Detecting Ambiguous Utterances in an Intelligent Assistant

Satoshi Akasaki
LY Corporation
sakasaki@lycorp.co.jp

Manabu Sassano
LY Corporation
msassano@lycorp.co.jp

Abstract

In intelligent assistants that perform both chatting and tasks through dialogue, like Siri and Alexa, users often make ambiguous utterances such as "*I'm hungry*" or "*I have a headache*," which can be interpreted as either chat or task intents. Naively determining these intents can lead to mismatched responses, spoiling the user experience. Therefore, it is desirable to determine the ambiguity of user utterances. We created a dataset from an actual intelligent assistant via crowdsourcing and analyzed tendencies of ambiguous utterances. Using this labeled data of chat, task, and ambiguous intents, we developed a supervised intent classification model. To detect ambiguous utterances robustly, we propose feeding sentence embeddings developed from microblogs and search logs with a self-attention mechanism. Experiments showed that our model outperformed two baselines, including a strong LLM-based one. We will release the dataset.[1]

1 Introduction

With the rise of AI-powered devices, intelligent assistants such as Siri and Alexa have gained popularity. These assistants interact with users in ways that allow them to search for information, operate devices, and even engage in human-like conversations (chat).

When responding to a user request, intelligent assistants must recognize its intent and trigger appropriate modules to fulfill the request. In recent years, while various methods have been utilized to determine intent, there are still challenges in handling ambiguous intents (Figure 1). For example, the utterance "*Tokyo station*" can be taken as either a route search for a train or a map search (both of which belong to task-oriented utterances), and "*I'm hungry*" can be taken as a casual conversation

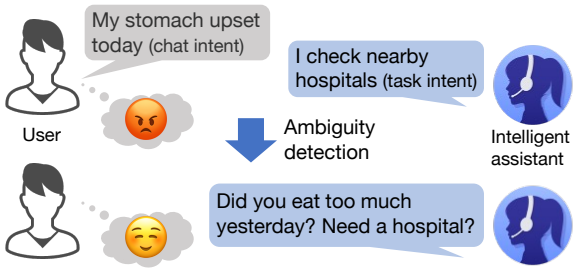

Figure 1: A dialogue with ambiguous intents. The example above results in a poor user experience because the system definitively estimates the intent of the utterances.

starter (non-task-oriented) or as a request for restaurant information (task-oriented). Such ambiguity of intent is particularly noticeable in intelligent assistants, where task-oriented and non-task-oriented utterances are mixed, and most utterances are short due to the characteristics of devices.

To address these challenges, researchers have made efforts to generate responses that help clarify the intent (Kiesel et al., 2018; Aliannejadi et al., 2019; Zamani et al., 2020). However, it is crucial first to identify which utterances require such clarification since generating clarification for every utterance is unrealistic. Furthermore, these efforts focused on task-oriented dialogue systems, and it remains unclear which types of utterances would exhibit ambiguous intents in intelligent assistants, which encompass a combination of task-oriented and non-task-oriented interactions.

Considering these, we set up the task of identifying utterances with ambiguous intents in intelligent assistants. To analyze and detect such ambiguity, we collected pairs of user utterances and system responses from the dialogue logs of a commercial intelligent assistant and labeled them using crowdsourcing. We referred to an existing dataset of intelligent assistants (Akasaki and Kaji, 2017) and

[1] https://research.lycorp.co.jp/en/softwaredata

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 386–394

assigned three labels: 'chat,' 'task,' and 'ambiguous.' This allows us to simplify the problem and flexibly consider the later process of the system. Using the dataset, we conducted an analysis to identify trends in the types of utterances that lead to ambiguous intents.

We developed the BERT-based classifier using the constructed dataset. To classify noisy utterances robustly, we fed sentence embeddings derived from large-scale search query logs and microblog logs corresponding to task intent and chat intent, respectively, into the model. We weighted those embeddings using a self-attention mechanism to consider which embedding is effective for the target utterance.

In the experiments, our method outperformed other classification models, including the resource-powerful LLM-based model, and accurately detected ambiguous utterances.

