-speech as the target, t_i. [13] also find that 54.6% of WSSE occur when target words are nouns. Hence, we use the neural parsing model benepar_en3 from [34] to identify the part-of-speech for candidate Syn-WSSE via SpaCy.[1] This model was trained and evaluated on the Switchboard dataset [35], and tags for this model are located in Table 1.1 of [36]. We consider the following noun-based tags: NN (noun, singular or mass), NNS (noun, plural), NNP (proper noun, singular), and NNPS (proper noun, plural). For example, in the following sentence *"The movies are very* ***campy****, but are feel good..."* campy is an adjective, so it does not meet our constraints for becoming a Candidate-WSSE.

Constraint #3: WSSE Are Antonyms. [13] find – confirming [37] – that the error word, e_i, in WSSE is often an antonym of the target word, t_i. [13] states that 24.9% of the *shared-feature errors* in their corpus are antonyms, and that "this might well be an underestimate." Using $\mathcal{F}$ =`gpt-4o-mini`, we use $p_{OppositeGenre}$ as shown in Figure 3, to obtain the opposite genre for each Candidate-WSSE. For example, the opposite of *comedy* is *horror*. We recognize that there are multiple possible opposite genres, and leave the prioritization to the $\mathcal{F}$ model.

Creation of D_r Datasets via Candidate-WSSE We use these three constraints to create the Candidate-WSSEs, which

[1] https://spacy.io/universe/project/self-attentive-parser

$p_{GetGenres}$

If there are no genres present in the text, respond with "None". Otherwise, list the genres mentioned in the text exactly as they appear in the following text: $\{S\}$.

$p_{OppositeGenre}$

With no other text, list only a genre which is the opposite of the following genre: $\{g\}$.

p_{CRS}

Pretend you are a movie recommender system.\n I will give you a conversation between a user and you (a recommender system). Based on the conversation, you reply me with 20 recommendations without extra sentences.\n Here is the conversation: $\{S\}$

Figure 3: *Prompts used for Syn-WSSE (§3.1) and CRS (§3.2).*

are special tokens, $\mathcal{T}(e_i, t_i, i)$, containing the target word, t_t, the error word, e_i, and a unique index for that token, i. This intermediate form allows us to uniformly randomly sample a percentage, $r \in \{0.0, 0.1, ..., 1.0\}$ of these special tokens to replace to create the synthetic WSSE: $\mathcal{T}(e_i, t_i, i)_{replace} \leftarrow e_i \oplus \square \oplus t_i$, e.g., *"horror comedy."* The rest of the Candidate-WSSE (those that are not sampled) are not replaced: $\mathcal{T}_{\neg replace} \leftarrow t_i$, e.g., *"comedy."*

Why not instead learn a Syn-WSSE distribution from data? There are two key reasons: (i) There is a lack of dataset availability from which to learn a distribution in the general dialogue context and the specific CRS context [35, 10, 18]; we view this as a direction for future work as these datasets are curated, enabled by improvements in automatic speech recognition systems in terms of disfluency transcription capabilities [38]. (ii) Our method is based on findings from the field of psycholinguistics that consider the causal mechanisms behind the findings (§3.1). Hence, this approach is more broadly generalizable than the learned distributions from limited datasets – i.e. Switchboard, SIMMC2, and Fisher English [35, 10, 18].

