Interspeech 2024
1-5 September 2024, Kos, Greece

Evaluating Speech Recognition Performance Towards Large Language Model Based Voice Assistants

Zhe Liu, Suyoun Kim, Ozlem Kalinli

Meta, Menlo Park, CA, USA

zheliu@meta.com, suyounkim@meta.com, okalinli@meta.com

Abstract

In recent years, there has been a rise in the popularity of large language model (LLM) based voice assistants. A practical question being raised in the evaluation of cascaded automatic speech recognition (ASR) systems in LLM-powered voice assistants is how to determine whether any errors in ASR transcriptions will result in task failures for the downstream assistants. Thus, measuring ASR systems that can reflect voice assistants' perception and judgement becomes increasingly important. In this paper, we propose novel evaluation metrics by leveraging the same assistant LLM to project ASR hypotheses into a vector space and compute their semantic distances with respect to the references. We perform experiments on a curated OpenAssistant test set and demonstrate that our presented methods with semantic embeddings calculated from LLMs are superior to conventional metrics on evaluating ASR performance towards LLM based voice assistants.

Index Terms: speech recognition, large language model, voice assistant, semantic embedding

1. Introduction

Large Language Models (LLMs) have demonstrated significant potential as highly capable AI assistants, excelling in intricate reasoning tasks that demand expert knowledge spanning diverse fields [1, 2, 3, 4, 5, 6]. In recent years, LLM-powered voice assistant is gaining more and more popularity, where it listens for a spoken query, understands the semantics, and provides the requested information [7, 8]. It typically operates as a cascaded system that initially leverages an automatic speech recognition (ASR) system to transcribe the utterance query into a text prompt. This text is then passed to the downstream LLM to generate a suitable response.

Recent studies indicate that LLMs can demonstrate a degree of robustness to word-level perturbations in the input prompts, including misspelling and grammar mistakes [9, 10, 11]. Then a practical question being raised in the evaluation of ASR systems is how to determine whether any errors in ASR hypotheses will result in task failures for the downstream LLM-powered voice assistants. Thus, measuring ASR systems that can reflect LLM based voice assistants' perception and judgement has been becoming increasingly important.

Word error rate (WER) is a common metric for the performance of an ASR system. Derived from the Levenshtein distance [12], WER can be calculated as

$$\text{WER} = \frac{\sum_{i=1}^{n} o_i}{\sum_{i=1}^{n} m_i} \tag{1}$$

where m_i is the number of words in the ith reference, and o_i refers to the sum of insertion, deletion, and substitution errors computed from the dynamic string alignment of the recognized sequence with the reference sequence. However, WER has limitations in measuring semantic correctness since it treats every error (insertion, deletion, or substitution) equally. For instance, if the reference query is "What is holi?" and two ASR systems generate different hypotheses: "What is a holi?" and "What is holly?", the former would be favored by a downstream assistant, although these two ASR hypotheses have the same WER.

To overcome these drawbacks of WER, alternative metrics for evaluating ASR performance have thus been proposed to better measure the semantic correctness. In particular, semantic distance (SemDist) [13] uses sentence embeddings to calculate the semantic dissimilarities between references and hypotheses. Specifically, for the ith utterance, we first compute the semantic embeddings of the reference and hypothesis as e_i^{ref} and e_i^{hyp}, by performing mean-pooling over all output token embeddings from the pre-trained RoBERTa model [14]. We then calculate their cosine distance as follows

$$\text{SemDist}_i = 1 - \frac{(e_i^{\text{ref}})^T \cdot e_i^{\text{hyp}}}{||e_i^{\text{ref}}|| \cdot ||e_i^{\text{hyp}}||} \tag{2}$$

$$\text{SemDist} = \frac{1}{n} \sum_{i=1}^{n} \text{SemDist}_i \tag{3}$$

It was shown in [13, 15] that SemDist obtains higher correlation with the downstream natural language understanding (NLU) tasks as well as users' perception of ASR performance, compared to WER. Notice that SemDist can be obtained in various ways depending on which pre-trained language models are used, and most prior work focuses on *bi-directional* models including BERT [16, 17], RoBERTa [14, 13], or XLM [15, 18].

