Last Utterance Proactivity Prediction in Task-oriented Dialogues

Sofia Brenna[1,2,*], Bernardo Magnini[2]

[1]Free University of Bozen-Bolzano, 3 Dominikanerplatz 3 - Piazza Domenicani 3, Bozen-Bolzano, 39100, Italy
[2]Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento, 38123, Italy

Abstract

While current LLMs achieve excellent performance in information seeking tasks, their conversational abilities when participants need to collaborate to jointly achieve a communicative goals (e.g., booking a restaurant, fixing an appointment, etc.) are still far from those exhibited by humans. Among various collaborative strategies, in the paper we focus on *proactivity*, i.e., when a participant offers useful information that was not explicitly requested. We propose a new task, called *last utterance proactivity prediction* aimed at assessing the capacity of an LLM to detect proactive utterances in a dialogue. In the task, a model is given a small portion of a dialogue (that is, a *dialogue snippet*) and asked to determine whether the last utterance of the snippet is proactive or not. There are several benefits in using dialogue snippets: (i) they are more manageable than full dialogues, allowing to reduce complexity; (ii) several phenomena in dialogue, including proactivity, depend on a short context, which allows a model to learn from snippets, rather than full dialogues; and (iii) dialogue snippets make it easier to experiment on balanced datasets, overcoming the skew distribution of proactivity in whole dialogues. In the paper, we first introduce a dataset for the last utterance proactivity prediction task. The dataset has then been used to instruct an LLM to classify proactivity. We run a series of experiments showing that predicting proactive utterance in a dialogue is feasible in a few-shot configuration, opening the road towards models that are able to generate proactive utterances like humans do.

Keywords

task-oriented dialogues, pragmatics, proactivity, automated annotation, large language models

1. Introduction

While current Large Language Models (LLMs) achieve excellent performance in information seeking tasks, their conversational abilities when participants need to collaborate to jointly achieve a communicative goals (e.g., booking a restaurant, fixing an appointment, etc.) are still far from those exhibited by humans. In the paper, we specifically focus on *proactivity* [2, 3, 4, 5], a collaborative behaviour investigated in the context of dialogue pragmatics [6]. Proactivity refers to the act of taking initiative, anticipating potential problems, and actively providing information and contributing to the conversation with ideas, suggestions or solutions. Proactivity involves participants actively participating in the dialogue, addressing concerns, and promoting a collaborative environment. The following is an example of a dialogue in which proactive utterances are underlined.

a: Hai qualche preferenza riguardo al luogo di lavoro?
b: Dopo aver fatto la triennale a Roma, mi piacerebbe tornare verso casa, a Firenze.
a: Al momento non abbiamo nessun annuncio che faccia al caso tuo nella zona di Firenze però ci sono delle opportunità di lavoro su Roma.[1]

NL4AI 2024: Eighth Workshop on Natural Language for Artificial Intelligence, November 26-27th, 2024, Bolzano, Italy [1]
*Corresponding author.
sbrenna@fbk.eu (S. Brenna); magnini@fbk.eu (B. Magnini)
0009-0001-3748-1448 (S. Brenna); 0000-0002-0740-5778 (B. Magnini)

[1]Example taken from the JILDA Corpus [7]. It may be translated to English as follows:

a: Do you have any preference about where to work?
b: After completing my Bachelor's degree in Rome,
I would like to move back towards home, to Florence.
a: We currently do not have any offers that fit your needs in the Florence area,
however, there are job opportunities in Rome.

In order to model proactivity, we follow a similar approach as Shaikh et al., which focuses on grounding acts, a class of collaborative behaviors investigated in dialogue pragmatics. The main idea is that grounding acts can be: (i) identified and annotated by a Large Language Model; (ii) modeled through appropriate fine tuning of the model itself. In addition, our work is related to recent approaches that use large language models as annotators [9], [10], [11]. In the long-term, our research goal is to instruct LLMs to be as proactive as humans are.

