Comparison of Large Language Models for Speaker Recognition

Richard Han
rrhan@student.ubc.ca

Mu-Chen Liu
ml0729@student.ubc.ca

Charles Yan
cy001226@student.ubc.ca

Abstract

Dialogue attribution in the context of text analysis refers to the process of associating dialogue with the correct speaker in a conversation or narrative text. This is a crucial task in various fields such as natural language processing (NLP), literary analysis, and dialogue systems. Automating this task can be challenging, especially in texts where multiple characters interact closely, or where there are limited clues to identify the speaker. In recent years, large language models (LLMs) have gained significant attention for their ability to handle a wide range of natural language tasks with high proficiency. In this paper, we explore the capabilities of zero-shot LLMs in dialogue attribution through experimental analysis. The Mistral 7B Instruct model achieves the highest overall accuracy, while the LLamA 2 Chat model struggles to follow the instructions in the prompt to perform the task. We observe a positive correlation between dialogue attribution performance and the amount of provided context, particularly when the speaker is not explicitly stated in the dialogue. Additionally, the Mistral 7B Instruct model shows a performance plateau when identifying speakers whose identities are directly mentioned in the dialogue.

1 Introduction

Dialogue attribution aims to identify the speaker of a quotation in narrative literature, such as novels or short stories (Su et al. 2024). This task is vital for various applications, including audiobook production, where labeled scripts enable a speech synthesis program to automatically read lines in unique voices for each character. It also supports text mining efforts by facilitating the extraction of character social networks (Chen, Ling, and Liu 2021). Additionally, dialogue attribution aids in the automatic visualization of scenes (Elson and McKeown 2010), enhancing both academic research and multimedia entertainment production.

The speaker identification process involves inputting a quotation from the text and outputting the identification of the speaker's name. For example, the first quote in Jane Austen's *Pride and Prejudice* is:

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?",

which is spoken by Mrs. Bennet. The reader must infer from the mention of "his lady" that the speaker of this quote is Mr. Bennet's wife. This is an example of the anaphoric speaker (AS), where the speaker is identified by an anaphoric expression that refers back to the information previously mentioned in the novel. The two other categories of quotations are the explicit speaker (ES) and the implicit speaker (IS). The former refers to cases where the speaker is identified by name within the paragraph, while the latter involves the absence of speaker information within the paragraph (H. He, Barbosa, and Kondrak 2013). In cases involving anaphoric and implicit speakers, the difficulty of identifying the speaker increases and must be determined from the surrounding context and an understanding of the character relationships. The process of manually labeling quotations is labor-

Course project for UBC CPSC 440/550, April 2024.

intensive and would greatly benefit from automation. However, the idiosyncrasies of literary texts present challenges to automating the identification process (Vishnubhotla, Hammond, and Hirst 2022).

Previous work on the identification of speakers in novels used handcrafted features and support vector machines (H. He, Barbosa, and Kondrak 2013). Researchers in subsequent studies employed BERT-based models (Vishnubhotla, Hammond, and Hirst 2022). More recently, generative LLMs, ChatGPT in particular, have been applied to this problem where they demonstrate impressive zero-shot performance (Su et al. 2024).

Building on these successes, we conducted experiments with zero-shot prompting on LLMs other than ChatGPT. The clarity and directness of the prompt, which must provide the model with sufficient context to understand the task requirements without prior specific training, largely determines the effectiveness of zero-shot prompting. In this paper, we explore how different prompt designs affect the performance of LLaMA 2 7B, LLaMA 2 13B, and Mistral 7B. To assess model accuracy, we develop a stricter metric than the one used by Su et al. (2024), who deem a response correct if the true speaker's name (or an alias) appeared as a substring in the response (Su et al. 2024) (Chen, T. He, et al. 2023). We further explore the efficacy of zero-shot prompting under both strict and lenient metrics.