In any LLM-powered voice assistant, ASR hypotheses of voice queries are passed as input prompts to the LLM. One promising idea is to utilize the *same* LLM to project the ASR hypotheses into a vector space based on their semantics. In this way, the resulting sentence embeddings are more closely correlated with the perception and judgement of the LLM based assistant. Thus, SemDist with semantic embeddings computed from *auto-regressive* LLMs can be a better metric on evaluating ASR performance towards the LLM based voice assistant.

The application of LLMs to sentence embeddings remains an area of ongoing research [19, 20]. In this work, we aim to investigate the capabilities of LLaMA-Chat [4] for extracting sentence embeddings and evaluate the corresponding SemDist based metrics towards the voice assistants powered by LLaMA-Chat. In particular, we make the following contributions

- To the best of our knowledge, our work is the first to study LLM-powered voice assistants' perception and judgement of ASR performance, and we conduct extensive experiments on comparing various ASR evaluation metrics;

10.21437/Interspeech.2024-635

- We propose multiple approaches on extracting semantic embeddings from the same LLMs used for voice assistants, and then introduce the corresponding evaluation metrics based on cosine distances between references and hypotheses;
- Oftentimes, it is crucial to compare the performance between two ASR systems and determine whether any improvement is due to chance or real. Our paper also presents statistical methods for the significant tests on both WER and semantic dissimilarity metrics at the same time.

The rest of the paper is organized as follows. In Section 2, we describe the details of our proposed metrics on evaluating ASR performance towards LLM based voice assistants. Next, Section 3 illustrates the experiments and results for comparing these metrics with the conventional ones. Finally, we conclude in Section 4.

2. Methodology

2.1. Sentence Representations and Metrics with LLMs

In this section, we introduce multiple approaches on computing semantic embeddings of ASR transcriptions with LLMs. The resulting sentence representations are then used to compute the corresponding SemDist metrics for evaluating ASR quality towards LLM-powered voice assistants. Specifically, three LLM based metrics are explored as follows.

2.1.1. Prompt LLMs with raw text

Given LLMs are auto-regressive models, for each ASR hypothesis, we can directly utilize it to prompt the LLM and extract the hidden vectors of the final token as its sentence embedding. Formally, suppose the ith ASR hypothesis is denoted by $\mathbf{x}_i = (x_{i1}, \ldots, x_{in_i})$, then we pass it to the LLM and get the hidden states

$$\mathbf{h}_0^{(i)}, \mathbf{h}_1^{(i)}, \ldots, \mathbf{h}_L^{(i)} = LLM(\mathbf{x}_i) \tag{4}$$

where $\mathbf{h}_0^{(i)}$ represents the hidden state for the embedding layer, $\mathbf{h}_l^{(i)}$ with $l \in \{1, \ldots, L\}$ represents the hidden state for the lth Transformer [21] layer, and L refers to the total number of Transformer layers. Suppose the last hidden state can be written as $\mathbf{h}_L^{(i)} = (h_{L1}^{(i)}, \ldots, h_{Ln_i'}^{(i)})$, then we use the hidden state of the last token $h_{Ln_i'}^{(i)}$ as the sentence embedding of $\mathbf{x}_i$.

Similarly, we can calculate the sentence embedding of the ith reference, denoted by $r_{Lm_i'}^{(i)}$. The resulting representations of the ith hypothesis and reference can be used to compute the SemDist metric. We refer this method as LLMSemDist-Raw in the rest of the paper and specifically

$$\text{LLMSemDist-Raw}_i = 1 - \frac{(r_{Lm_i'}^{(i)})^T \cdot h_{Ln_i'}^{(i)}}{||r_{Lm_i'}^{(i)}|| \cdot ||h_{Ln_i'}^{(i)}||} \tag{5}$$

Alternatively, other aggregation or pooling methods from the hidden states can also be used. For example, we can take the average of the last hidden states over all the tokens, or we can take the average of final token's hidden states over multiplier layers. A comparison of these different aggregation methods is studied in Section 3.