3.2. LLM-Based Zero-Shot CRS

We use the setup of [32]. Each conversation, $c \in C$, has a ground truth item, g_C. The LLM $\mathcal{F}$, takes in a task description template T, format requirement F, and the conversational context S. We use p_{CRS} as shown in Figure 3, combining T and F, and appending S. Then, the output is $\mathcal{F}(T, F, S)$. To evaluate the output natural language recommendation list from the LLM, we apply a post-processor Φ to convert it to a ranked list L_c. The process can be described as follows:

$$L_C = \Phi(\mathcal{F}(T, F, S))$$

We measure CRS performance in terms of traditional recommender systems metrics, adapted for the conversational setting via the post-processor Φ in the CRS pipeline. For all three metrics, the individual scores are averaged for $c \in C$, for $k \in \{5, 10\}$ where k indicates the length of L_c. The rank of g_c in $(L_c)_0^k$ is $r(g_c)$, and $\mathbb{I}(\cdot)$ is the binary indicator function. All three metrics are fractions $\in [0, 1]$:

- **Mean Recall@k** $= \mathbb{E}_c \left[\mathbb{I}\{g_c \in (L_c)_0^k\}\right]$
  This metric aggregates the presence of g_c in $(L_c)_0^k$.

	RECALL@5						
	RECALL@$5_{0.0}$	$\Delta_{0.1}$	$\Delta_{0.2}$	$\Delta_{0.3}$	$\Delta_{0.5}$	$\Delta_{0.7}$	$\Delta_{1.0}$
`llama`	0.104	18.18	18.18	4.55	13.64	9.09	13.64
`mixtral`	<u>0.081</u>	11.76	11.76	0.00	5.88	5.88	5.88
`gemini`	0.090	-10.53	-10.53	-10.53	-21.05	-10.53	-5.26
`gpt-4o`	**0.133**	-14.29	-10.71	0.00	-7.14	-17.86	-7.14
`gpt-4o-mini`	0.114	0.00	-4.17	-4.17	-16.67	-16.67	-25.00
	RECALL@10						
	RECALL@$10_{0.0}$	$\Delta_{0.1}$	$\Delta_{0.2}$	$\Delta_{0.3}$	$\Delta_{0.5}$	$\Delta_{0.7}$	$\Delta_{1.0}$
`llama`	0.152	3.13	18.75	9.38	12.50	-3.13	12.50
`mixtral`	0.133	0.00	-3.57	0.00	0.00	0.00	0.00
`gemini`	<u>0.128</u>	0.00	-3.70	-11.11	3.70	-7.41	-3.70
`gpt-4o`	**0.199**	-7.14	-4.76	-2.38	-14.29	-21.43	-14.29
`gpt-4o-mini`	0.185	-2.56	-7.69	-10.26	-25.64	-23.08	-30.77

Table 1: ***Best***, <u>*Worst;*</u> *Most Resilient* , *Least Resilient*; Δ_r *is percent change in Recall@k for* $D = r$ *and* $D = 0.0$.

- **Mean NDCG@k** $= \mathbb{E}_c \left[\mathbb{I}\{g_c \in (L_c)_0^k\} \cdot \frac{1}{\log_2(r(g_C)+1)}\right]$
  This metric aggregates the ranked presence of g_c in $(L_c)_0^k$, discounting lower-ranked g_c occurrences.
- **Mean MRR@k** $= \mathbb{E}_c \left[\mathbb{I}\{g_c \in (L_c)_0^k\} \cdot \frac{1}{r(g_c)+1}\right]$
  This metric aggregates the ranked presence of g_c in $(L_c)_0^k$.

We apply the following filters: (i) As part of the post-processor, Φ, we count recommended items which are within an edit distance of 2 from the ground truth items as correct. (ii) We exclude conversations where previously-mentioned items are recommended, e.g. *S: I want something with drama. R: How about The Theory of Everything? S: No, it should be based in the USA. R: Ok, how about Spider-Man: Across the Spider-Verse? S: No, it should be realistic. R: R: How about The Theory of Everything?* (iii) We only consider turns at which the recommendation is made.

INSPIRED Dataset For our experiments, we utilize the INSPIRED dataset [7, 32]. In this dataset, pairs of crowdworkers recommend movies to each other, with one acting as "seeker" and the other acting as "recommender" to simulate a user interacting with a CRS. We select 228 dialogues and conduct experiments on the selected INSPIRED dataset. Detailed in the next section, we apply our synthetic WSSE process to create multiple datasets $D_r : r \in \{0.0, 0.1, ..., 1.0\}$, where r is the percentage of synthetic WSSE present.

