Evaluating the Affordances of Large Language Models for Enhancing Rule-Based Chatbots for Task-Based Language Teaching

Daniela Verratti Souto
9cts-Term Paper
5692183

Mareike Schmitt
9cts
6061924

Megan Horikawa
9cts-Term Paper
6196282

{daniela.verratti-souto, megan.horikawa, mareike.schmitt}@student.uni-tuebingen.de

Abstract

This study evaluates the performance of a large language model integrated into a rule-based chatbot designed for task-based language teaching. Two tasks were used to assess the model's ability to detect relevance, grammatical constructs, and factuality in user inputs. We compared the behavior of the model based on outputs generated using the few-shot and chain-of-thought prompting strategies. Chain-of-thought prompting performed slightly better than few-shot prompting, although, in most cases, it showed greater variability across numbers of examples. Qualitative analysis highlights struggles with vague or unexpected inputs, as well as difficulties in detecting specific grammatical structures and identifying untrue information. However, the classification of relevant input yielded promising results. There was some evidence to suggest that binary factuality checks are also viable. Future work includes expanding datasets and exploring the effect of a larger number of few-shot examples, as well as the integration of an LLM into the conversation flow of the task chatbots in the AISLA system.

1 Introduction

The educational sector has experienced rapid transformations, particularly with the integration of digital technologies in classrooms. Traditional paper-based resources are increasingly being replaced by tablets and other digital tools, reflecting a broader trend towards a more technology-driven approach to learning. These changes have accelerated over time, and while they promise enhanced educational experiences, they have often felt overwhelming. The introduction of advanced technologies such as ChatGPT and large language models (LLMs) has only added to this complexity, initially sparking both excitement and skepticism (Xu et al., 2024).

Despite initial uncertainties, LLMs are now recognized for their potential to transform learning environments. They offer learners limitless opportunities for enhancing skills such as reading, writing, and critical thinking across various educational levels. Through tools that generate practice problems, summaries, and language acquisition aids, LLMs can support complex problem-solving processes and promote deeper engagement with the material. Teachers can also leverage these models for personalized instruction, lesson planning, and grading automation, thus improving teaching efficiency and tailoring learning experiences to individual student needs (Kasneci et al., 2023).

In light of these capabilities, we propose the integration of an LLM into rule-based chatbots powered by Amazon Lex v2 in AISLA, an English language learning application designed for schools. The chatbots are task-based, focusing on language teaching. Task developers are required to design these chatbots by manually defining "slots", namely, information that needs to be obtained at each conversation turn in order to complete the task. Additionally, they must provide a list of possible values for each slot and sample student utterances at each point in the conversation. Deviation from the sample utterances or pre-defined slot values often causes breakdowns in communication, forcing the chatbot to re-elicit the slot.

By incorporating an LLM, learner utterances can potentially be evaluated in different aspects. Here, we focus on the ability of the LLM to check whether the elicited grammatical forms are used, whether the user's input is relevant to the task at hand and, when necessary, to compare the truth value of the user's proposition with a ground truth provided by the task instructions. Meanwhile, maintaining the rule-based chatbot at the center of the conversation ensures that interactions follow a predefined, goal-oriented path. This integration aims to enhance the natural flow of interactions, making tasks more intuitive, while also reducing the time needed for chatbot development. Ultimately, this hybrid approach seeks to blend the robustness of rule-based systems with the flexibility and intelligence of LLMs to create an innovative and effective language learning tool.

To justify this integration, it is necessary to evaluate the capabilities of the LLM to successfully judge utterance relevance, factuality and construct presence in the context of the AISLA system, preferably with authentic learner language. Accordingly, this paper sets out to answer the following research questions:

Q1. Can an LLM identify relevant information from learner language for a task-required piece of information ('slot') in the context of a TBLT-based chatbot for English learners?

Q2. Can LLMs recognize whether a grammatical construction is present in a given learner utterance?

Q3. Can LLMs identify whether a learner utterance is factually correct in the context of a TBLT-based task?

Q4. How do different prompting strategies affect the capabilities of LLMs to identify relevance, grammatical constructs and factuality in the context of a TBLT-based conversational agent?