2. Last Utterance Proactivity Prediction

The goal of the paper is to show the feasibility of automatic detection of proactive utterances in task-oriented dialogues. We propose a task, *Last Utterance Proactivity Prediction*, where a portion of a dialogue (i.e., a dialogue snippet) is given to a model, which has to predict whether the last utterance of the snippet is proactive or not proactive. Using dialogue snippets, instead of full dialogues, brings several benefits: (i) dialogue snippets are much more manageable than full dialogues, allowing us to reduce the complexity of understanding and annotation; (ii) several phenomena in dialogue, including proactivity, depend on a short context, which allows a model to learn from snippets, rather than full dialogues; and (iii) dialogue snippets make it easier to experiment on balanced datasets, overcoming the skew distribution of proactivity in whole dialogues (it has been estimated that about 85% of the utterance in a task-oriented dialogue is not proactive).

We started with D-PRO[2], a corpus of manually annotated task-oriented dialogues, which includes 151 dialogues from different sources, amounting to 2,855 turns and over 6,000 utterances, and carried out the following steps:

- we transformed the whole-dialogue annotation task to a one-utterance annotation task: given a short dialogue context, the model needs to establish whether the final utterance is either 'proactive' or 'not_proactive';
- in order to shorten the provided dialogue context, we collected 4 conversational turns' worth of excerpts (*snippets*) from each dialogue. We believe 4 turns to be a convenient context for proactivity annotation since statistics in the D-PRO Corpus on turn-adjacency between proactive utterances and the turn that triggers them revealed that an average of 77.7% proactive utterances are a direct response to the previous turn's utterances;
- to restore balance among labels we choose the same number of snippets that ended without proactivity as the snippets that ended with proactivity.

A relevant consequence of the reduction of the provided dialogue context, is a significant reduction of the input prompt length for a LLM, and therefore a reduction of computational need.

3. Experimental Setting

This section reports the main features of the setting we used to experiment the last utterance proactivity task introduced in Section 2.

Dataset for the experiments. The dataset for the experiments has been derived from D-PRO, a corpus equipped with manually curated proactivity-oriented annotations. D-PRO comprises 151 dialogues from 5 task-oriented dialogue sub-corpora, namely, Italian Whatsapp Corpus ([12]), the Italian Nespole! Corpus ([13, 14]), Jilda ([7, 15]), the Italian Ubuntu Chat Corpus ([16]), and Multiwoz 2.2 ([17]). Most of the dialogues are in Italian, with the only exception of the Multiwoz 2.2 dialogues and some dialogues from the Italian Whatsapp Corpus due to code mixing and code switching employed by the speakers. D-PRO proactivity annotations are performed at the utterance level.
The composition of our experimental dataset is as follows: from D-PRO we gathered as many 4-turn

[2]https://github.com/sofiabrenna/dpro

```
1 You must answer a closed-ended question. You will be asked to say whether the final utterance of a piece of
  dialogue between an Agent and a Client is proactive or not_proactive. An utterance in the dialogue is
  'proactive' if the speaker takes the initiative offering information that was not explicitly requested in the
  previous utterances (for instance with a request). An utterance is 'not_proactive' if the speaker just reacts
  to previous utterances. Output exclusively either 'proactive' or 'not_proactive'.
  Here are example dialogues with answers:
2 Example Dialogue:
  U5    Agent: Thanks for that information.
  U6    Agent: What area are you looking to stay in?
  U7    Client: The area doesn't matter
  U8    Client: I would like an expensive hotel if you can find one.
3 Last utterance proactivity: proactive
4 Example Dialogue:
  [...]

5 Is the final utterance 'proactive' or 'not_proactive'?
6 Target Dialogue:
  U17   Client: There are 6 people,
  U18   Agent: The table will be reserved for 15 minutes.
  U19   Agent: Reference number is: EJAOP5QE.
  U20   Client: Thanks for your help!
  U21   Agent: You are quite welcome.
7 Last utterance proactivity:
8 Model output: not_proactive
```

```
1 = system prompt
2 = user prompt → few-shot snippets
3 = assistant prompt → label
4 = user prompt → few-shot snippets
5 = user prompt → task
6 = user prompt → target snippet
7 = assistant prompt → label trigger
8 = model output
```

Figure 1: Prompt given to the LLM.

proactive dialogue snippets as there were proactive utterances, so that each snippet ended with a different proactive utterance. Then, we extracted as many non-proactive snippets as there were non-proactive utterances, so that each snippet ended with a different non-proactive utterance. Finally, we selected an equal amount of non proactive dialogue snippets as proactive ones at random to restore balance between the two types of snippets.