2 Background and Related Work

2.1 PNDC

A popular dataset for training and testing models for dialogue attribution is the *Project Dialogism Novel Corpus* (PDNC) which contains fully annotated dialogue on a collection of novels (Vishnubhotla, Hammond, and Hirst 2022). PDNC was originally released in 2022 and continues to get updates. It currently includes 29 novels with over 36,000 annotated quotations. Annotations for each quotation include the speaker, any addressees, as well as any referring expressions. Furthermore, aliases for all character names are present in the dataset, allowing for more lenient attributions.

2.2 Previous Approaches

Previous works in speaker identification have included supervised learning methods (H. He, Barbosa, and Kondrak 2013) and frameworks based on pre-trained models (Chen, Ling, and Liu 2021). H. He, Barbosa, and Kondrak (2013) employed a supervised learning approach, leveraging features associated with speaker names and neighboring utterances, achieving approximately 80% accuracy. Su et al. (2024) introduced the Speaker Identification via Generation (SIG) method, which encodes the quotation and its context based on a specially designed prompt template. This approach allows the model to either generate the speaker directly or evaluate the generation probability for any speaker candidates. They compared the performance of ChatGPT and SIG, finding that SIG outperforms the zero-shot ChatGPT, particularly excelling in implicit scenarios with up to a 17% improvement (Su et al. 2024). Chen, T. He, et al. (2023) introduced Symbolization, Prompt, and Classification (SPC), a framework that identifies implicit speakers in novels by symbolizing mentions, using prompts for classification, and aligning closely with pre-training tasks. SPC significantly enhances accuracy, outperforming existing methods and the newer ChatGPT with a 4.8% improvement and a 47% reduction in identification errors (Chen, T. He, et al. 2023).

However, the performance disparities in speaker identification among LLMs remain unclear. The prompt structures in previous work are not clearly written. Our study seeks to address three critical gaps in this domain: First, we need to investigate the effectiveness of zero-shot prompting further when assessed with strict and lenient metrics. Second, the impact of tailored system prompts specific to each model on LLM performance is not well-established. Third, the influence of expanding the context window size on the accuracy of LLM performance remains uncertain.

3 Methodology

3.1 Dataset

To evaluate the LLMs, we make inferences on the PDNC dataset. Due to resource constraints, we will only use annotated dialogue from *Pride and Prejudice* and *Emma* as our dataset. These are the same books that H. He, Barbosa, and Kondrak (2013) use, providing a solid benchmark for comparing our approach with LLMs. Table 1 shows the total number of quotes in each novel we use. Similarly, it shows the distribution of the quotation types (AS, IS, ES) introduced in 1.

Given some quote from the novel, we want the LLM to be able to predict who the speaker is. We can then use the PDNC dataset to get the true labels of the speaker. In addition to the characters true name, the PDNC also contains the aliases of each character. This allows some leeway in the inference as an alias of the characters name might be present in the context rather than their main name.

Since a single quote in PDNC could be comprised of multiple sub-quotations that are separate and the quotation text only contains text enclosed in quotation marks, important information between sub-quotations may be missing. Fortunately, PDNC also contains the locations of all sub-quotations in the novels. This allows us to get all of the text from the beginning of the first sub-quotations to the end of the last sub-quotation. The resulting string is then used for our inference.

To obtain the context around each quote, we chose to tokenize the surrounding by sentence using the Python NLTK library (Bird, Klein, and Loper 2009). We then take the N before and after the quote as our context window. This approach is similar to the approach of Chen, T. He, et al. (2023).

Table 1: Number of anaphoric, implicit, and explicit speaker quotes in novels.

Novel	Anaphoric Speaker	Implicit Speaker	Explicit Speaker	Total
Pride and Prejudice	306	637	327	1270
Emma	337	875	391	1593

3.2 Large Language Models

We measure the performance of the Mistral-7B-Instruct-v0.2 model (Jiang et al. 2023) on the *Pride and Prejudice* and *Emma* novels from PDNC. This model is fine-tuned for instructions that match the instruction-like prompts that we feed the model. In addition to the Mistral model, we also use the LLaMA 2 models (Touvron et al. 2023). We specifically choose the 7B and 13B parameter models to make inferences faster given our limited access to hardware. Here, we use the LLaMA models fine-tuned for chat instead of for instructions. There is an instruction-tuned model available for LLaMA2, however, this is trained on instructions related to code and does see some reduced performance for non-code related tasks (Rozière et al. 2024). These models have the benefit of being free and open-source.