2 Background

2.1 AISLA

AISLA (https://aisla.kibi.group/) (Bear et al., 2024) is a task-based conversational agent for English as a foreign language. AISLA uses the task based language teaching (TBLT) framework, which emphasizes functional language use over traditional form-focused methods of language teaching. An English learner interacts with AISLA by simulating different conversations based around a task, such as making a reservation at a restaurant, or describing a recent trip to a friend.

Amazon Lex v2, a service for building a conversational AI interfaces, powers AISLA's chatbots. This service requires the learner to fill a slot during each turn of the conversation. These are pieces of information that are required to complete the task; for instance, in the context of booking a table at a restaurant, the slots include time and date for the reservation, party size and sitting preferences. Sometimes, a specific grammatical form is expected, while other times a more open answer is accepted. If the input does not match the predefined answers programmed into the chatbot, the conversation cannot continue, and the bot will keep prompting for the slot until an acceptable input is provided or until the maximum number of attempts is reached. This often results in cases where the learner gives a valid but unrecognized answer, causing them to become stuck at that point in the dialogue. This can lead to frustration and may prevent the learner from completing the task.

The use of LLMs in conjunction with the Amazon Lex bots is justified by the fact that it is not realistic to expect task creators to be able to foresee all valid responses for a conversation turn. Furthermore, it is unclear how the Amazon Lex interface compares the predefined answers to the input provided, making it difficult for content creators to reliably choose appropriate responses for each slot. In some instances, Lex has accepted a word with the opposite semantic meaning to the predefined answers as valid input, even though it contradicts the expected answer in the context of the conversation. For example, the student response, "slower" was accepted as a correct response by Lex to the following "Was Peter faster or slower than Andy?" when the task instructions indicate that Peter was in fact faster. Incorporating an LLM, we aim to explore potential solutions to address these challenges within the system.

2.2 Reference Work

The integration of Large Language Models (LLMs) into language teaching and assessment has sparked considerable interest in recent years, with significant advancements being driven by collaborations between industry and academia. One prominent example is the partnership between Duolingo and OpenAI, which introduced the subscription service Duolingo Max. This service offers features such as "Role Play" and "Explain My Answer," enabling interactive language learning experiences where chatbots engage users in conversations that are supplemented by feedback and grammar instruction (Caines et al., 2023).

In addition to OpenAI's GPT, other LLMs like the 'text-to-text Transformer' (T5), PaLM, LaMDA by Google, LLaMA, and Open Pre-trained Transformers (OPT) have been explored for their potential to enhance language education. T5, for instance, has been particularly effective in text-to-text generation tasks, such as question answering and text summarization, proving useful in the creation of reading exercises and prompts for writing and speaking activities (Caines et al., 2023). Similarly, BART has demonstrated state-of-the-art performance in generating parallel sentences for tasks like machine translation and grammatical error correction (Lewis et al., 2019).

LLMs have also been applied to text simplification, which is crucial for beginner language learners. By leveraging LLMs, learners can access texts at various difficulty levels, enabling a more tailored and accessible learning experience. Another innovative use of LLMs is in human-in-the-loop content generation, where models like GPT-3 generate texts based on specific prompts, and human evaluators curate these outputs to ensure quality. This iterative process allows the model to learn from feedback, refining its output over time (Caines et al., 2023). This approach builds on earlier work by Fan et al. (2018), who explored hierarchical neural story generation, emphasizing the importance of human oversight in content generation.

The application of neural models for language assessment has also evolved, with increasing reliance on LLMs due to their capacity to handle complex linguistic tasks. However, studies made by Caines et al. (2023) suggest that while LLMs excel in many areas, their integration with traditional feature-based models yields the most accurate assessments. For instance, combining BERT-style models (Devlin et al., 2019) with feature-based approaches has resulted in superior performance, particularly in educational settings where the stakes of accurate assessment are high. Furthermore, LLM-enhanced models have been shown to outperform traditional models on benchmark datasets and novel evaluation metrics, despite the higher computational and environmental costs (Caines et al., 2023).