Data splitting. From each of the 5 corpora in D-PRO we randomly selected 30 dialogue snippets as a train set (to be used as few-shot examples), 50 snippets as a validation set (to be used for parameter optimization), and 100 snippets as a test set[3].

Model. For the choice of the best model for proactivity prediction we carried out a number of experiments, reported in Section 4.1. The model selected is Openai's GPT-4o-2024-08-06, used with temperature = 0.

Prompt optimisation. A prompt engineering phase took pace, so that various prompt proposals were tested in the same setting (the same train snippets used as few-shot examples, same validation snippets used to evaluate the model). Figure 1 shows the final prompt used in our experiments. The prompt consists of two main parts: (i) the system prompt, which contains the general task instructions given to the model; (ii) the messages prompt, that is further dividend into alternating user messages and assistant messages: this is the part of the prompt where the model receives few-shot examples (user messages) with answers (assistant message). The final user/assistant pair contains the target dialogue which is being evaluated by the model at current time.

Baseline. A random chance baseline (accuracy = 0.5, see Table 8) is created by eliminating any system and message prompt except for "Output exclusively either "proactive" or "not_proactive." and providing the target dialogue snippet.

4. Parameter Setting

We first present several optimization trials performed on a single corpus (MultiWoz) in order to select the best LLM (4.1), to assess the impact of the number (4.2) and of the order (4.3) of few-shot example

[3]The set sizes were established on the capacity of the least proactive sub-corpus, MultiWOZ, which featured 90 proactive utterances, hence 90 proactive snippets and 90 non-proactive snippets.

Table 1
Testing across various GPT Models though Openai APIs.

Metric	gpt-4o-2024-05-13	gpt-4o-2024-08-06	gpt-4o-mini	gpt-4o-mini-2024-07-18	**St.Dev.**
Total Accuracy	0.64	0.74	0.70	0.68	0.04
Total Precision	0.67	0.88	0.73	0.71	0.09
Total Recall	0.56	0.56	0.64	0.60	0.04
Total F1 Score	0.61	0.68	0.68	0.65	0.03

Table 2
Testing proactivity prediction with different numbers of few-shot dialogue snippets on the DEV set.

Few-shot examples	0	10	12	15	20	25	30	St.Dev.
Total Accuracy	0.66	0.72	0.76	0.74	0.72	0.72	0.70	0.02
Total Precision	0.63	0.79	0.81	0.88	0.82	0.82	0.78	0.04
Total Recall	0.76	0.60	0.68	0.56	0.56	0.56	0.56	0.06
Total F1 Score	0.69	0.68	0.74	0.68	0.67	0.67	0.65	0.03

Table 3
Testing proactivity prediction with different orders of few-shot dialogue snippets on DEV set. The variability of the results is measured with standard deviation (St.Dev.).

Shuffles	0	1	2	3	4	5	St.Dev.
Total Accuracy	0.76	0.66	0.68	0.72	0.74	0.66	0.04
Total Precision	0.84	0.79	0.71	0.76	0.80	0.70	0.05
Total Recall	0.64	0.44	0.60	0.64	0.64	0.56	0.10
Total F1 Score	0.73	0.56	0.65	0.70	0.71	0.62	0.07
Cohen's Kappa	0.55	0.36	0.39	0.47	0.51	0.35	0.09

snippets. Secondly, we run test on the DEV set of each of the five corpora in order to select the best few-shot snippets order for each corpus (4.4).

4.1. Setting the Large Language Model

Once the best prompt had been established, we tested the APIs of various models to pick the best cost/performance trade-off, reported in Table 1. GPT-4o-2024-08-06 was selected as the best performing model (Accuracy: 0.74, F1: 0.68) with lower fares than GPT-4o-2024-05-13 and better scores than both GPT-4o-mini and GPT-4o-mini-2024-07-18.

4.2. Setting the Number of Few-shot Dialogue Snippets

Table 2 reports the performance of the LLM on the DEV set augmenting (from 0 to 30) the amount of few shot dialogue snippets. It can be noted that increasing the few shots examples till 15 increases the precision of the model, while more examples results in worse precision. On the other side, the highest recall is obtained with 0 examples (resulting in more false positive cases). The best accuracy (0.76) and F1 Score (0.74) are obtained with 12 examples.