2 Related work

2.1 Domain and Intent Determination

Domain and intent determination of utterances in dialogue systems is the subject of many studies (Kim et al., 2016; Chen et al., 2019; Gangadharaiah and Narayanaswamy, 2019; Louvan and Magnini, 2020). Some studies determined ambiguous utterances by setting a threshold on the confidence of the system's domain/intent prediction. However, in multi-domain systems or intelligent assistants, it is difficult to define individual thresholds because they must be adjusted each time the number of domains/intents changes.

Some efforts focused on ambiguous utterances and determined them using supervised learning (Kim et al., 2021; Alfieri et al., 2022; Qian et al., 2022; Tanaka et al., 2023). However, those are limited to task-oriented systems and are difficult to apply to intelligent assistants. Kim et al. (2021) automatically collect ambiguous utterances by exploiting a user satisfaction metric (Kiseleva et al., 2016b,a). Specifically, they regard utterances with unsatisfactory system responses as ambiguous and collect such utterances by exploiting subsequent feedback utterances (*e.g., "Thank you," "That's wrong"*). However, in actual settings, users often output feedback utterances without meaning. This makes it challenging to collect clean training data.

Akasaki and Kaji (2017) constructed a dataset of user utterances collected from an intelligent assistant and classified them into either non-task-oriented (chat) or task-oriented (task) intents. However, their definitive labeling approach makes it challenging to handle utterances with ambiguous intents.

We address these problems by introducing an additional label to the classifier that signifies the ambiguity of the intent in the utterance. Additionally, we use the method of Kim et al. (2021) as a baseline to clarify the difficulties associated with the collection of such data.

2.2 Generating Clarification Question

There are efforts to generate clarifying questions for ambiguous utterances in dialogue systems (Kiesel et al., 2018; Aliannejadi et al., 2019; Zamani et al., 2020; Dhole, 2020). Although generating and outputting clarifying questions can resolve the ambiguity of intent, most studies focus only on the generation aspect while overlooking the critical consideration of when and to which utterances the clarification should be applied. To address this problem, Aliannejadi et al. (2021) constructed the dataset suitable for determining when a clarifying question should be asked given the current context of the conversation. Although they targeted open-domain dialogues, their focus was only on information-seeking dialogues used in search engines and did not include chit-chat. It is thus difficult to apply their approach to intelligent assistants.

Based on the situations, we determine the ambiguity of utterances in intelligent assistants, which encompasses both task-oriented and non-task-oriented interactions, for the later clarification of intents.

2.3 Intelligent Assistants

Previous studies on intelligent assistants (Kiseleva et al., 2016b,a; Sano et al., 2016, 2017) mainly investigated user behaviors, including the prediction of user satisfaction, user engagement, and reformulation. For example, Jiang et al. (2015) investigated predicting the level of user satisfaction with the responses of the system. Hashimoto and Sassano (2018) detected absurd conversations of intelligent assistant by detecting feedback utterances that show users' favorable (*e.g., "great"*) and unfavorable (*e.g., "what?"*) evaluations of system responses.

We focus on ambiguous utterances that tend to be common in intelligent assistants and try to detect them for postprocessing.

3 Detecting Ambiguous Utterances in Intelligent Assistants

This section describes the intelligent assistant handled in this paper and the task settings.

3.1 Intelligent Assistant

Examples of intelligent assistants include Apple's Siri and Amazon's Alexa. These systems use voice or text to interact with users and carry out the user's requests (Tulshan and Dhage, 2019). Although there are differences among systems, they have the typical functions of multi-domain task-oriented dialogue systems, such as web-based information retrieval (*e.g.,* weather forecast and traffic information) and terminal operation (*e.g.,* phone call and open application), as well as the capability of open-domain non-task-oriented dialogue systems, *i.e.,* human-like chatting. Therefore, responding to a broader range of requirements is necessary than the traditional dialogue systems (Kiseleva et al., 2016b). We use Yahoo! Voice Assist[2], a commercial Japanese intelligent assistant, to collect logs of dialogues.

3.2 Task Settings

We set up the task with reference to the existing domain and intent determination tasks. Existing efforts (§ 2.1) typically classify which domain an utterance belongs to or which intent is within the domain in task-oriented dialogue systems. However, since intelligent assistants are hybrids of multi-domain task-oriented and open-domain non-task-oriented dialogue systems, handling both utterances is necessary. In addition, typical dialogue systems commonly involve the classification of detailed domains or intents. However, as domains are not static but expand over time, organizing and updating training data is costly.