2.1.2. Utilize the same prompts with assistants

System prompts are typically required to better instruct an LLM to act as a role of assistant [22]. Table 1 shows an example of template for prompting LLM based voice assistants. One natural approach is to leverage the same prompt and have it passed to the LLM. Then the resulting hidden state could be good semantic representation of the input query towards the LLM based voice assistants. Specifically, for the ith ASR hypothesis $\mathbf{x}_i$, we use $prompt(\mathbf{x}_i)$ to denote the resulting prompt with $\mathbf{x}_i$ plugged in. Then we obtain

$$\mathbf{h}_0^{(i)}, \mathbf{h}_1^{(i)}, \ldots, \mathbf{h}_L^{(i)} = LLM(prompt(\mathbf{x}_i)) \tag{6}$$

With a slight abuse of notation, let $h_{Ln_i''}^{(i)}$ represent the last hidden state of the last token, and we utilize it as the sentence embedding of $\mathbf{x}_i$. Similarly, we can compute the corresponding SemDist metric. We refer this method as LLMSemDist-Prompt in the rest of the paper and specifically

$$\text{LLMSemDist-Prompt}_i = 1 - \frac{(r_{Lm_i''}^{(i)})^T \cdot h_{Ln_i''}^{(i)}}{||r_{Lm_i''}^{(i)}|| \cdot ||h_{Ln_i''}^{(i)}||} \tag{7}$$

Table 1: *An example of text prompt template for triggering LLM based voice assistants, where "{raw text}" represents a placeholder for any ASR reference or hypothesis*

You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.
If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.
Human: {raw text}
Assistant:

2.1.3. Prompt LLMs with explicit one word limitation

The prompt template with explicit one word limitation (EOWL) is first introduced in [19], which is simple and straightforward by instructing LLMs in predicting the meaning of sentence in one word. With this template, for the ith ASR hypothesis $\mathbf{x}_i$, we consider

$$LLM(\{\textit{This sentence: "}\mathbf{x}_i\textit{" means in one word:}\}) \tag{8}$$

Unlike the usage of hidden states in [19], we utilize the logits of the LLM's output for the next token as the representation of $\mathbf{x}_i$. Here, the underlying motivation is that this prompt attempts to predict and summarize the meaning of sentence in one word, and the resulting logits could represent the entire sentence from semantic point of view. We refer this method as LLMSemDist-EOWL and let $g_{\text{hyp}}^{(i)}$ and $g_{\text{ref}}^{(i)}$ denote the resulting logits for the ith hypothesis and reference, respectively, then

$$\text{LLMSemDist-EOWL}_i = 1 - \frac{(g_{\text{ref}}^{(i)})^T \cdot g_{\text{hyp}}^{(i)}}{||g_{\text{ref}}^{(i)}|| \cdot ||g_{\text{hyp}}^{(i)}||} \tag{9}$$

Similar to (3), for each of the three LLMSemDist based metrics introduced above, we can take the average of utterance-level numbers to obtain the overall metric on an evaluation set. The metric is bounded between 0 and 2, where lower scores indicate higher semantic similarity and vice versa. In particular, if the hypothesis agrees with the reference, the metric is 0.