Large Language Models We investigate two types of LLMs in our study: API-based, and open-source. These LLMs serve as $\mathcal{F}$, the large language model backbone for the conversational recommender system. We generally opt to use smaller, distilled versions of the models for compatibility with edge devices. Open-sourced LLMs are runnable via a local machine, as the organization releases the model parameters. We use `Llama3-8B-Instruct`[2] [20], and `Mixtral-8x7B-Instruct-v0.1` [21] with 4-bit quantization. For both, we set the temperature to 1×10^{-3}, the max tokens to 512, and run on a server with four 24GB NVIDIA A5000 GPUs. API-based LLMs are only accessible via an API, as the organization does not release the model parameters. We use `gemini-1.5-flash-8b` [4]. We set the temperature to 0.1, and the max tokens to 512. We notice that for a small number of requests, the API responds with *Finish Reason 4*: "Meaning that the model was reciting from copyrighted material." In these cases, we follow a 2-step process to resolve the requests: (i) we prepend "Do not respond with any copyrighted material. " to the prompt. (ii) If that does not resolve the issue, we exclude the sample (this occurs in $< 1\%$ of cases). We use

[2]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

NDCG@5							
	NDCG@$5_{0.0}$	$\Delta_{0.1}$	$\Delta_{0.2}$	$\Delta_{0.3}$	$\Delta_{0.5}$	$\Delta_{0.7}$	$\Delta_{1.0}$
llama	0.076	17.58	16.77	7.04	13.65	7.43	12.57
mixtral	0.051	6.81	7.63	2.63	6.40	2.92	8.95
gemini	0.056	-8.21	-13.00	-11.90	-17.14	-1.94	-6.48
gpt-4o	**0.105**	-13.20	-13.71	-6.87	-12.36	-16.80	-14.02
gpt-4o-mini	0.083	3.85	-4.86	-4.71	-20.74	-12.99	-25.06
NDCG@10							
	NDCG@$10_{0.0}$	$\Delta_{0.1}$	$\Delta_{0.2}$	$\Delta_{0.3}$	$\Delta_{0.5}$	$\Delta_{0.7}$	$\Delta_{1.0}$
llama	0.091	9.70	17.00	9.31	13.21	1.34	12.07
mixtral	0.068	0.19	-1.56	1.75	2.86	8.11	15.40
gemini	0.068	-2.27	-8.80	-11.90	-3.84	-1.50	-5.72
gpt-4o	**0.126**	-9.73	-10.44	-6.53	-15.10	-18.38	-16.37
gpt-4o-mini	0.105	1.29	-6.82	-8.43	-24.80	-16.95	-28.20

Table 2: ***Best***, *Worst*; *Most Resilient* , *Least Resilient*; Δ_r *is percent change in NDCG@k for* $D = r$ *and* $D = 0.0$.

MRR@5							
	MRR@$5_{0.0}$	$\Delta_{0.1}$	$\Delta_{0.2}$	$\Delta_{0.3}$	$\Delta_{0.5}$	$\Delta_{0.7}$	$\Delta_{1.0}$
llama	0.067	17.18	16.00	8.29	13.63	6.52	12.09
mixtral	0.041	3.85	5.20	4.43	7.13	9.63	21.77
gemini	0.045	-6.67	-14.56	-12.81	-14.56	3.51	-6.84
gpt-4o	**0.096**	-12.74	-15.04	-9.94	-14.63	-16.43	-17.09
gpt-4o-mini	0.073	5.76	-5.32	-5.10	-22.80	-11.29	-25.19
MRR@10							
	MRR@$10_{0.0}$	$\Delta_{0.1}$	$\Delta_{0.2}$	$\Delta_{0.3}$	$\Delta_{0.5}$	$\Delta_{0.7}$	$\Delta_{1.0}$
llama	0.073	13.19	15.93	9.41	13.49	3.58	11.80
mixtral	0.048	0.10	-0.19	3.48	5.23	5.66	16.80
gemini	0.050	-3.49	-12.37	-12.63	-8.08	3.32	-6.71
gpt-4o	**0.105**	-11.03	-13.48	-9.26	-15.81	-16.96	-17.93
gpt-4o-mini	0.082	3.97	-6.41	-7.32	-24.64	-13.23	-26.76