Considering these points, Akasaki and Kaji (2017) set up the task of determining whether a user utterance is a 'task' (task-oriented intent) or a 'chat' (non-task-oriented intent) in intelligent assistants. This allows us to mitigate the impact of changes in specifications such as domain and, if necessary, to perform a detailed categorization for each result. We follow this setting and design the problem as a multi-class classification problem, adding the label 'ambiguous' to indicate the intent of the utterance is uncertain or challenging to determine. This simplifies the problem setting and allows the system to respond accordingly if a given utterance is detected as 'ambiguous' by asking clarifying questions (§ 2.2). For example, for the case of the utterance "*My neck hurts*," by detecting it as 'ambiguous,' the system would say, "*You must be in a lot of pain. Can I help you find a hospital?*" or something like that to avoid spoiling the user experience.

We define the ambiguous utterances handled in this study as follows:

Ambiguous utterances. *Utterances for which the intention cannot be uniquely determined.*

Note that there are two types of ambiguous utterances: those that are ambiguous as to which specific task intent they belong to (*e.g., "University of Tokyo"* (a map search or a web search)), and those that are ambiguous between a task intent and a chat intent (*e.g., "I have a headache"* (a nearby hospital search or a self-disclosure of chat)). Even in the case of the former, the detailed intent cannot be uniquely determined. We thus collectively treat them as ambiguous labels.

4 Dataset

This section details the dataset construction and our analysis of the ambiguous utterances.

4.1 Construction Procedure

From dialogues between users and the system between 2014 and 2022 on Yahoo! Voice Assist,[3] we randomly collected 20,000 Japanese conversations $(u_0, r_{-1}, u_{-1}, r_{-2}, u_{-2})$ consisting of the previous system responses r_{-1}, r_{-2} and the user utterances u_{-1}, u_{-2} for the target user utterance u_0 that appeared more than 10 times. At this time, the number of identical utterances u_0 is limited to a maximum of 5. Here, we ensured privacy by removing utterances that contained personal information and finally got 17,794 conversations.

We presented the collected conversations to workers of Yahoo! Crowdsourcing (see Appendix A).[4] First, we showed a webpage explaining the intelligent assistant's functions, then asked workers to "Select the intent of u_0 in the displayed conversation from labels: chat, task, or ambiguous." We also provided examples of labeled conversations. To ensure the dataset's quality, we adopt the following policies:

[2] https://v-assist.yahoo.co.jp/

[3] We cannot disclose the detailed statistics of the original log data since it is confidential.

[4] https://crowdsourcing.yahoo.co.jp/

Label	Utterance
Chat	*Sing please.* *What is your hobby?* *Let's play word chain game.* *Do you like dogs?*
Task	*Show me a picture of cats.* *How high Mt. Fuji?* *A barber near here.* *Wake me up at 9:00.*
Ambiguous	*I'm sleepy.* *Akihabara station.* *Yahoo!'s* *My neck hurts.*

Table 1: Example user utterances (*translated*)

Label	#Examples	#Letters	#Tokens
Chat	5,123	7.61	4.20
Task	6,177	7.72	3.85
Ambig.	6,494	5.47	2.78
Total	17,794	-	-

Table 2: Dataset statistics. **#Letters** and **#Tokens** are average values.

1. Engaged only exemplary workers, selected based on their past task history provided by the service.

2. Incorporated a validation question for each task, easily answerable if workers had reviewed the instructions and examples. We accepted results only if the validation question was answered correctly.

3. Mitigated label inconsistency by assigning 10 workers to each conversation. We obtained an inter-rater agreement of 0.612 by Fleiss's Kappa, indicating substantial agreement.

We assigned the label that received the majority of votes to each conversation. In cases where no label received more than 5 votes, indicating a split decision, we assigned the 'ambiguous' label, as the lack of consensus among workers suggested ambiguity. Table 1 shows examples of utterances in the dataset. The utterances with the 'ambiguous' label can be interpreted as either task or chat intent, or any of several intents within the task.