4.3. Assessing the Impact of Few-shot Dialogue Snippets Order

This experiment assesses the stability of the model while changing the order of the few-shot examples. As literature points out [18, 19, 20], LLMs suffer of a position bias when handling a longer context: we found that this is the case also in our experiments and that there is up to 10 points of a difference in accuracy when testing with different orders of the same set of examples. Table 3 reports the results of the experiments under six random changes of 12 examples.

Table 4
Optimising proactivity prediction with different few-shot dialogue snippets order on each dataset DEV set. Configuration: 12 few-shot snippets and 50 validation snippets. s-n indicates re-shuffled sets of 12 few-shot snippets; highest scores across re-shuffles are marked in bold. Results that show a statistically significant increase compared to the average ($p < 0.05$) are highlighted in green.

Dataset	Metric	s-0	s-1	s-2	s-3	s-4	Average	St.Dev.
Whatsapp	Accuracy	0.66	**0.72**	0.62	**0.72**	0.68	0.68	0.05
	Precision	0.70	**0.87**	0.69	0.82	0.76	0.77	0.09
	Recall	**0.56**	0.52	0.44	**0.56**	0.52	0.52	0.06
	F1 Score	0.62	0.65	0.54	**0.67**	0.62	0.62	0.06
	Cohen's Kappa	0.31	**0.43**	0.23	**0.43**	0.35	0.35	0.10
Nespole	Accuracy	0.84	**0.86**	0.80	0.80	0.82	0.82	0.03
	Precision	**0.84**	0.80	0.78	0.76	0.81	0.80	0.03
	Recall	0.84	**0.96**	0.84	0.88	0.84	0.87	0.06
	F1 Score	0.84	**0.87**	0.81	0.81	0.82	0.83	0.03
	Cohen's Kappa	0.67	**0.72**	0.59	0.59	0.63	0.64	0.06
Ubuntu	Accuracy	**0.68**	0.64	**0.68**	0.64	**0.68**	0.66	0.02
	Precision	0.68	0.65	0.67	0.65	**0.71**	0.67	0.02
	Recall	0.68	0.60	**0.72**	0.60	0.60	0.64	0.06
	F1 Score	0.68	0.62	**0.69**	0.62	0.65	0.65	0.04
	Cohen's Kappa	**0.35**	0.27	**0.35**	0.27	**0.35**	0.32	0.05
Jilda	Accuracy	0.70	**0.74**	**0.74**	0.70	0.68	0.71	0.02
	Precision	0.69	0.75	0.73	**0.78**	0.74	0.74	0.04
	Recall	0.72	0.72	**0.76**	0.56	0.56	0.66	0.09
	F1 Score	0.71	0.73	**0.75**	0.65	0.64	0.70	0.04
	Cohen's Kappa	0.39	**0.47**	**0.47**	0.39	0.35	0.41	0.05
Multiwoz	Accuracy	**0.82**	0.76	0.76	0.76	0.66	0.75	0.03
	Precision	**0.94**	0.88	0.81	0.81	0.79	0.85	0.06
	Recall	**0.68**	0.60	**0.68**	**0.68**	0.44	0.62	0.04
	F1 Score	**0.79**	0.71	0.74	0.74	0.56	0.71	0.03
	Cohen's Kappa	**0.63**	0.51	0.51	0.51	0.30	0.49	0.06

4.4. Setting Few-Shot Snippets Order

In this experiment we select the best order of the few-shot snippets for each of the five dialogue datasets (i.e., Whatsapp, Nespole, Ubuntu, Jilda and MultiWoz) for the last utterance proactivity prediction task. We tested 5 different orders of the dialogue snippets randomly shuffling the same set of snippets selected in section 4.2. We used the following configuration for the experiments: 12 random few-shot snippets; 50 validation (DEV) snippets; 5 random shuffles of the few-shot snippets; average of the performances of the 5 shuffles for each corpus. The selection of the optimal order is given by the highest average accuracy and F1.

5. Results and Discussion

In this section we present the results of the last utterance proactivity prediction task in two different configurations: using few-shots from individual corpora (5.1), and mixing few-shots from all corpora (5.3), introducing transfer learning as well. Lastly, we describe an experiment in Section 5.4 that attempts to evaluate the model's stability in corrupted context scenarios.