2.2. Coupled Bootstrap for Significance Analysis

When it comes to comparing the performance of two ASR systems on the same evaluation dataset, we would want to consider both the raw transcription quality and the task success rate of the downstream LLM based voice assistants. That is, we would need to compare their WERs, and at the same time, the LLM based metric LLMSemDist discussed in the previous section. Specifically, the absolute WER and LLMSemDist-EOWL differences between two ASR systems B and A are given by

$$\Delta W_{abs} = \text{WER}_B - \text{WER}_A = \frac{\sum_{i=1}^{n}(o_i^B - o_i^A)}{\sum_{i=1}^{n} m_i} \tag{10}$$

$$\begin{aligned}\Delta S_{abs} &= \text{LLMSemDist-EOWL}_B - \text{LLMSemDist-EOWL}_A \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{(g_{\text{ref}}^{(i)})^T}{||g_{\text{ref}}^{(i)}||} \cdot \left(\frac{g_{\text{hyp}_A}^{(i)}}{||g_{\text{hyp}_A}^{(i)}||} - \frac{g_{\text{hyp}_B}^{(i)}}{||g_{\text{hyp}_B}^{(i)}||} \right)\end{aligned} \tag{11}$$

Statistical significance analysis is typically needed to better understand whether any difference between ASR systems is due to chance or real. The work of [23] presents a *bootstrap* approach for significance analysis on ASR evaluation which makes no distributional approximations and is easy to use.

In this work, since we need to compare two metrics (WER and LLMSemDist-EOWL) simultaneously, consider the following algorithm that we name as *coupled bootstrap*.

For any $k = 1, \ldots, K$ where K is a large number, we randomly sample (with replacement) n utterances from the ASR evaluate set to generate a bootstrap sample of the evaluation results data, and then using this bootstrap sample, we compute the two metrics at the same time, denoted as $\Delta W_{abs}^{(k)}$ and $\Delta S_{abs}^{(k)}$.

Once we have all $\{(\Delta W_{abs}^{(k)}, \Delta S_{abs}^{(k)})\}_{k=1,\ldots,K}$, then the 95% confidence intervals for ΔW_{abs} and ΔS_{abs} can be determined by the empirical percentiles at 2.5% and 97.5% of the bootstrap sample statistics, respectively

$$C.I._{95\%}(\Delta W_{abs}) = \left[\Delta W_{2.5\%}^{Boot}, \Delta W_{97.5\%}^{Boot}\right] \tag{12}$$

$$C.I._{95\%}(\Delta S_{abs}) = \left[\Delta S_{2.5\%}^{Boot}, \Delta S_{97.5\%}^{Boot}\right] \tag{13}$$

Then we claim ASR system B is statistically significantly better than system A if *both* confidence intervals above do not contain the origin point of 0.

3. Experiments

In this section, we perform experiments for comparing various ASR evaluation metrics and measuring how they are correlated with the downstream LLM based assistant tasks.

3.1. Datasets and Models

We use the public OpenAssistant corpus in our experiments. We selected around 4k sentences with less than 100 characters and use them as text scripts for speech data collection. The speech data is collected using mobile devices through crowdsourcing from a data supplier for ASR. The data is properly anonymized and no User Identifiable Information (UII) is contained in the dataset.

Consider a cascaded voice assistant with an ASR model and 13B LLaMA-Chat model [4] as the downstream assistant. For each reference text in the curated OpenAssistant dataset, it is plugged into the template in Table 1 and then prompts the LLM. The ASR model is an RNN-T model with the Emformer encoder [24], LSTM predictor, and a joiner. It contains around 80M parameters and is trained from scratch using an in-house video dataset, which is sampled from public social media videos and de-identified before transcription; both transcribers and researchers do not have access to any UII.

We evaluate this ASR model on the curated OpenAssistant dataset and generate the hypotheses. The WER is calculated as 8.96. Then similarly, each ASR hypothesis is plugged into the template in Table 1 and prompts the LLM. For each utterance query, we compare the LLM's responses between the reference prompt and hypothesis prompt. Through human annotations, if the two responses have the equal quality and completeness, then the query is labeled as a successful assistant task; otherwise, it is labeled as a failure.

Note that in order to measure the predictive capabilities of various ASR evaluation metrics towards this LLM based assistant, we exclude the utterances that have no ASR errors. We also exclude the ones where the LLM doesn't provide useful information even with reference prompts. Among the remaining 2.5k utterances, approximately half of the queries for assistant task are labeled as success by human annotators.