4.2 Analysis of Ambiguous Utterances

Table 2 shows the dataset statistics. The label 'ambiguous' has the highest number, indicating that many utterances are ambiguous from a human perspective. The average number of letters and tokens [5] is relatively smaller than other dialogue systems, reflecting the nature of the intelligent assistant, which is mainly voice input for daily use. We see that ambiguous utterances are shorter than others. The omission of letters or words easily obscures intention, and the short utterances are also difficult to understand in terms of intent.

To confirm the trend of ambiguous intents in detail, we categorized utterances into seven types based on existing studies of intent classification (Meguro et al., 2014; Akasaki and Kaji, 2017) and manually classified u_0 of 1,000 randomly sampled conversations. Table 3 shows the results, revealing a prevalence of speech recognition and noun-related errors. Speech recognition errors included typical misrecognitions, inaccuracies related to kana-kanji conversion, and word omissions, leading to meaningless or confusing intents. Many nouns and phrases were difficult to parse for meaning; for example, "*Meeting space*" can refer to both route search and information search. Requests, commands, and questions were generally used for information retrieval or terminal operation but could be interpreted differently. For instance, "*Want to go home*" might indicate a desire for a chat or route search. Disclosure of information, often used in chatting, could also imply task requests. For example, "*I'm unwell*" might mean searching for a hospital or just be an icebreaker. Interesting examples like "*(^o^)*" show ambiguity in user satisfaction or emoticon use. These findings underscore the need for disambiguating intents specific to intelligent assistants.

Note that some conversations appear to suddenly shift topics with no contextual context. This is unique to intelligent assistants, a mixture of task and non-task-oriented dialogue systems and indicates that contextual information is less useless for determining intent compared to other dialogue systems.

5 Detection Method

Here, we describe our detection method. We use BERT (Devlin et al., 2019) as a base classifier $BERT(\cdot)$ and perform fine-tuning on the developed corpus to conduct multi-class classification. However, since the utterances of intelligent assistants are often short and noisy, it is difficult to classify them accurately.

To address this problem, Akasaki and Kaji (2017) improved the accuracy of an intent classification model by feeding the likelihood of the

[5] We use MeCab (https://taku910.github.io/mecab/) (ver. 0.996) with ipadic as a tokenizer.

Label	# Ex.	Example of conversation (u_0: r_{-1}: u_{-1}:) (*translated*)
Speech Recognition	341	u_0:*Today's Tokyo's* r_{-1}:*Currently, this word is becoming a hot topic...* u_{-1}:*The latest topic.* u_0:*Sin.* r_{-1}:*I sing with all my might. Den-den mushi and katatsumuri...* u_{-1}:*Sing.*
Noun	319	u_0:*Pokka Lemon* r_{-1}:*It seems that the specified app is not installed.* u_{-1}:*Open Mobage* u_0:*Meeting space* r_{-1}:*I, it's troubling to be suddenly told that.* u_{-1}:*Hug.*
Question	110	u_0:*Maybe it will rain today.* r_{-1}:*I hope I have air-conditioning.* u_{-1}:*It's hot today.* u_0:*Do you want to eat something?* r_{-1}:*It's 11:09.* u_{-1}:*What time is it now?*
Self-disclosure	82	u_0:*I'm unwell.* r_{-1}:*You can see the progress here.* u_{-1}:*Show me.* u_0:*Something is wrong with my eyes.* r_{-1}:*You're welcome.* u_{-1}:*No problem.*
Command / Request	104	u_0:*I want to go home.* r_{-1}:*Let's continue the conversation.* u_{-1}:*What should we talk about?* u_0:*I want to take a nap.* r_{-1}:*I am a Voice Assistant.* u_{-1}:*Tell me your name.*
Correction	16	u_0:*17:05* r_{-1}:*Did you sleep well? The weather in Tokyo is cloudy...* u_{-1}:*What time is it now?* u_0:*The conversation isn't continuing.* r_{-1}:*Of course.* u_{-1}:*What are you thinking?*
Other	28	u_0:*Heeheehee...* r_{-1}:*I search about ending Puzzle & Dragons.* u_{-1}:*End 'Puzzle & Dragons.'* u_0:*(^o^)* r_{-1}:*With Yahoo! Loco, you can search for various shops and...* u_{-1}:*Prison School.*

Table 3: Example of ambiguous utterances. Due to space limitations, we show the conversation up to one turn preceding u_0. One of the seven categories was assigned to each conversation.