Table 3: ***Best***, *Worst*; *Most Resilient* , *Least Resilient*; Δ_r *is percent change in MRR@k for* $D = r$ *and* $D = 0.0$.

gpt-4o and gpt-4o-mini from OpenAI [28, 19].

4. Results & Discussion

We discuss the results of our experiments shown in Tables 1, 2, and 3. Surprisingly, we find that some LLMs show improved CRS performance in the presence of WSSE, while others are vulnerable to added WSSE and show deteriorated performance. In terms of initial performance, gpt-4o obtains the best performance, and mixtral obtains the worst performance. However, gpt-4o degrades in performance with increased Syn-WSSE, showing a 14.29% drop in Recall@5, a 9.73% drop in NDCG@5, and a 12.74% drop in MRR@5 at $D = 0.1$. On the other hand, mixtral generally improves in performance with increased Syn-WSSE, showing a 11.76% gain in Recall@5, a 6.81% gain in NDCG@5, and a 3.85% gain in MRR@5 at $D = 0.1$. Across the range of D, the most resilient model is llama, even improving in performance ($\Delta_r > 0$) in all cases but one, followed by mixtral. gpt-4o-mini is the least resilient, sustaining the largest drops in performance, followed by gemini. gpt-4o-mini never improves in performance, and gemini does in two cases.

4.1. Improved Performance $\Delta_r > 0$

Focusing our attention on the models that improved performance with an increasing number of disfluencies, llama is largely the most resilient model to WSSE disfluencies, improving in performance in the presence of Syn-WSSE as indicated by the positive Δ_r values. mixtral is also resilient, and also improves in performance in many cases.

Interestingly, both major LLM architectures – single transformer and mixture-of-experts[3] (MoE) – are represented in this category: llama is a large transformer model, and mixtral is an MoE model. We hypothesize that differences in pre-training and post-training (i.e. RLHF, DPO, etc.) methods and data impacts model resiliency, rather than the overall architecture. We hypothesize that these models may have been trained on **synthetic disfluent training data** [10, 18, 17] during pre-training or post-training. Hence, the type of data that LLMs are trained on has an impact on CRS performance.

Another perspective is that Syn-WSSE is a form of data augmentation itself, introducing genre diversity into the input text (by inserting the opposite genre into the text). The WSSE may be introducing **novel and diverse user interests** [22, 23, 24, 25] with the opposite genre augmentation. We hypothesize that some models (i.e. llama and mixtral) are able to take advantage of these novel and diverse genres for increased performance – perhaps due to their pre-training and post-training data and methods – while other models are not.

4.2. Deteriorated Performance $\Delta_r < 0$

Now turning to the models that deteriorated in performance with an increasing number of disfluncies, gpt-4o and gpt-4o-mini tie for the least resilient model to WSSE disfluencies, degrading in performance in the presence of Syn-WSSE as indicated by the negative Δ_r values. gemini is also not resilient, and also degrades in performance in many cases.

Naturalistic Disfluency Levels ($\Delta_{0.1} - \Delta_{0.3}$) In this range, disfluency levels are the most naturalistic as compared to real-world scenarios [11, 12]. gpt-4o is the least resilient model, and consistently deteriorates in performance. MRR@5 is the most impacted metric, dropping by 15.04% on $D = 0.2$ for gpt-4o. Recall@10 is the least impacted, still dropping by 7.14% on $D = 0.1$ for gpt-4o. This indicates that disfluency impacts the ranking of the recommended movies more than absolute inclusion/exclusion in L_c.