5.1. Few-Shots from Individual Corpus and Testing on Individual Corpus

We tested the model by using the prompt and few-shot configuration that gave the best results on the development set for each corpus individually. As Table 5 shows, while IAA with the ground truth labels is still not optimal, and scores pretty low on both Whatsapp and Ubuntu (*fair agreement*[4]), we reach a

[4]According to Landis and Koch's scale [21].

Table 5
Individual corpus testing. For comparison, DEV Average column reports averaged results over the five corpora on the developement set.

Metric	Whatsapp	Nespole	Ubuntu	Jilda	Multiwoz	Average	DEV Average
Accuracy	0.64	0.86	0.64	0.75	0.77	0.73	0.73
Precision	0.69	0.85	0.68	0.74	0.81	0.75	0.76
Recall	0.50	0.88	0.52	0.78	0.69	0.67	0.66
F1 Score	0.58	0.86	0.59	0.76	0.75	0.71	0.70
Cohen's Kappa	0.27	0.72	0.27	0.50	0.54	0.46	0.44

Table 6
Testing the model with few-shot examples taken from all five corpora. Results are given for the best inter-corpus order over five runs.

Metric	Whatsapp	Nespole	Ubuntu	Jilda	Multiwoz	Average
Accuracy	0.63	0.87	0.66	0.69	0.77	0.72
Precision	0.74	0.88	0.74	0.69	0.78	0.77
Recall	0.4	0.86	0.5	0.6	0.68	0.62
F1 Score	0.52	0.87	0.6	0.69	0.76	0.68
Cohen's Kappa	0.27	0.74	0.32	0.37	0.54	0.44

moderate agreement in Jilda and Multiwoz, and a *substantial agreement* in Nespole (0.72) that is just below the IAA score between human annotators (0.77). Results are consistent with the outcomes that we obtained on the development set in Table 4. On average, Nespole achieved the best accuracy (0.86), followed by MultiWoz (0.77), Jilda (0.75), Whatsapp and Ubuntu (both 0.64). For all dateset the results are largely above the baseline (i.e., 0.50 accuracy, equivalent to chance: see also Table 8, Baselines - Full Context column.), showing that the model has correctly learned our definition of proactivity. The fact that Nespole has obtained the best results is somehow surprising, given that this corpus is quite complex: utterances are longer than in the other corpora, and so are dependencies between a proactive utterance and its own trigger utterance. Longer utterances, on the other hand, mean that the model is given a slightly richer context on which to base its judgments, which may help with the annotation process. As far as lowest scores are concerned, the poor results for Whatsapp and especially Ubuntu were expected, since these are less structured (both syntactically and grammatically), more chaotic, and multi-party dialogue corpora, where proactivity is much more difficult to be unanimously detected also by humans and where the human-human IAA scores the lowest (0.63 and 0.41 respectively).

5.2. Few-Shots from All Corpora and Testing on Individual Corpus

Secondly, we run some in-context learning [22] experiments to check whether few-shot examples from different corpora could improve the performance on one individual target corpus. The idea is drawn from works on multi-task learning [23], where more than one task is learned simultaneously by the model, and transfer learning, where improvement is obtained in a new task through the transfer of knowledge from a related task that has already been learned [24, 25]. Our intuition is that example variety on very similar tasks may be the key to improvement on single target task. Following this line, we combine dialogue snippets from all the five corpora as few-shot examples, so that we have 5 sets of 12 snippets each. Given the position bias hold by the model, we decided to keep the intra-corpus examples order the same as the one used in 5.1, and to randomly shuffle the inter-corpus order 5 times to select the optimal few-shot prompt (same methodology in 4.4 and 5.1).

The outcomes of the tests with the optimal prompt are reported Table 6, that is directly comparable to Table 5. We found out that the only corpus in our experiment that suffered the mixed few-shot

Table 7
Best and average results over five runs with different inter-corpus order.

Few-shot examples	60-best	15-best	60-average	15-average
Accuracy	0.71	0.69	0.69	0.68
Precision	0.75	0.72	0.74	0.73
Recall	0.63	0.60	0.59	0.56
F1 Score	0.69	0.66	0.66	0.63
Cohen's Kappa	0.42	0.38	0.38	0.36

prompting is Jilda, with a significant drop in performance, while every other corpus has very similar or slightly higher scores. Ablation tests on the Jilda corpus led to an accuracy of 0.71 while removing the most chaotic corpora (Ubuntu and Whatsapp), proving still that the individual corpus few-shot approach works best for this one corpus.