3.3 Prompt Structures

Two prompt structures were used for inferences, a no-context prompt and a context prompt. The no-context prompt provides no neighboring context of the quotation and follows the structure

```
OUTPUT THE NAME OF THE CHARACTER WHO SAID:
'{quote}'
Only give me the speaker's name and nothing else.
Please do NOT include the quote in the response.
```

where `{quote}` is some dialogue quote from the novel. This prompt provides no additional information to the model beyond the quotation itself, so we expect it to perform relatively poorly. The context prompt gives the model some neighboring context of the quote as an input. This prompt follows the structure

```
CONTEXT: '\{left\_context\} + \{quote\} + \{right\_context\}'
```

```
GIVEN CONTEXT, OUTPUT THE NAME OF THE CHARACTER WHO SAID:
'{quote}'
Only give me the speaker's name and nothing else.
Please do NOT include the quote in the response.
```

Several context lengths were considered. We inferred on 1, 2, 4, and 8 length contexts. The context lengths are measured in sentences. That is, a context length of 2 would add two sentences before the quote and two sentences after the quote for the context. We will henceforth refer to the no-context prompt as a length 0 context prompt.

We also consider another permutation in the prompts. For the Mistral instruct models, we can wrap the instruction prompt with [INST] and [/INST]. The resulting prompt would thus be

```
[INST]
{prompt}
[/INST]
```

where {prompt} is one of the two prompts described above. This system prompt will enforce guardrails on the model, which we expect to improve its performance. We distinguish between the Mistral model using the wrapping by specifying that it is Mistral 7B Instruct with [INST].

4 Results

For each model, we vary the context lengths for inference and compare the model output with the true labels in PDNC to calculate accuracy. We calculate an overall accuracy as well as accuracies for AS, IS, and ES quotes. Furthermore, we consider two different metrics for determining whether an inference is correct. The weak metric looks at the entire output of the model and looks for the correct character name. A problem with this metric is that an output containing all names would be considered 100% accurate on all trials. Furthermore, it is not useful for predicting names when the true name is not known. The strong metric that we have created only considers the first name that appears. This metric is considered stronger since the model output may contain extra names despite our prompt structure specifying only a single speaker. It follows that the strong accuracy must be less than or equal to the weak accuracy. Accuracies using the weak metric are shown in Table 2 and the strong in Table 3.

Table 2: Weak Metric of LLMs on Anaphoric (AS), Implicit (IS), and Explicit (ES) Speakers Dependent on Context Lengths

		Context Length					
Model	Accuracy (%)	0	1	2	4	8	16
Mistral 7B Inst (Without [INST] Wrap)	Total	35.6	49.0	57.9	65.8	73.3	78.7
	AS	48.3	75.9	86.2	96.6	93.1	100.0
	IS	20.2	27.1	39.5	50.4	62.8	69.8
	ES	72.7	95.5	93.2	90.9	90.9	90.9
Mistral 7B Inst (With [INST] Wrap)	Total	31.8	49.2	58.5	68.4	76.4	82.3
	AS	26.3	51.6	69.2	80.7	87.7	90.7
	IS	15.2	24.6	34.8	48.7	61.4	71.6
	ES	71.3	98.5	98.7	98.5	97.6	96.9
LLaMA 7B Chat	Total	2.8	24.4	22.3	21.7	24.6	23.2
	AS	3.4	32.0	28.1	30.5	36.4	30.2
	IS	1.1	10.9	11.5	10.6	13.5	14.0
	ES	5.8	46.0	39.8	36.9	37.2	36.1
LLaMA 13B Chat	Total	4.5	6.0	3.9	3.1	3.8	4.7
	AS	3.7	5.0	4.7	4.4	3.7	5.8
	IS	3.3	4.5	2.9	1.6	3.3	3.8
	ES	7.9	10.0	5.3	5.3	5.0	5.7

