Leveraging Third-Party LLMs' Annotations for Sensitive Conversational Data Abstractive Summarization

Anonymous ACL submission

Abstract

Previous studies have demonstrated the effectiveness of Large Language Model (LLMs) in various text annotation tasks. However, the use of LLMs as annotators still presents significant limitations that impede their practical efficiency, especially when used through an external API. Particularly, when dealing with sensitive or confidential information in the data to be annotated, relying on a third-party API for LLMs may not be suitable due to privacy concerns. For instance, annotating customer service call transcripts using an LLM for summaries may risk exposing sensitive information discussed during the conversation. In this study, we address this specific challenge by proposing a pipeline that leverages LLM annotations while maintaining the confidentiality of sensitive information submitted through the API.

1 Introduction and Related Work

Recent studies has shown that LLMs have emergent abilities (Wei et al., 2022), i.e., unpredictable abilities that are not present in smaller pretrained models. Among these emergent abilities are the *in-context learning* and *instruct following* (Zhao et al., 2023). In-context learning was initially introduced with the release of GPT-3 (Brown et al., 2020) where the authors demonstrated that their autoregressive LLMs could perform specific tasks when provided with an instruction and zero/few demonstration examples of the task to be performed. Subsequently, the LLM is capable of performing predictions on unseen examples by completing the text without the need for any further gradient updates. Instruct following, on the other hand, consists of fine-tuning LLMs on tasks phrased as instructions. This fine-tuning step can improve the LLMs' performance and generalization on unseen tasks, as demonstrated by Chung et al. (2022). Moreover, it can also help to better align the LLMs' outputs with human intents, as shown by Ouyang et al. (2022). Several recent studies capitalized on these two emergent abilities to perform data augmentation and annotation and potentially fine-tune smaller models in a supervised fashion (Sahu et al., 2022; Yoo et al., 2021; Shridhar et al., 2022). To showcase the effectiveness of LLMs in performing annotation tasks, Gilardi et al. (2023) conducted a study where ChatGPT outperformed Mechanical Turk annotators on 4 out of 5 classification tasks. Furthermore, Soni and Wade (2023) demonstrated that this capability can be extended to generative tasks, highlighting that human annotators were unable to differentiate between generated and human-written summaries. However, a significant limitation is that ChatGPT's weights are not accessible to researchers and NLP practitioners, and querying the model should be done through an OpenAI API. Even with the recent release of open LLMs, such as Llama (Touvron et al., 2023) and Falcon (Penedo et al., 2023), the majority of NLP practitioners are still unable to utilize these models privately due to their limited resources. This constraint is particularly restrictive in scenarios where the data to be annotated or augmented is confidential or sensitive. Our main focus in this study is the dialogue summarization task for customer service calls. These calls often involve a lengthy exchange between a customer and an agent regarding an issue. Therefore, an automated summarization system can greatly enhance service efficiency by generating a compact summary that effectively conveys the relevant and salient information within the dialogue (Zou et al., 2021). However, training a dialogue summarization system is a challenging task, primarily due to two reasons. Firstly, the availability of publicly annotated data is limited. Secondly, concerns related to the confidential nature of customer service calls create privacy and security concerns about the direct usage of third-party LLMs for annotation. In our work, we address these challenges by proposing a pipeline that enables the use of external

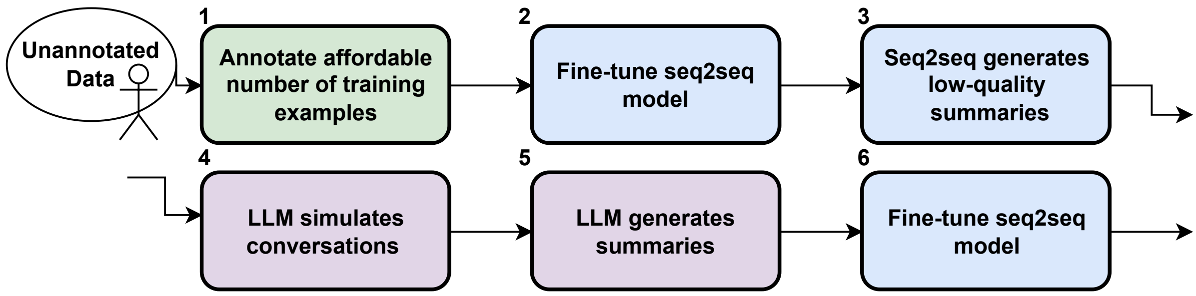


Figure 1: The pipeline involving three actors: Human annotators (green) responsible for the initial manual annotation. The seq2seq model (blue) that is fine-tuned using annotated and simulated data. The LLM (purple) accessed through an external API, which is utilized for generating simulated conversation-summary pairs.