5.3. Few-Shot from All Corpora, Testing on All Corpora

We finally tested on the cumulative test set of all corpora, with mixed few-shot examples. We experimented with two configurations of few-shot snippets: i. 60 snippets in optimal order as in Table 6; ii. 15 snippets in total, with 3 random snippets per corpus. Outcomes in Table 7 show best and average results over 5 runs for both the 60 and the 15 few-shot examples setting.

5.4. Testing with Corrupted Context: Masking the Trigger Utterance

We investigated the performance of the LLM in a corrupted context situation. According to definition, the two key characteristics of proactivity are not being solicited and being beneficial to the dialogue goals, hence proactivity can only be defined in terms of the previous context. When we corrupt the context before the snippet's final utterance, we may be compromising the data required for the proactivity annotation task. We implemented two corrupted context situations: *triggering utterance removed*, where the text of the utterance that triggers the final utterance in the dialogue snippet is removed from the snippet, and *triggering utterance masked*, where the text of the trigger utterance is masked by a placeholder. In both circumstances, the presence of a corrupted utterance is indicated by the utterance number, whereas only the content is erased or masked. Since we need to test the effect of the context corruption, the model still learns the original full context task from the few-shot examples. We anticipate that the LLM's performance will suffer since a critical component of the dialogue (the triggering utterance) has been compromised. Also, we expect the number of false positives to increase significantly. This is due to the possibility that by eliminating the trigger utterance, we will also eliminate the element that renders the final utterance either proactive or non-proactive. Specifically, we are deleting the element of the context that allows us to determine whether the content of the last utterance is novel (i.e., proactive) or unrequested by the trigger. Since the request is missing from the corrupted setting, the triggered response seems to be proactive rather than solicited, resulting in an increase in "proactive" labels.

Results, presented in Table 8, confirm our intuition, showing that the model accuracy moves from 0.80 of the full context to 0.66 and 0.64 when the triggering utterance is removed and masked, respectively. The majority of the performance drop is attributable to an increase in false positives, from 2 in the full context to 8 in the corrupted context, as well as a drop in true negatives, which supports our hypothesis. On the other hand, our experiments show that even with insufficiently task-specific few-shot examples, the model can perform significantly better than the random chance baselines (see also 3) through a solid instruction prompt.

Table 8
Proactivity prediction with corrupted dialogue snippets on MultiWoz. Highlighted TN and FP are statistically different from the test with full context (p-value = 0.04123); results with Trigger Utterance both Empty and Masked are statistically lower ($p < 0.01$) than in Full Context setting.

	TESTS			BASELINES		
Trigger Utterance	**Full Context**	**Empty**	**Masked**	**Full Context**	**Empty**	**Masked**
True Positives	17	16	15	20	22	23
True Negatives	23	17	17	5	5	5
False Positives	2	8	8	20	20	20
False Negatives	8	9	10	5	3	2
Accuracy	0.80	0.66	0.64	0.50	0.54	0.56
Precision	0.89	0.67	0.65	0.50	0.52	0.53
Recall	0.68	0.64	0.60	0.80	0.88	0.92
F1 Score	0.77	0.65	0.62	0.62	0.66	0.68
Cohen's Kappa	0.59	0.31	0.26	-0.01	0.07	0.11

6. Conclusions and Future Work

We introduced a new task, namely, last utterance proactivity prediction, aiming at assessing the capacity of Large Language Models to detect and annotate proactive behaviours in task-oriented dialogues. The task allows us to shorten the context from a whole dialogue to a dialogue snippet, simplify the annotation process, and balance the dataset for positive and negative labels. We showed that a few-shot approach with GPT-4o achieves encouraging performance on a test set composed of dialogue snippets collected from five different corpora, and that in particular for the Nespole corpus the agreement between the model labels and the human-annotated gold labels is nearly equivalent to the agreement between humans.

As for future work, there are several ongoing activities. First, we are still investigating techniques to further improve the performance on the task, especially in testing on the combined dialogues from all corpora. Then, we plan to use the GPT-4o model to automatically annotate a large amount (i.e., about 100K) of dialogue snippets, in order to create a training corpus, which, in turn, will be used to instruct an open source model (e.g., Llama 3 8B) to detect proactivity.

