Interspeech 2024
1-5 September 2024, Kos, Greece

Leveraging Large Language Models for Post-Transcription Correction in Contact Centers

Bramhendra Koilakuntla[1], *Prajesh Rana*[1], *Paras Ahuja*[1], *Srikanth Konjeti*[1], *Jithendra Vepa*[1]

[1]Observe.AI, India

{koilakuntla.bramhendra,prajesh.rana,paras.ahuja,srikanth.konjeti,jithendra}@observe.ai

Abstract

Contact centers depend on Automatic Speech Recognition (ASR) to power their downstream tasks. However, any mis-transcription in the ASR can have a significant impact on their downstream tasks. This issue is compounded by the extensive array of diverse brand and business names. Traditional transcription correction methods have a long development cycle and require skilled resources. Most of the time these errors will have a context, suggesting a search and replace solution in the post-call analytics platform. But identifying these contexts is time-consuming and tedious. Moreover, these words may get recognized in various similar forms, further complicating the situation. To tackle this, we propose a post transcription correction module by employing Large Language Models (LLMs) to detect these contexts, termed 'anchors'and to correct phonetically similar misrecognised words. By leveraging anchor phrases, we can pinpoint the error occurrences and correct the misrecognized.

Index Terms: Post-call Transcription, Automatic Speech Recognition, Large Language Model, Transcription Correction

1. Introduction

Contact centers play a pivotal role in various industries, serving as the front line for customer interactions. To streamline their operations, call centers heavily rely on ASR technology, which converts spoken language into text. This text used by post-call platforms to facilities lot of downstream tasks which assess the agents and customer behaviours. However, a persistent challenge in the call center industry lies in accurately recognizing brand names and vertical-specific terms, such as medical jargon and university names. Correcting such misrecognized words important as they are coupled with downstream tasks. Various approaches in the literature such as out of vocabuluary (OOV) modules and language module rescoring are aimed to do these tasks but they often have long development cycle and require computational and technical resources. Especially if we are using external ASR vendor, the implementation becomes complicated.

Contact centers typically adhere to predefined scripts, which often provide context for error words. By identifying this context, we can employ a straightforward search and replace mechanism to correct the words in post-call analytis platform. However, manual identification of these contexts is challenging due to the long tail. Recent advancements in LLMs have made various tasks more accessible with a simple prompt. Here, we utilize LLMs to extract context phrases or anchors. The LLMs searches for suitable instances where the word might appear, providing both the left context and the potentially misrecognized word. We then replace the phrase with the correct word. There by improving the post call transcription quality for better analytics. Additionally, one common ASR error involves substituting the word with a phonetically similar one. The presented approach is model agnostic, can be developed even with external vendors. Here, our aim is to address this error by tasking LLMs with generating a list of all phonetically similar words. With human intervention, we can identify and finalize potential errors, replacing those words with the correct ones.

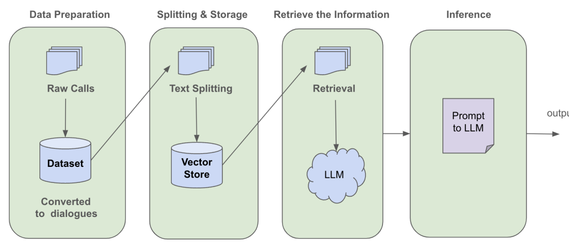

Figure 1: *Diagram for the LLM based Post Transcription Correction*

2. Approach

Correct Words: This refers to the accurate transcription of the brand name, product name, or industry-specific terms.

Target Words: These encompass all incorrectly transcribed words corresponding of Correct words

Anchors: These are collections of phrases derived from the transcription that contain the target words and correct words.

We leverage past conversations between agents and customers related to the target brand, inputting them into the LLM for retrieval purposes. Using a retrieval chain, we query this data to extract relevant contexts related to the target words. We then identify the best candidates for phonetically similar words that may be susceptible to mistranscription from the correct word. This approach involves multiple steps, including data preparation, text splitting, and vectorization of the data. The processed data is then stored into a vector store and presented to the retrieval chain using an efficient system prompt. Once the retriever chain is established, the subsequent step entails generating anchors based on the retrieved data using appropriate prompts. The approach is illustraed in Figure 1.