LLMs, for annotating customer service calls with summaries without compromising the confidentiality of sensitive information within the calls. Our pipeline for annotating customer service calls with summaries from unannotated transcripts involves six steps as detailed in section 2. We applied our pipeline to an internal dataset of customer service call transcripts and conducted additional experiments on SAMSum (Gliwa et al., 2019) and three other public datasets that cover a wide range of domains.

2 Pipeline

Figure 1 depicts our proposed pipeline, which involves three actors: human annotators, a seq2seq model, and an Online LLM accessed via API. Initially, the pipeline begins with a corpus consisting of unannotated dialogues. In the first phase, human annotators are assigned the task of annotating a minimum number of training examples with summaries that do not contain any confidential information. These examples should be sufficient to fine-tune the seq2seq model, enabling it to generate summaries that capture the main topic of the dialogue, regardless of their quality and factuality. The fine-tuned model is then utilized to annotate the remaining dialogues with summaries. Although these generated summaries are expected to be of low quality, it is not a limitation in our case, as their purpose is to provide a diverse range of topics that can be used to simulate conversations using the LLM. Thus, the next phase is to query an instruction-fine-tuned LLM to simulate conversations based on the summaries generated by the seq2seq model, and subsequently, the same LLM is employed to generate summaries for the simulated conversations. The instructions used to query the LLM should follow the instructions given to the human annotators in the first phase. We provide the instructions used in our experiments in Appendix B. Finally, we fine-tune our seq2seq model using the simulated corpus, and then further refine it by conducting further fine-tuning on the annotated data from the first phase. Unless otherwise mentioned, in this work, we consider 150 training examples in the first phase. We use the BART (Lewis et al., 2020) and BARThez (Kamal Eddine et al., 2021) models for English and French data, respectively, as our seq2seq model. Additionally, we utilize the `gpt-3.5-turbo` model as the instruction-fine-tuned LLM accessed through an API.

3 Experiments on SAMSum

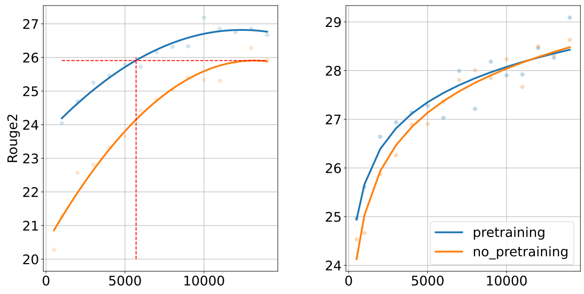
As mentioned earlier we opted for experimenting on SAMSum - a publicly available dialogue summarization dataset. SAMSum has 14732 train, 819 test and 818 validation examples. This choice was motivated by the following factors: First the lack of public customer service calls transcripts. Secondly, the possibility for more extensive analysis enabled by the existence of annotated validation and test sets. Consequently, utilizing SAMSum enables replication of the results presented in this study, free from any constraints pertaining to confidentiality or sensitive data. To simulate a real-world scenario where the data is unannotated, we randomly sampled 150 training examples from the train set and we consider these examples as the one annotated by humans in the first phase of the pipeline. We conduct additional experiments on three other datasets in Appendix A.

3.1 Experimental Setup

We experimented with both BART-Base and BART-Large as our pipeline seq2seq model. For all the reported results we fine-tuned the model for five

Training Data	Rouge1	Rouge2	RougeL
FTS	49.7/52.8	25.5/27.9	41.5/44.0
HAE	42.1/45.8	18.4/20.6	34.8/36.4
SE	41.3/43.9	15.1/16.4	32.8/34.0
SE + HAE	46.2/48.5	21.5/22.4	37.7/39.0

Table 1: Performance comparison of BART-Base (left) and BART-Large (right) models fine-tuned on different training sets. FTS includes the full training set with 14732 human-annotated examples. HAE represents the 150 human-annotated examples from Phase 1 in our pipeline. SE denotes the 14732 simulated examples generated in Phases 4 and 5.



(a) BART-Base

(b) BART-Lage

Figure 2: The evolution of the seq2seq model performance in function of the number of training examples.

epochs and used a learning rate that warmed up during 6% of the training steps and then decreased linearly to 0 at the end of the training. We fixed the batch size to 8 and chose the maximum learning rate from $\{10^{-5}, 5.10^{-5}, 10^{-4}\}$ based on the best validation score. All experiments were conducted on a single Nvidia V100 (32GB) GPU.