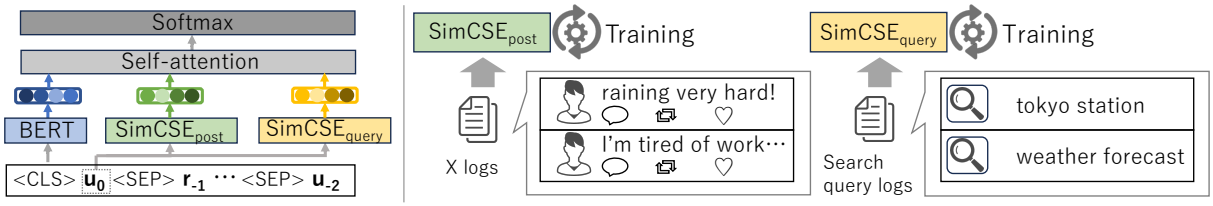

Figure 2: Overview of our detection method: we obtain the sentence embedding of utterance u_0 from two models and apply self-attention to them along with the BERT outputs.

utterance calculated using language models trained on search query logs corresponding to task intents and X logs corresponding to chat intents. We adopt a similar idea and feed features derived from the language models into the BERT model for classification. Here, we use a vector representation of utterances rather than the scalar likelihoods since the latter are less informative. Also, they input the features directly into the model, whereas we input them through a self-attention mechanism (Devlin et al., 2019) that considers the relatedness of the vector representations.

We show the overview of the proposed method in Figure 2. Specifically, we first pretrain two BERT models using search query logs and X logs, respectively. From these models, we build sentence embedding models $SimCSE_{query}$ and $SimCSE_{post}$ using unsupervised SimCSE (Gao et al., 2021), which has contributed to improving the accuracy in various NLP tasks in the past.

At the detection phase, we apply the self-attention mechanism to $SimCSE(u_0)_{query}$, $SimCSE(u_0)_{post}$ and the output of $BERT((u_0, r_{-1}, u_{-1}, r_{-2}, u_{-2}))$ as:

$$\alpha_{ij} = \frac{\exp(\sigma(W_x x_{ij} + b_x))}{\sum_j \exp(\sigma(W_x x_{ij} + b_x))} \quad (1)$$

$$x_{ij} = \tanh(W_e e_i + W_e e_j + b_e) \quad (2)$$

$$\acute{e}_i = \sum_j \alpha_{ij} e_j \quad (3)$$

$$o = [\acute{e}_i; ...; \acute{e}_N] \quad (4)$$

This method first obtains the similarity x_{ij} between e_i and e_j, where each e represents a BERT embedding and sentence embedding. We use additive attention that consists of a feed-forward network to calculate those alignment scores. We then compute the importance weight α_{ij} using the softmax function. The resulting $\acute{e}_i$ are concatenated and used as output o. This output is input to the following softmax layer for 3-label classification. This captures the relationships and importance of each embedding to utterance u_0 and enables robust detection of utterances with ambiguous intents while considering the task-specific and chat-specific nature of intelligent assistants.

6 Experiments

In this section, we build several intent classifiers and investigate their performances.

6.1 Comparison Methods

AllAmbiguous: Outputs an 'ambiguous' label for all the utterances.

Threshold: A method that judges ambiguity based on a threshold of the system. Using the dataset constructed by Akasaki and Kaji (2017), which classifies whether the utterance is a chat or a task intent, we fine-tune BERT to perform binary classification. At the test time, for the label with the largest softmax score, we output an 'ambiguous' label when its score is below the threshold, which was determined using the development data.

Feedback: The method used by Kim et al. (2021) collects ambiguous utterances based on user feedback. Specifically, when a user provides negative feedback utterance u_1 (*e.g.,* "*what?*", "*It's wrong*") to the system's response r_0, it assumes that the intent of the preceding user utterance u_0 is ambiguous. We identify negative feedback utterances using Hashimoto and Sassano (2018)'s method and label the preceding dialogues $(u_0, r_{-1}, u_{-1}, r_{-2}, u_{-2})$ as 'ambiguous,' while labeling the remaining dialogues as 'non-ambiguous.' We then fine-tune BERT for binary classification using the collected data.