2.1. Call Sampling and Text Formatting

To simplify inputting calls into the LLM, we selected approximately 100 calls between agents and customers. These calls,

```
"anchors": [
{
  "actual_text": "calling pay more",
  "corrected_output": "calling paymor "
}

{
  "actual_text": "choosing pay more",
  "corrected_output": "choosing paymor"
}

{
  "actual_text": "calling pay shore customer
support",
  "corrected_output": "calling paymor
customer  support"
}

{
   "actual_text": "contacting ray more dot
com",
   "corrected_output": "contacting paymor dot
com"
}
]
```

Figure 2: *Example illustration of anchors generation*

from interactions with the same customer, feature different agents and conversation durations. We formatted them to depict dialogue, consolidating all conversations into a single text file. Each conversation was appended sequentially, marked with 'start of conversation' and 'end of conversation' tokens for clarity and structure.

2.2. Text Splitting and Storage

To tailor the data for the LLM, we first break the large document into smaller segments using a recursive splitting technique. This step is crucial for optimizing the model's comprehension. Organizing related text fragments in close proximity within these chunks enhances the model's coherence and relevance in generating responses. Post chunking, we employ an open-source embedder to create embeddings for each segment, storing them in the vector store.

2.3. Retrieval-Augmented Generation

In our methodology, we utilize Retrieval-Augmented Generation (RAG) to retrieve the anchors. First, we convert the sampled calls into embeddings and then provide them to the LLM to generate the RAG chain. This chain functions as an inference mechanism, enabling us to efficiently query the data and obtain the desired anchors. To facilitate this process, we have employed langchain[1] as our development tool and OpenAI GPT-3.5 for LLM [2].

3. Generating The Anchors

3.1. Anchors Identification

In this phase of anchors generation, our objective is to produce all possible contexts surrounding the correct word. For instance, if the correct word is a brand name like 'Paymor', our goal is to generate various contexts in which 'Paymor' might appear. Additionally, we aim to identify contexts where 'Paymor' could potentially be mis-transcribed as another word. For example, such contexts might include phrases like 'for calling pay more', 'for connecting pay', 'for contacting pay dot ', and 'for choosing pay'. These diverse contexts, including those with potential

Original Conversation	Corrected Conversation
agent : Thank you for calling pay shore customer support. My name is Sarah, how can i assist you today?	agent : Thank you for calling paymor customer support. My name is Sarah, how can i assist you today?
customer : Hi Sarah, I'm having trouble accessing my account online. It keeps saying my password is incorrect.	customer : Hi Sarah, I'm having trouble accessing my account online. It keeps saying my password is incorrect.
agent : I'm sorry to hear that. Let me assist you with that. Can you please provide me with your ray more account username?	agent : I'm sorry to hear that. Let me assist you with that. Can you please provide me with your paymor account username?

Figure 3: *Example illustration of word correction*

transcription errors, serve as valuable resources for target word replacement and facilitate accurate anchors generation. An example illustration of anchors generation is shown in Figure 2. Once the anchors are generated we use a search and replace mechanism to replace the mistranscribed words to correct ones. In Figure 3 an example conversation is shown.

3.2. Phonetically Similar Word Search

In the phonetically similar word search, our objective is to identify all words that sound similar to the correct word, which could potentially be mistranscribed variations of the correct word. For example, if the brand name is 'Paymor', we seek to gather all possible phonetically similar words, such as 'pay more '. These phonetically similar words serve as potential candidates for correcting the target word. We have conducted experiments to demonstrate effectiveness of this approaches. Table 1 shows the improvements in the recognised words after applying anchors corrections and similar word corrections. We've conducted an evaluation to assess the time and the number of anchors that can be manually retrieved. Our findings indicate that certain anchors were overlooked, and there is potential for significant time reduction in the retrieval process.

Method	Actual Count	Post Correction	Time in hr
Manual	1160	3050	15
Proposed	1160	3201	0.08

Table 1: *Results for the word 'Paymor' of 500 calls. Before applying the correction and post applying LLM correction*

4. Conclusion

In summary, our proposed method offers a simple yet robust solution to tackle transcription errors in contact centers. With a post-transcription correction module leveraging Large Language Models to identify contextual cues, we can swiftly locate and rectify errors, thus improving transcription accuracy. This approach holds significant potential for post call analytics in contact centers.

5. References

[1] "langchain," https://www.langchain.com/retrieval, accessed: Apr 24, 2024.

[2] "OpenAI," https://platform.openai.com/docs/api-reference/introduction, accessed: Apr 24, 2024.