Balanced Disfluency Levels ($\Delta_{0.5}$) At $D = 0.5$, the half of all genres have induced Syn-WSSE. gpt-4o-mini is the least resilient, sustaining the largest percent drops in performance with $> 20\%$ drops in all metrics. In all cases except Recall@5, $\Delta_{0.4}$ and $\Delta_{0.6}$ are both greater than $\Delta_{0.5}$ for gpt-4o-mini.

Extreme Disfluency Levels ($\Delta_{0.6} - \Delta_{1.0}$) In the range of $\Delta_{0.6} - \Delta_{1.0}$, disfluency levels are extreme. The purpose of studying Syn-WSSE at these levels is to understand how these types of errors are represented by LLMs, in how they impact downstream task (CRS) performance in a stress test. gpt-4o-mini is the least resilient, sustaining the largest percent drops in performance with $> 11\%$ drops in all metrics. Interestingly, this minimum drop is less than the minimum $> 20\%$ drop in all metrics at the $D = 0.5$ disfluency level, indicating that the model recognizes an over-abundance of genres as a noisy signal.

4.3. Model Selection for CRS

In light of the differences in performance on the CRS task for different backbone LLMs, we recommend testing CRS systems with different backbone LLMs in the presence of Syn-WSSE during development. Our study shows that the choice of backbone LLM is a critical design decision for real-world situations, as this choice can impact system performance either positively ($\Delta_r > 0$) or negatively ($\Delta_r < 0$).

[3] MoE models consist of individual transformers linked together by a routing model, which forwards inputs to the "expert" transformer(s).

5. References

[1] M.-H. Shih, H.-L. Chung, Y.-C. Pai, M.-H. Hsu, G.-T. Lin, S.-W. Li, and H.-y. Lee, "Gsqa: An end-to-end model for generative spoken question answering," *arXiv preprint arXiv:2312.09781*, 2023.

[2] M. Rohmatillah, B. G. Ngo, W. Sulaiman, P.-C. Chen, and J.-T. Chien, "Reliable dialogue system for facilitating student-counselor communication," in *Proc. of Annual Conference of International Speech Communication Association*, 2024, pp. 1003–1004.

[3] OpenAI, "ChatGPT," 2023. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[4] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*, 2023.

[6] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," in *INTERSPEECH*, 2023.

[7] S. A. Hayati, D. Kang, Q. Zhu, W. Shi, and Z. Yu, "Inspired: Toward sociable recommendation dialog systems," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8142–8152.

[8] M. Kim, M. Kim, H. Kim, B.-w. Kwak, S. Chun, H. Kim, S. Kang, Y. Yu, J. Yeo, and D. Lee, "Pearl: A review-driven persona-knowledge grounded conversational recommendation dataset," *arXiv preprint arXiv:2403.04460*, 2024.

[9] S. Maji, M. Fereidouni, V. Chhetri, U. Farooq, and A. Siddique, "Mobileconvrec: A conversational dataset for mobile apps recommendations," *arXiv preprint arXiv:2405.17740*, 2024.

[10] B. Marie, "Disfluency generation for more robust dialogue systems," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 11 479–11 488.

[11] E. Shriberg, "Preliminaries to a theory of speech disfluencies," Ph.D. dissertation, 1994.

[12] ——, "Disfluencies in switchboard," in *International Conference on Spoken Language Processing*, vol. 96, 1996, pp. 11–14.

[13] T. A. Harley and S. B. G. MacAndrew, "Constraints upon word substitution speech errors," *Journal of Psycholinguistic Research*, vol. 30, no. 4, p. 395–418, 2001.

[14] J. F. Tree, "The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech," *Journal of Memory and Language*, vol. 34, no. 6, p. 709–738, 1995.