GPT-4o: Recent advances in various NLP tasks have shown success with LLMs. We use GPT-4o (ver. 202405) for few-shot classification, providing prompts and labeled examples (see Appendix B) to classify utterances.

SVM: We train support vector machine (SVM) (Cortes and Vapnik, 1995) using the dataset for multi-class classification. We employ tf-idf calculated from the training data as features. We vectorize the utterance u_0, vectorize and then average the remaining $(r_{-1}, u_{-1}, r_{-2}, u_{-2})$, and concatenate them.

BERT: We fine-tune BERT using the dataset for multi-class classification. Each utterance and response of $(u_0, r_{-1}, u_{-1}, r_{-2}, u_{-2})$ is concatenated by $[SEP]$ tags and input to the BERT encoder. To investigate the impact of model size, we conduct experiments using both the base and large models.

Proposed: We fine-tune BERT with the proposed methods (§ 5) using the dataset.

6.2 Settings

The BERT models for classification were pretrained using the default settings of *bert-base-cased* and *bert-large-cased* respectively on 18 million Japanese Wikipedia sentences from February 2021.[6] We finetuned the BERT models using hyperparameters in Table 4, and used the model with the highest F_1-score in the development data. For **Threshold**, we use 15,600 conversations derived from Akasaki and Kaji (2017). For **Feedback**, we applied the method to the data of Yahoo! Voice Assist and sampled 100,000 conversations. For **SVM**, we perform L2-regularized linear SVM and the C parameter is tuned using the development data. We used 10-fold cross-validation to tune and evaluate the models. We implemented the models using Python3 and Tensorflow2.

Parameter	Value
Epoch	10
Sentence length	128
Batch size	16
Dropout rate	0.1
Learning rate	2e-5
Weight decay rate	0.01
Dimensions of sentence embedding	768
Number of head for self-attention	8
Optimizer	AdamW
Tokenizer	SentencePiece

Table 4: Hyperparameters of the BERT models.

For sentence embeddings, we used 50 million top-frequent Japanese web search[7] queries from July 2021 to July 2022 and 50 million randomly sampled Japanese tweets[8] from the same period to pretrain BERT models with the settings of *bert-base-cased*. Using each model, we finally trained 1 epoch of unsupervised SimCSE with recommended parameters.

6.3 Results

Table 5 shows the overall result of classification. While the performance of baseline methods that do not utilize the constructed dataset is poor, the performance of **SVM**, **BERT** and **Proposed** is better, demonstrating the necessity of labeled data. There is no significant performance difference between **SVM** and **BERT**, suggesting that due to the short and noisy nature of the utterances, there is a limit to performance improvement, whether using simple text features or employing large-scale models. We see that **Proposed** achieved the highest performance. This indicates that utilizing external knowledge of X and search query logs with the

[6]Note that *bert-large-cased* is only used in **BERT** for the large model.

[7]We use Yahoo! JAPAN (https://www.yahoo.co.jp/) as the search engine. Due to the confidential nature of the data, we cannot disclose detailed statistics, but the number of unique search queries amounts to approximately 8.1 billion annually.

[8]We use the complete set of tweet data provided under a contract with X Corp. Due to the terms of the contract, detailed statistics are confidential.

	Acc.	Prec.	Rec.	F_1
AllAmbiguous	36.52	12.17	33.33	17.83
Threshold	69.38	68.53	70.58	68.09
GPT-4o	68.61	68.28	69.62	68.21
SVM	76.52	76.50	77.06	76.72
BERT (base)	77.53	77.32	78.27	77.48
BERT (large)	77.64	77.49	78.49	77.63
Proposed	**79.10**	**79.12**	**79.72**	**79.31**

Table 5: Overall performances. **Proposed** outperforms all comparisons significantly (measured by the Wilcoxon rank-sum test with p-value < 0.05). We excluded **Feedback** because the model only outputs either 'ambiguous' or not.

self-attention mechanism is effective for the intent detection task in intelligent assistants.