3.2 Results

Table 1 shows the results of the fine-tuning on data produced by different phases in the pipeline. The first row corresponds to the fine-tuning on the full training set, which can be considered as the theoretical upper bound performance that can be achieved by the seq2seq model when the pipeline is applied. We can observe that when the model is fine-tuned solely on the data simulated by the LLM, it lags significantly behind the performance achieved through full training. Similarly, there is a notable gap of approximately 7 Rouge1 points between the full training and fine-tuning solely on the human-annotated examples. However, this gap is almost halved when the fine-tuning on the human-annotated examples is preceded by fine-tuning on the LLM's simulated data. This finding highlights the substantial positive contribution of the pipeline to the final performance.

How many training points do we need to achieve the full training performance?

Based on the results presented in Table 1, we expect that training on the simulated data can serve as a pretraining step to boost the performance of the seq2seq model when fine-tuned with the end task data. To further investigate this assumption, we analyze the learning curve of the seq2seq model by gradually incorporating more training points during the fine-tuning process. We compare the model's performance in two scenarios: first, by directly applying fine-tuning to the annotated data, and second, by performing fine-tuning after pretraining the model on simulated data. Figures 2a and 2b illustrate the learning curve of BART-Base and BART-Large, respectively, with and without the pretraining step. First, in the case of BART-Base, we observe a significant improvement in the model's performance when it was pretrained on the simulated data. This improvement in performance persisted as we added more examples during fine-tuning, allowing the model to achieve its full performance using approximately 5700 fine-tuning examples instead of the original 14732. On the other hand, BART-Large demonstrates superior performance when pretrained on simulated data until around 10000 fine-tuning points. Beyond that, both the pretrained and directly fine-tuned models exhibit similar performance. We leave the investigation of this behavior for a future work.

4 Experiments on Internal Data

The internal data that we used in our experiments are the transcripts of customer service calls. They consist of a dialogue between a single agent and a single client where the agent tries to assist the customer to resolve an issue or address a concern. The dialogues were transcribed using an internal French Automatic Speech Recognition (ASR) tool with a word error rate of around 10%. For our experiments, we used a total of 10000 unannotated transcriptions. To apply our pipeline to the internal data, an internal team of linguists annotated an additional 410 examples. From this set of 410 annotated examples, we randomly selected 150 examples for the initial phase of our pipeline, while

Training Data	Rouge1	Rouge2	RougeL
HAE	47.2/50.2	22.3/22.5	30.1/29.0
SE	52.0/53.1	25.2/26.2	33.3/35.5
SE + HAE	53.5/54.2	26.3/27.5	35.0/36.3

Table 2: Performance comparison of BARThez (left) and mBARThez (right) models fine-tuned on different training sets

Training Data	Coh.	Cons.	Flu.	Rel.
Gold	83.1	85.8	75.0	86.3
HAE	9.4	11.3	11.0	12.9
SE	20.7	25.8	21.5	25.8
SE + HAE	31.5	29.0	27.4	34.4

Table 3: Human evaluation using best-worst scaling.

the remaining 260 examples served as a test set. Due to the limited number of annotated data, we did not employ a validation set. Instead, we utilized a fixed learning rate of 5.10^{-5} and conducted three epochs of fine-tuning. These hyper-parameter choices were based on the optimal configuration obtained from the SAMSum experiments, as detailed in Section 3.1.

API compute budget: As mentioned earlier, we utilized the `gpt-3.5-turbo`, incurring a cost of 0.0015 USD per 1K tokens during the experimental phase. On average, each example involved a total of 1160 tokens. For the generation of 10K synthetic examples, the total cost amounted to 17.4 USD.

4.1 Results

The performance of BARThez and mBARThez when fine-tuned on different data sources is presented in Table 2. The application of our pipeline resulted in a substantial performance boost for the seq2seq models, with a gain of 6.3 and 4 absolute points in Rouge1 for BARThez and mBARThez, respectively. One notable finding concerning the internal data is that fine-tuning solely on the simulated data yielded superior results compared to fine-tuning on the initial set of 150 human-annotated examples. This outcome contrasts with the findings from SAMSum. A possible explanation for this discrepancy is that the prompt utilized for generating the simulated conversation-summary pairs was more adequate in the case internal data, enabling the generation of examples that closely align with the distribution of human-annotated instances.