Table 3: Strong Metric of LLMs on Anaphoric (AS), Implicit (IS), and Explicit (ES) Speakers Dependent on Context Lengths

		Context Length					
Model	Accuracy (%)	0	1	2	4	8	16
Mistral 7B Instruct (Without [INST] Wrap)	Total	31.2	46.5	56.4	64.4	71.3	77.2
	AS	44.8	69.0	86.2	96.6	89.7	100.0
	IS	14.0	24.8	37.2	48.1	60.5	67.4
	ES	72.7	95.5	93.2	90.9	90.9	90.9
Mistral 7B Instruct (With [INST] Wrap)	Total	30.4	48.3	57.1	67.0	75.4	81.5
	AS	24.4	51.0	68.9	80.2	87.2	90.2
	IS	13.7	23.2	32.2	46.3	59.7	70.4
	ES	70.8	98.3	98.7	98.5	97.5	96.8
LLaMA 7B Chat	Total	2.6	23.7	21.5	21.1	23.5	21.7
	AS	3.0	31.6	27.2	29.9	35.0	28.5
	IS	1.0	10.0	10.5	10.1	12.5	12.5
	ES	5.7	45.3	39.4	36.2	36.2	35.1
LLaMA 13B Chat	Total	3.0	5.6	3.8	3.0	3.6	4.6
	AS	2.5	4.7	4.5	4.4	3.6	5.3
	IS	1.5	4.1	2.8	1.5	3.1	3.8
	ES	6.5	9.5	5.2	5.0	4.7	5.6

5 Discussion

When making inferences on the Mistral 7B model without the `[INST]` wrapping, we found that outputs would sometimes contain code snippets or other extraneous text after the answer of the character name. Compared to the different prompts and LLMs we use, the situation is especially prevalent in the Mistral 7B model without `[INST]` wrapping. This results in many more tokens being consistently generated and significantly increasing inference time. Due to this, we limited the inference to just the first 100 quotes in each novel only for this model. This issue was also present in the LLaMA models although not at a high enough rate to significantly impact overall inference times.

Another issue we found was that in the LLaMA models, the output would be a simple *"Thanks!"* or *"Thank you!"*. This issue was more noticeable in the 13B model than in the 7B model. We suspect that this is due to the model being fine-tuned for chat. It is strange, however, that the 13B model would be more susceptible to this occurring. We can see this affects the inference accuracy in both the weak and strong metrics. For LLaMA 13B, total accuracy remains less than 10% regardless of the context window. LLaMA 7B improves especially given a larger context window, but the total accuracy plateaus quickly at around 20%.

Despite these issues, we see there is a correlation between the context length and the accuracy. This is across the 3 quotation types, AS, IS, and ES. A context length of 0 performs the worst as expected due to the limited information that the model is given. Increasing the context length increases the inference accuracy across all the quotation types. Of course, there does appear to be a maximum. The Mistral 7B with `[INST]` performs no better than 98.7% on ES quotes despite increasing the context length. In fact, in this particular example, a context length of 16 performs worse than all the other positive context lengths, although not by much.

Generally, the models struggled most with IS. Since no speaker information is included in the paragraph, more context is required. The accuracy of IS quotes is consistently lower than the other quotation types. Using a context window length of 16, we have yet to see a plateau of the accuracy for IS. With longer lengths, we could potentially see even more improvements. AS quotes follow as the next hardest quotation to infer the speaker of. Similar to IS, there appear to be more accuracy improvements available if we continue to increase context length. ES quotes are the easiest to infer as the speaker should be explicitly present in the paragraph. We quickly see a plateau in the ES accuracy at just a context length of 1. Further increases to context length do not continue to increase the ES accuracy.