Table 6 shows the F_1-scores for each label. **SVM**, **BERT** and **Proposed** outperformed the other methods, indicating that they can learn the tendency of utterances, including ambiguous intents, by utilizing the constructed dataset. Even when **Threshold** achieved a moderate performance in Table 5, its F_1-score of the 'ambiguous' label was notably low. We observed that it could hardly output the 'ambiguous' label, indicating the difficulty of making ambiguity judgments based on the threshold. Although **Feedback** learned from the data derived from negative feedback utterances, its F_1-score of the 'ambiguous' label was still low, indicating that the collected training data actually contained a lot of noise. Despite being a larger model than other models, **GPT-4o** exhibits lower performances. This might be because LLMs find it difficult to understand the concept of ambiguity. It also indicates the need to use knowledge outside the dialogue, as in the **Proposed**, to complement the clues. Among the models using labeled data, **SVM** shows the lowest performance due to insufficient expressiveness. Interestingly, despite the difference in model sizes, there is no significant performance difference between the base and large model of **BERT**. This can be attributed to the short length of the target utterances, which may prevent the large model from fully leveraging its capabilities. **Proposed** outperformed **SVM** and **BERT** in all labels, but particularly the gain in the 'ambiguous' label was high, exceeding 3%. This indicates that the introduced sentence embeddings and self-attention mechanism effectively detect ambiguous utterances. Overall, these findings demonstrate the effectiveness of the annotated dataset and our proposed method.

	Chat	Task	Ambiguous
AllAmbiguous	–	–	53.50
Threshold	75.57	80.09	48.62
Feedback	–	–	40.69
GPT-4o	69.72	79.80	55.11
SVM	80.49	82.10	67.57
BERT (base)	80.33	83.73	68.39
BERT (large)	80.54	84.14	68.17
Proposed	**82.26**	**84.19**	**71.53**

Table 6: F_1-scores by label for each method.

6.4 Qualitative Analysis

We checked the output of **Proposed** and confirmed that it detected utterances with speech recognition errors more accurately than other errors. Such utterances are relatively easy to detect by capturing features such as character omissions. Additionally, **Proposed** could detect ambiguous utterances such as "*I want to go for a drink* (Command / Request)", which could either be looking for a bar or just an expression of desire, by considering the sentence embeddings and self-attention mechanism.

We found many errors in detecting utterances corresponding to the 'noun' label in Table 3. They are usually short utterances with only one noun (phrase) and are challenging to handle even with a large-scale model. For example, "*Shinagawa* (place name)" is likely to be an ambiguous task request for searching train routes, maps, or web information, while "*Lexus* (car name)" is likely to be unambiguous because the only applicable task request is web search. To distinguish such examples, it is necessary to incorporate detailed external knowledge about nouns, for example, recognizing that "Lexus" is a car brand, or implement a process that outputs a clarification question whenever the utterance is a single noun.

7 Summary

We focused on detecting ambiguous utterances in intelligent assistants. Using crowdsourcing, we labeled real log data and analyzed trends in the dataset. To robustly detect ambiguous utterances, we proposed using sentence embeddings from external resources with a self-attention mechanism. Experiments showed the effectiveness of our dataset and method.

We plan to integrate our method along with a module of clarification questions into the actual system. This improves user experience and allows us to gather feedback from users. We will release the dataset to facilitate future studies.

8 Ethics Statement

To maximize the privacy of the users from whom the dataset was derived, we limited the user utterances included in the collected conversations to those that appeared at least 10 times in the logs. In addition, we carefully checked whether these utterances contained personal information such as person names, addresses, and telephone numbers and removed conversations containing such utterances.

Acknowledgments

The author would like to thank the anonymous reviewers, who provided insightful comments.

References

Satoshi Akasaki and Nobuhiro Kaji. 2017. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Andrea Alfieri, Ralf Wolter, and Seyyed Hadi Hashemi. 2022. Intent disambiguation for task-oriented dialogue systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 5079–5080.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval (SIGIR)*, pages 475–484.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL-HLT)*, pages 4171–4186.