3.2. Setups

We compare the following ASR metrics towards LLaMA-Chat

- WER: defined in (1);
- RareWER: measures WER only on words outside the top 90% cumulative word frequency distribution [25], computed on the video dataset. These words are often proper nouns and more important to the meaning of the utterance than common words;
- EER: entity error rate, measures WER only on the tagged named entities using the BERT-NER model [16, 26];
- SemDist: defined in (2)-(3), with semantic embeddings computed from the pre-trained RoBERTa model;
- LLMSemDist, including LLMSemDist-Raw, LLMSemDist-Prompt and LLMSemDist-EOWL, with their utterance-level metrics defined in (5), (7) and (9), respectively.

For each evaluation metric, we use the following criteria to evaluate its predictive capability on the task success rate of downstream LLaMA-Chat

- AUC, given the label of any assistant task is binary;
- Psuedo-R^2 from logistic regressions: each ASR evaluation metric is used as a single covariate to fit a logistic regression model with the assistant task labels as the response. Then we compute the Efron-R^2 [27] and McFadden-R^2 [28] scores. In particular, Efron-$R^2 = 1 - \sum_{i=1}^{n}(y_i - p_i)^2 / \sum_{i=1}^{n}(y_i - \bar{y})^2$, where y_i is the ith outcome label, p_i is the ith predicted outcome probability, and $\bar{y}$ is the expected value of the observed outcomes. McFadden-$R^2 = 1 - \ln \hat{\mathcal{L}}_{full} / \ln \hat{\mathcal{L}}_{null}$, where $\hat{\mathcal{L}}_{full}$ is the estimated likelihood of the full model and $\hat{\mathcal{L}}_{null}$ is the estimated likelihood of the null model (model with only intercept).

Each of these criteria above is between 0 and 1. The higher the criterion, the better the metric's performance at distinguishing between the success and failure labels of assistant tasks.

3.3. Comparison Results of ASR Metrics

To better evaluate the metric of EER, the curated OpenAssistant data is partitioned into two subsets, one containing the queries that have at least one tagged named entities in the references and the other without any tagged named entities.

Table 2 and Table 3 show the evaluation results of various ASR metrics' predictive abilities on the curated OpenAssistant data. For the results of AUC, we also report the 95% confidence interval. From the results, we can see that

- The metric of EER and RareWER outperforms WER, which is expected since the former two metrics focus more on the key words in the ASR references;
- The RoBERTa model based SemDist is more predictive than the three metrics above, which indicates that it can better represent the semantic information of the input query;
- The proposed three LLM based metrics demonstrate stronger capabilities than all other metrics. Specifically, using the sentence embeddings derived from the same assistant LLM has advantages over the pre-trained RoBERTa. Among these, LLMSemDist-EOWL slightly outperforms the other two.

Table 2: *Evaluation results of metrics' predictive abilities on curated OpenAssistant set **without** tagged entity names.*

	AUC (with C.I.)	*Efron-R^2*	*McFadden-R^2*
WER	0.679 [0.632, 0.725]	0.086	0.059
RareWER	0.690 [0.644, 0.733]	0.099	0.075
SemDist	0.745 [0.702, 0.786]	0.176	0.139
LLMSemDist-Raw	0.793 [0.753, 0.829]	0.245	0.195
LLMSemDist-Prompt	0.801 [0.763, 0.837]	**0.266**	**0.218**
LLMSemDist-EOWL	**0.803 [0.765, 0.839]**	0.256	0.212

Table 3: *Evaluation results of metrics' predictive abilities on curated OpenAssistant set **with** tagged entity names.*

	AUC (with C.I.)	*Efron-R^2*	*McFadden-R^2*
WER	0.702 [0.675, 0.728]	0.102	0.073
RareWER	0.713 [0.687, 0.739]	0.117	0.081
EER	0.723 [0.699, 0.748]	0.146	0.111
SemDist	0.754 [0.730, 0.778]	0.188	0.146
LLMSemDist-Raw	0.789 [0.765, 0.812]	0.206	0.158
LLMSemDist-Prompt	0.795 [0.772, 0.817]	0.213	0.166
LLMSemDist-EOWL	**0.802 [0.780, 0.824]**	**0.251**	**0.204**

To study the effort of various aggregation or pooling approaches on the LLM's hidden states, Table 4 show the comparison results for the variants of the LLMSemDist-Raw metric. We can see that taking the average of final token's hidden states over multiplier Transformer layers (first layer, middle layer (the 20th), and last layer) achieve the best result.