[15] S. Oviatt, "Predicting spoken disfluencies during human-computer interaction," *Computer Speech and Language*, 1995.

[16] A. Roelofs, "Self-monitoring in speaking: In defense of a comprehension-based account," *Journal of Cognition*, vol. 3, no. 1, p. 18, 2020.

[17] T. Passali, T. Mavropoulos, G. Tsoumakas, G. Meditskos, and S. Vrochidis, "Lard: Large-scale artificial disfluency generation," *arXiv preprint arXiv:2201.05041*, 2022.

[18] J. Yang, D. Yang, and Z. Ma, "Planning and generating natural and diverse disfluent texts as augmentation for disfluency detection," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1450–1460. [Online]. Available: https://aclanthology.org/2020.emnlp-main.113

[19] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[20] A. Dubey, A. Jauhri, A. Pandey, A. Kadian *et al.*, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[21] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[22] J. Wang, H. Lu, Y. Liu, H. Ma, Y. Wang, Y. Gu, S. Zhang, N. Han, S. Bi, L. Baugher *et al.*, "Llms for user interest exploration in large-scale recommendation systems," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 872–877.

[23] Y. Su, X. Wang, E. Y. Le, L. Liu, Y. Li, H. Lu, B. Lipshitz, S. Badam, L. Heldt, S. Bi *et al.*, "Long-term value of exploration: Measurements, findings and algorithms," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 636–644.

[24] J. Chen, C. Gao, S. Yuan, S. Liu, Q. Cai, and P. Jiang, "Dlcrec: A novel approach for managing diversity in llm-based recommender systems," *arXiv preprint arXiv:2408.12470*, 2024.

[25] K. C. Mahajan, A. Porobo Dharwadker, R. Shah, S. Qu, G. Bang, and B. Schumitsch, "Pie: Personalized interest exploration for large-scale recommender systems," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 508–512.

[26] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[27] E. Yuan, W. Guo, Z. He, H. Guo, C. Liu, and R. Tang, "Multi-behavior sequential transformer recommender," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1642–1652.

[28] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[29] B. Zheng, Y. Hou, H. Lu, Y. Chen, W. X. Zhao, M. Chen, and J.-R. Wen, "Adapting large language models by integrating collaborative semantics for recommendation," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1435–1448.

[30] S. Kim, H. Kang, S. Choi, D. Kim, M. Yang, and C. Park, "Large language models meet collaborative filtering: An efficient all-round llm-based recommender system," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 1395–1406.

[31] Z. Yang, J. Wu, Y. Luo, J. Zhang, Y. Yuan, A. Zhang, X. Wang, and X. He, "Large language model can interpret latent space of sequential recommender," *arXiv preprint arXiv:2310.20487*, 2023.

[32] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, and J. McAuley, "Large language models as zero-shot conversational recommenders," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 720–730.

[33] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao, "Large language models are zero-shot rankers for recommender systems," in *European Conference on Information Retrieval*. Springer, 2024, pp. 364–381.

[34] N. Kitaev and D. Klein, "Constituency parsing with a self-attentive encoder," in *Association for Computational Linguistics*, 2018.

[35] M. Mitchell, B. Santorini, M. Marcinkiewicz, and A. Taylor, "Treebank-3 ldc99t42 web download," p. 2, 1999.

[36] A. Taylor, M. Marcus, and B. Santorini, "The penn treebank: An overview," *Treebanks: Building and Using Parsed Corpora*, pp. 5–22, 2003.

[37] W. Hotopf, "Semantic similarity as a factor in whole-word slips of the tongue," *Errors in linguistic performance: Slips of the tongue, ear, pen, and hand*, pp. 97–109, 1980.

[38] M. Teleki, X. Dong, S. Kim, and J. Caverlee, "Comparing ASR systems in the context of speech disfluencies," in *Interspeech 2024*, 2024, pp. 4548–4552.