As discussed in 2.2, other methods for identifying speakers exist. Chen, Ling, and Liu 2021 use a neural network based approach employing deep learning and achieve an identification accuracy of 82.5%. Su et al. 2024 employ speaker identification via generation (SIG) to achieve a maximal identification accuracy of 86.31% on PDNC. H. He, Barbosa, and Kondrak 2013 use a supervised machine learning model to achieve an accuracy of 86.5% in *Pride and Prejudice* and 80.1% in *Emma*. These results are comparable with the total accuracy of the Mistral 7B Instruct model.

6 Conclusion

We used Mistral 7B fine-tuned for instructions as well as LLaMA 7B and 13B fine-tuned for chat on the novels *Pride and Prejudice* and *Emma*. We used labelled quotations from PDNC to measure the accuracy of the inferences. LLaMA 13B and 7B struggled due to being fine-tuned for chat. Mistral 7B performed considerably better than LLaMA. In all models tested, we saw an increase in accuracy by increasing the length of context surrounding the quote. Using context lengths up to 16, we still do not see diminishing returns on the accuracy, implying that there may be more accuracy gains to be achieved.

Using LLMs is a promising approach to classifying and identifying speakers in quotations. It performs similarly to existing methods, however, performance could still be improved. We saw the potential effects that fine-tuning could make on the accuracy of inferences and the validity of responses, although this difference was seen on two different model architectures. This follows the general guidelines for using LLMs, where fine-tuning is recommended for maximal performance. Despite our lack of direct fine-tuning to this specific problem, Mistral 7B Instruct performed well, especially on larger context lengths. There are many promising directions for future research that could be explored in the future, such as larger or more varied context windows, retrieval augmented generation, and few-shot prompting.

References

Bird, S., E. Klein, and E. Loper (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media. ISBN: 9780596555719. URL: `https://books.google.ca/books?id=KGIbfiiP1i4C`.

Chen, Yue, Tianwei He, Hongbin Zhou, Jia-Chen Gu, Heng Lu, and Zhen-Hua Ling (2023). "Symbolization, Prompt, and Classification: A Framework for Implicit Speaker Identification in Novels." *The 2023 Conference on Empirical Methods in Natural Language Processing*. URL: `https://openreview.net/forum?id=olEEp3Phda`.

Chen, Yue, Zhen-Hua Ling, and Qing-Feng Liu (2021). "A Neural-Network-Based Approach to Identifying Speakers in Novels." *Proc. Interspeech 2021*, pp. 4114–4118. DOI: `10.21437/Interspeech.2021-609`.

Elson, David and Kathleen McKeown (Jan. 2010). "Automatic Attribution of Quoted Speech in Literary Narrative."

He, Hua, Denilson Barbosa, and Grzegorz Kondrak (Aug. 2013). "Identification of Speakers in Novels." *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Hinrich Schuetze, Pascale Fung, and Massimo Poesio. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1312–1320. URL: `https://aclanthology.org/P13-1129`.

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed (2023). *Mistral 7B*. arXiv: `2310.06825 [cs.CL]`.

Rozière, Baptiste, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve (2024). *Code Llama: Open Foundation Models for Code*. arXiv: `2308.12950 [cs.CL]`.

Su, Zhenlin, Liyan Xu, Jin Xu, Jiangnan Li, and Mingdu Huangfu (2024). *SIG: Speaker Identification in Literature via Prompt-Based Generation*. arXiv: `2312.14590 [cs.CL]`.

Touvron, Hugo et al. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv: 2307.09288 [cs.CL].

Vishnubhotla, Krishnapriya, Adam Hammond, and Graeme Hirst (2022). *The Project Dialogism Novel Corpus: A Dataset for Quotation Attribution in Literary Texts*. arXiv: 2204.05836 [cs.CL].

A Supplementary material

https://github.com/rrhan0/CPSC-440-540-speaker-identification