Kaustubh D Dhole. 2020. Resolving intent ambiguities by retrieving discriminative clarifying questions. *arXiv preprint arXiv:2008.07559*.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL-HLT)*, pages 564–569.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chikara Hashimoto and Manabu Sassano. 2018. Detecting absurd conversations from intelligent assistant logs by exploiting user feedback utterances. In *Proceedings of the 2018 World Wide Web Conference (WWW)*, pages 147–156.

Jiepu Jiang, Ahmed Hassan Awadallah, Rosie Jones, Umut Ozertem, Imed Zitouni, Ranjitha Gurunath Kulkarni, and Omar Zia Khan. 2015. Automatic online evaluation of intelligent assistants. In *Proceedings of WWW*, pages 506–516.

Johannes Kiesel, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen. 2018. Toward voice query clarification. In *Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval (SIGIR)*, pages 1257–1260.

Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. Intent detection using semantically enriched word embeddings. In *Proceedings of the 2016 IEEE SLT Workshop*.

Joo-Kyung Kim, Guoyin Wang, Sungjin Lee, and Young-Bum Kim. 2021. Deciding whether to ask clarifying questions in large-scale spoken language understanding. In *Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 869–876. IEEE.

Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan Crook, Imed Zitouni, and Tasos Anastasakos. 2016a. Predicting user satisfaction with intelligent assistants. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval (SIGIR)*, pages 45–54.

Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan C. Crook, Imed Zitouni, and Tasos Anastasakos. 2016b. Understanding user satisfaction with intelligent assistants. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval (CHIIR)*, pages 121–130.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 480–496.

Toyomi Meguro, Yasuhiro Minami, Ryuichiro Higashinaka, and Kohji Dohsaka. 2014. Learning to control listening-oriented dialogue using partially observable markov decision processes. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(4):1–20.

Kun Qian, Satwik Kottur, Ahmad Beirami, Shahin Shayandeh, Paul A Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2022. Database search results disambiguation for task-oriented dialog systems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL-HLT)*, pages 1158–1173.

Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. 2016. Prediction of prospective user engagement with intelligent assistants. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1203–1212.

Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. 2017. Predicting causes of reformulation in intelligent assistants. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 299–309.

Shohei Tanaka, Koichiro Yoshino, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Reflective action selection based on positive-unlabeled learning and causality detection model. *Computer Speech and Language*.

Amrita S Tulshan and Sudhir Namdeorao Dhage. 2019. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *Proceedings of the Advances in Signal Processing and Intelligent Recognition Systems (SIRS)*, pages 190–201. Springer.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of the Web Conference 2020 (WWW)*, pages 418–428.

A Crowdsourcing

We used Yahoo! Crowdsourcing to annotate conversations (§ 4.1). This is a Japanese crowdsourcing service with over 3 million users. The service has a unique list of excellent workers compiled from the users' past task histories. By utilizing this list, it is possible to allow only superior workers to participate in tasks in advance. We utilized this service to pay the superior workers a reward of 15 yen ($0.1) for every set of 10 conversation annotations.

B Settings of GPT-4o

Here, we describe the settings of GPT-4o used in the paper. We use GPT4-o on 1 June 2024; the temperature is set to 0. The following prompt is used for classification (§ 6). For few-shot classification, we give 10 examples of training data.

```
*We provide a dialogue of an intelligent assistant, and we would like you to assign a specific label to them.

*We provide the conversations chronologically, where 'U' denotes the user's utterance and 'R' denotes the system's response.

*Read the conversation and determine which of the following labels it belongs to.

*Labels:

  • Chat: The user wants to do casual conversation with the assistant, such as "Good morning," "Sing," "I like you," and "Let's chat"
  • Task: The user intends to search for information or perform device operations, such as "Yahoo stock price," "Today's economic news," "Dog videos," "Open LINE," and "Alarm"
  • Ambiguous: The intent is ambiguous due to various reasons like speech recognition error and can fit into either Chat or Task, such as "Tokyo station," "I have a stomach ache," and "Today's Tokkyo"

*U0: [UTTERANCE]

*R1: [RESPONSE]

*U1: [UTTERANCE]

*R2: [RESPONSE]

*U2: [UTTERANCE]

*Label: [LABEL]
```