Table 4: *Evaluation results of the predictive abilities of different hidden states aggregation methods for* LLMSemDist-Raw *on curated OpenAssistant set **with** tagged entity names.*

	AUC (with C.I.)	*Efron-R^2*	*McFadden-R^2*
Last Token; Last Layer	0.789 [0.765, 0.812]	0.206	0.158
Last Token; Avg Layer	**0.795 [0.772, 0.817]**	**0.211**	**0.162**
Avg Token; Last Layer	0.761 [0.736, 0.784]	0.200	0.159

It is worth noting that different ASR metrics can be combined altogether. Table 5 shows the results on the linear combinations of multiple metrics. We can see that the aggregation of LLMSemDist-EOWL and EER gives the best performance, with the latter having the linear interpolation weight of 0.10.

Table 6 shows several examples of spoken queries and notice that each of these have the same utterance-level WER of 0.25. We can see that the metric of LLMSemDist-EOWL can better distinguish between the success and failure labels of assistant tasks. Note that LLM can to some extent tolerate the errors in ASR transcriptions, such as the first three queries.

Table 5: *Evaluation results of the predictive abilities of linear combinations between* LLMSemDist-EOWL *and* WER *or* EER *on curated OpenAssistant set **with** tagged entity names.*

	AUC (with C.I.)	*Efron-R^2*	*McFadden-R^2*
LLMSemDist-EOWL	0.802 [0.780, 0.824]	0.251	0.204
EOWL + WER	0.740 [0.715, 0.764]	0.154	0.110
EOWL + 0.10 × WER	0.792 [0.769, 0.814]	0.239	0.196
EOWL + 0.01 × WER	0.802 [0.780, 0.824]	0.251	0.205
EOWL + EER	0.800 [0.778, 0.822]	0.255	0.204
EOWL + 0.10 × EER	**0.816 [0.795, 0.837]**	**0.286**	**0.238**
EOWL + 0.01 × EER	0.805 [0.783, 0.826]	0.256	0.208

Table 6: *Examples on the references and ASR hypotheses of utterance queries and the corresponding* LLMSemDist-EOWL *scores, as well as the labels of downstream assistant tasks (i.e. whether the task is successful).*

Reference	*Hypothesis*	*Score*	*Label*
where's amazon rainforest located	where's the amazon rainforest located	0.0055	Yes
who who invented calculus	who is who invented calculus	0.0441	Yes
what's autism spectrum disorder	what's autism spectrural disorder	0.0634	Yes
who invented the jeans	who invented the genes	0.1192	No
what are the five boroughs of new york	what are the fried burgers of new york	0.1282	No

3.4. Significance Studies

In this section, we demonstrate an example of statistical significance analysis on two ASR evaluation metrics, WER and LLMSemDist-EOWL.

Consider two ASR systems. System A is the ASR model used in the experiments above, while system B is a larger ASR model with around 1B parameters.

Table 7 shows the 95% confidence intervals for the absolute WER and LLMSemDist-EOWL differences of ASR system B compared with ASR system A, computed from the coupled bootstrap method that is previously introduced. We can notice that for each confidence interval, the origin point of 0 is not included. Overall, we can see that ASR system B is statistically significantly stronger than system A.