References

[1] G. Bonetta, C. D. Hromei, L. Siciliani, M. A. Stranisci, Preface to the Eighth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: Proceedings of the Eighth Workshop on Natural Language for Artificial Intelligence (NL4AI 2024) co-located with 23th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2024), 2024.

[2] V. Balaraman, B. Magnini, Proactive systems and influenceable users: Simulating proactivity in task-oriented dialogues, in: Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue-Full Papers, Virually at Brandeis, Waltham, New Jersey, July. SEMDIAL, 2020.

[3] V. Balaraman, B. Magnini, Pro-active systems and influenceable users: Simulating pro-activity in task-oriented dialogues, Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue (2020).

[4] P.-M. Strauss, W. Minker, Proactive spoken dialogue interaction in multi-party environments, Springer, 2010.

[5] Y. Deng, W. Lei, W. Lam, T.-S. Chua, A survey on proactive dialogue systems: Problems, methods, and prospects, arXiv preprint arXiv:2305.02750 (2023).

[6] S. C. Levinson, Pragmatics, Cambridge University Press, Cambridge, United Kingdom, 1983.

[7] I. Sucameli, A. Lenci, B. Magnini, M. Simi, M. Speranza, Becoming jilda, in: Proceedings of the Seventh Italian Conference on Computational Linguistics CLIC-it 2020, CEUR-WS, Bologna, 2020.

[8] O. Shaikh, K. Gligorić, A. Khetan, M. Gerstgrasser, D. Yang, D. Jurafsky, Grounding gaps in

language model generations, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 6279–6296.
[9] T. Labruna, S. Brenna, A. Zaninello, B. Magnini, Unraveling chatgpt: A critical analysis of ai-generated goal-oriented dialogues and annotations, arXiv preprint arXiv:2305.14556 (2023).
[10] B. Ding, C. Qin, L. Liu, L. Bing, S. Joty, B. Li, Is gpt-3 a good data annotator?, 2022. URL: https://arxiv.org/abs/2212.10450. doi:10.48550/ARXIV.2212.10450.
[11] F. Huang, H. Kwak, J. An, Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech, ArXiv abs/2302.07736 (2023).
[12] F. Hewett, Sequential Organisation in WhatsApp Conversations., Tesi di laurea triennale non pubblicata, Libera Università di Berlino, semestre estivo, 2017.
[13] S. Burger, L. Besacier, P. Coletti, F. Metze, C. Morel, The nespole! voip dialogue database, in: Seventh European Conference on Speech Communication and Technology, 2001.
[14] N. Mana, S. Burger, R. Cattoni, L. Besacier, V. MacLaren, J. McDonough, F. Metze, The nespole! voip multilingual corpora in tourism and medical domains, in: Eighth European Conference on Speech Communication and Technology, 2003.
[15] I. Sucameli, A. Lenci, B. Magnini, M. Speranza, M. Simi, Toward data-driven collaborative dialogue systems: The jilda dataset, Italian Journal of Computational Linguistics (2021).
[16] R. Lowe, N. Pow, I. V. Serban, J. Pineau, The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems, in: Proceedings of the SIGDIAL 2015 Conference, 2015, pp. 285–294.
[17] X. Zang, A. Rastogi, S. Sunkara, R. Gupta, J. Zhang, J. Chen, Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines, arXiv preprint arXiv:2007.12720 (2020).
[18] C. Zheng, H. Zhou, F. Meng, J. Zhou, M. Huang, Large language models are not robust multiple choice selectors, in: The Twelfth International Conference on Learning Representations, 2023.
[19] X. Chen, R. A. Chi, X. Wang, D. Zhou, Premise order matters in reasoning with large language models, arXiv preprint arXiv:2402.08939 (2024).
[20] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics 12 (2024) 157–173.
[21] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, biometrics (1977).
[22] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, Z. Sui, A survey on in-context learning, arXiv preprint arXiv:2301.00234 (2022).
[23] Y. Zhang, Q. Yang, An overview of multi-task learning, National Science Review 5 (2018) 30–43.
[24] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big data 3 (2016) 1–40.
[25] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI global, 2010, pp. 242–264.