4.2 Human Evaluation

To validate our findings from the automatic evaluation, we conducted a human evaluation on our internal data. In this evaluation we consider the dimensions proposed by Fabbri et al. (2021). These dimensions are *coherence* (collective quality of all sentences), *consistency* (the factual alignment between the summary and the summarized source), *fluency* (quality of individual sentences) and *relevance* (selection of important content from the source). For simplicity, we adopt the *the best-worst scaling* approach (Narayan et al., 2018; Kamal Eddine et al., 2021, 2022), where we compare all summaries pairs and we report for each model the percentage of time its summary was chosen as *best*. In the human evaluation we include the models form table 2 in addition to the *gold* summaries. For this annotation task, we randomly selected 50 conversations from the test set and enlisted the participation of 19 internal volunteers. Each conversation was annotated by three different participants, resulting in an average of approximately 8 conversations per volunteer. A summary is considered as *best* only if it is judged by at least two annotators to be so.

Results. Table 3 shows the best-worst scaling score for each of the four dimensions. For all the dimensions, we obtain the same ranking order as in the automatic evaluation with wider and more interpretable margins. The performance of the seq2seq model maintains a noticeable improvement margin across all the considered aspects when our pipeline is applied. As a result, the human evaluation validates the automatic one.

5 Conclusion

In this work we have presented a novel pipeline that harnesses the power of LLMs accessed through APIs to provide effective summarization of customer service calls while maintaining the confidentiality of sensitive data. Our pipeline has been successfully applied to four public datasets and an internal customer service private dataset. In both cases, the automatic evaluation indicates that the proposed pipeline significantly enhances the performance of the summarization model, particularly in scenarios where annotated data is scarce. We finally conducted a human evaluation on the internal data that validated the results of the automatic evaluation.

Limitations

In this section, we outline certain limitations that merit additional exploration:

1. **Sensitive Data Leakage**: While the risk of sensitive data leakage in the summaries submitted through the API is extremely low, we have conducted a thorough examination to address this concern through supplementary analyses. Initially, we manually assessed 200 random summaries and found no disclosure of confidential information in any of them. Additionally, within our specific context, we evaluated the potential risk of an isolated entity's leakage and determined that such leakage would not reveal the participant's identity. However, it's important to note that when applying our methodology in other contexts, additional risk analysis may be necessary. We will explore additional directions that can limit the leakage risk in our future research:

    - Reducing confidential information using classification tools for anonymization in the third phase of the pipeline.
    - Introducing penalties for the generation of confidential information during the summarization process. These penalties can be enforced through supervised or reinforcement learning techniques.

2. **Reproducibility**: Because of the confidential nature of the task, we were unable to publish the internal dataset used in the initial experiments. To address this issue, we conducted extensive experiments on four publicly available abstractive summarization datasets, covering a wide range of domains. We provide both the code for reproducing the results on these public datasets and the data generated in each phase of the pipeline.

References

Asma Ben Abacha, Wen-wai Yim, Yadan Fan, and Thomas Lin. 2023. An empirical study of clinical note generation from doctor-patient encounters. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2291–2302, Dubrovnik, Croatia. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexander R Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *EMNLP-IJCNLP 2019*, page 70.

Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. 2021. Barthez: a skilled pretrained french sequence-to-sequence model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9369–9390.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. AraBART: a pretrained Arabic sequence-to-sequence model for abstractive summarization. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 31–42, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions. *arXiv preprint arXiv:2212.00193*.

Mayank Soni and Vincent Wade. 2023. Comparing abstractive summaries generated by chatgpt to real summaries through blinded reviewers and text classification algorithms. *arXiv preprint arXiv:2303.17650*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yicheng Zou, Lujun Zhao, Yangyang Kang, Jun Lin, Minlong Peng, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021. Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14665–14673.

A Additional Experiments

To further validate our approach, we apply our pipeline to three additional abstractive summarization datasets. These datasets are the following:

• **mts-dialog** (Ben Abacha et al., 2023): A dataset containing 1.7k brief doctor-patient conversations alongside their corresponding summaries.:

• **dialogSUM** (Chen et al., 2021): A dataset containing face-to-face spoken dialogues covering a range of daily-life topics. It encompasses 13,460 dialogues, each accompanied by manually labeled summaries.:

• **CNN/DM** (See et al., 2017): An English-language dataset containing news articles from CNN and the Daily Mail, accompanied by highlights concatenated abstractive summaries.

We replicate the experimental setup described in Section 3.1. However, for practicality, we restrict the number of CNN/DM articles used for generating low-quality summaries to 10K.
The results on these three datasets, as shown in Table 4, are consistent with our initial findings, demonstrating that our approach can be generalized to other domains, such as news articles and doctor-patient conversations summarization.