Table 7: *Confidence intervals for the absolute WER and LLMSemDist-EOWL difference of ASR system B compared with ASR system A.*

	System A	System B	Δ_{abs} $(B - A)$
WER	8.96	6.68	[-2.61, -1.97]
LLMSemDist-EOWL	0.0191	0.0136	[-0.0063, -0.0045]
is overall stat. significant	-	-	Yes

4. Conclusion

In this paper, we introduce novel ASR evaluation metrics by utilizing the same assistant LLMs to project the hypotheses into a vector space and compute corresponding semantic distances. Experiments on a curated OpenAssistant test set demonstrate that our proposed methods with semantic embeddings computed from assistant LLMs are superior to conventional metrics on evaluating ASR quality towards LLM based voice assistants. In future, we plan to explore how the proposed metrics can be used to train ASR systems, as an additional objective.

5. References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[2] B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon *et al.*, "Bloom: A 176B-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, and A. Hartshorn, "LLaMA 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[5] OpenAI, "ChatGPT: Optimizing language models for dialogue," Feb 2022. [Online]. Available: https://openai.com/blog/chatgpt/

[6] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[7] A. Mahmood, J. Wang, B. Yao, D. Wang, and C.-M. Huang, "LLM-powered conversational voice assistants: Interaction patterns, opportunities, challenges, and design guidelines," *arXiv preprint arXiv:2309.13879*, 2023.

[8] X. L. Dong, S. Moon, Y. E. Xu, K. Malik, and Z. Yu, "Towards next-generation intelligent assistants leveraging LLM techniques," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 5792–5793.

[9] H. Wang, G. Ma, C. Yu, N. Gui, L. Zhang, Z. Huang, S. Ma, Y. Chang, S. Zhang, L. Shen *et al.*, "Are large language models really robust to word-level perturbations?" *arXiv preprint arXiv:2309.11166*, 2023.

[10] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, N. Z. Gong, Y. Zhang *et al.*, "PromptBench: Towards evaluating the robustness of large language models on adversarial prompts," *arXiv preprint arXiv:2306.04528*, 2023.

[11] R. Ma, M. Qian, P. Manakul, M. Gales, and K. Knill, "Can generative large language models perform ASR error correction?" *arXiv preprint arXiv:2307.04172*, 2023.

[12] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[13] S. Kim, A. Arora, D. Le, C.-F. Yeh, C. Fuegen, O. Kalinli, and M. L. Seltzer, "Semantic distance: A new metric for ASR performance analysis towards spoken language understanding," in *Proceedings of Interspeech*, 2021.

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[15] S. Kim, D. Le, W. Zheng, T. Singh, A. Arora, X. Zhai, C. Fuegen, O. Kalinli, and M. L. Seltzer, "Evaluating user perception of speech recognition system quality with semantic distance metric," in *Proceedings of Interspeech*, 2022.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] R. Whetten and C. Kennington, "Evaluating and improving automatic speech recognition using severity," in *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, 2023, pp. 79–91.

[18] A. Conneau and G. Lample, "Cross-lingual language model pretraining," *Advances in Neural Information Processing systems*, vol. 32, 2019.

[19] T. Jiang, S. Huang, Z. Luan, D. Wang, and F. Zhuang, "Scaling sentence embeddings with large language models," *arXiv preprint arXiv:2307.16645*, 2023.

[20] X. Li and J. Li, "DeeLM: Dependency-enhanced large language model for sentence embeddings," *arXiv preprint arXiv:2311.05296*, 2023.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[23] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 2004, pp. I–409.

[24] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.

[25] A. Xiao, W. Zheng, G. Keren, D. Le, F. Zhang, C. Fuegen, O. Kalinli, Y. Saraf, and A. Mohamed, "Scaling ASR improves zero and few shot learning," *arXiv preprint arXiv:2111.05948*, 2021.

[26] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003.

[27] B. Efron, "Regression and anova with zero-one data: Measures of residual variation," *Journal of the American Statistical Association*, vol. 73, no. 361, pp. 113–121, 1978.

[28] D. McFadden, "Conditional logit analysis of qualitative choice behavior," *IURD Working Paper Series*, 1972.