B Instructions

First, to generate simulated conversations for the SAMSum dataset, we utilized the following instruction to query the LLM. This instruction closely aligns with the guidelines provided to the human annotators in the original study (Gliwa et al., 2019). To determine the length of the simulated conversation, we employ a simple linear regression model that predicts the number of utterances based on the number of words within the summary.

```
formality = ["formal", "informal", "semi-
formal"]

Text = f"""
Based on the following summary write a
natural messenger-like conversation simi-
lar to those written on a daily basis:

summary:  {summary}

- Use arround {length} utterances.
- the dialogue should be written in {ran-
dom.choice(formality)} language.
"""
```

Similarly, To generate simulated summaries we query LLM with an instruction that follows the guidelines provided to the human-annotators. To choose the length of the generated summaries, we use a simple linear regression model that predicts the number of words in the summary given the number of words and utterances within the conversation.

```
Text = f"""
generate a brief abstractive summary of
the following dialogue:


{dialogue}

The summary should:
(1) be rather short,
(2) extract important pieces of informa-
tion,
(3) include names of interlocutors,
(4) be written in the third person.

the summary should contain around
{length} words.
"""
```

For internal data, we ensure better diversity in simulated conversations by randomly choosing a client's personality from a list proposed by the LLM:

```
conv = ["longue et complexe", "courte",
"longue", "complexe"].

client = [
"Le client est une personne exigeante
et pointilleuse qui veut être assurée de
recevoir un service impeccable.",
"Le client est une personne impatiente
et pressée qui veut des réponses rapides
et des solutions immédiates.",
"Le client est une personne curieuse et
posée qui pose de nombreuses questions
pour obtenir toutes les informations
nécessaires.",
"Le client est une personne insatisfaite
et mécontente qui exprime ouvertement sa
frustration et son mécontentement.",
"Le client est une personne enthousi-
aste et énergique qui se montre très
intéressée par le produit ou le service
offert.",
"Le client est une personne hésitante et
indécise qui a besoin de conseils et de
recommandations pour prendre une déci-
sion.",
"Le client est une personne confiante
et sûre d'elle qui sait exactement ce
qu'elle veut et exige un service person-
nalisé.",
"Le client est une personne émotionnelle
et sensible qui souhaite être écoutée et
comprendre que son point de vue est pris
en compte.",
"Le client est une personne frugale et
soucieuse des prix qui recherche con-
```

Training Data	mts-dialog			dialogSUM			CNN/DM		
	R1	**R2**	**RL**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
FTS	34.7/39.8	12.9/17.0	28.7/32.3	45.6/47.7	19.3/21.5	36.9/39.0	32.8/33.3	12.9/13.1	23.1/23.1
HAE	32.7/33.0	12.6/13.7	25.9/26.4	39.5/42.1	13.7/16.0	31.8/33.8	31.3/30.2	11.8/11.6	22.7/20.4
SE	31.9/33.8	11.3/12.3	25.0/25.8	40.5/41.1	15.2/16.3	32.6/33.1	28.6/29.4	11.1/11.5	19.2/19.9
SE + HAE	34.2/36.3	13.7/15.5	27.1 /29.0	43.3/44.2	17.2/18.9	34.8/35.8	32.2/33.6	12.0/13.3	22.4/23.5

Table 4: Performance comparison of BART-Base (left) and BART-Large (right) models fine-tuned on different training sets. FTS includes the full training set with human-annotated examples. HAE represents the subset of human-annotated examples from Phase 1 in our pipeline. SE denotes the simulated examples generated in Phases 4 and 5.

```
stamment les meilleures offres et les
promotions.",
"Le client est une personne fidèle et
loyale qui appelle pour exprimer sa sat-
isfaction et sa reconnaissance envers
l'entreprise.",
]

text = f"""
A partir du résumé qui suit, génère une
conversation entre un client et un con-
seiller téléphonique, en respectant les
conditions suivantes:
- La conversation doit être ran-
dom.choice(conv).
- Les interventions du client sont
précédées de "Client:  " et celles de
l'agent de "Agent:  ".
- Dans cette conversation ran-
dom.choice(client).


Résumé:  {summary}
"""
```

Fianlly to generate the summaries for the internal data simulated conversations we use the following instruction.:

```
text = f"""
Générez un résumé abstractif de cette
conversation en français en utilisant
environ {str(int(length*0.2))} mots.

{conversation}
"""
```