In the realm of task-oriented dialogue (ToD) systems, pre-trained language models have been widely adopted. Pioneering studies by Zhang et al. (2020) and Peng et al. (2020) laid the groundwork by integrating generative models into dialogue systems, which were further enhanced through techniques like contrastive state training and belief state differences. Recent advancements have seen the introduction of instruction-tuned LLMs, such as those described by Mok et al. (2024), who proposed a novel framework for argument filling in API calls within ToD systems. This framework utilizes a two-phase instruction-tuning process, combining supervised fine-tuning with rejection sampling to improve model robustness and accuracy. These methods have demonstrated significant potential in generating structured and contextually appropriate responses, particularly when using models like LLAMA-7B and ChatGPT, evaluated on datasets such as STAR and SGD (Mok et al., 2024).

The continued development of LLMs, coupled with advances in instruction-tuning and prompt design, underscores the growing potential of these models to revolutionize educational technology and task-oriented dialogue systems. As these technologies mature, their integration into language learning and assessment frameworks is likely to become increasingly sophisticated, offering more personalized and effective learning experiences.

2.3 Prompting LLMs

LLMs have demonstrated remarkable capabilities across a wide range of natural language processing tasks (Hengle et al., 2024). For instance, the emerging capability of in-context learning (ICL), first shown by Brown et al. (2020), has introduced the concept of Zero- and Few(k)-Shot-Prompting. Zero-shot prompting involves providing an LLM, like GPT-3, with only a natural language instruction without examples. This method tests the model's ability to generalize across tasks based on its pretraining. Zero-shot prompting is effective in simple tasks like language modelling, basic translations, and factual question-answering, where it can deliver competitive results without specific task examples. However, the model's performance is often limited in more complex tasks that require task-specific context or nuanced reasoning.

Few-shot prompting (FS), also known as k-shot prompting, involves presenting the model with a small number (k) of task-specific examples alongside the instruction. This method boosts the model's performance on a wide range of NLP tasks, such as common-sense reasoning, translation, and reading comprehension. Few-shot prompting bridges the gap between zero-shot and fully fine-tuned models, enabling LLMs to achieve near state-of-the-art performance in various benchmarks, even without gradient updates. The effectiveness of few-shot learning improves significantly with the size of the model, as larger models like GPT-3 demonstrate a steeper improvement curve, especially in reasoning and completion tasks (Brown et al., 2020). However, tasks that require complex reasoning, such as solving multi-step mathematical problems or understanding symbolic relations, have posed significant challenges for these models when using standard prompting techniques.

Chain-of-Thought (CoT) prompting is an approach that enhances the reasoning capabilities of large language models by encouraging them to generate a series of intermediate reasoning steps, akin to how humans break down complex problems. Instead of merely predicting an output, the model processes the task step-by-step, offering a more structured and interpretable pathway to solving complex tasks. For example, when solving mathematical problems, models like PaLM 540B, prompted with CoT reasoning, outperformed standard few-shot prompting approaches, achieving new state-of-the-art results on benchmarks such as GSM8K (Cobbe et al., 2021). In fact, PaLM 540B surpassed fine-tuned models like GPT-3, which had previously been among the top-performing models in similar tasks (Wei et al., 2023).

A key benefit of CoT prompting is its interpretability. By producing a chain of intermediate steps, the model provides insight into how it arrives at its conclusions. This allows for greater transparency in the model's reasoning process, facilitating easier debugging and identification of errors in reasoning, especially when compared to black-box outputs produced by standard prompting methods (Wei et al., 2023).

3 Data

Data was collected from a pilot study of AISLA conducted in three 7th grade gymnasium-level English classes (Bear et al., 2024). A total of 305 student inputs, randomly sampled from conversations with two task chatbots, were manually annotated. Completion of each task generally involved between 5 and 8 turns.

The first task, "Comparing Athletes," is considered a "closed" task, requiring specific ground truth values in the responses. In more general terms, the instructions detailed the outcomes of several sporting events where two athletes, Peter and Andy, had competed. This included information about their performance in absolute terms. For instance, the instructions list how many seconds each of them took to run 100 meters, or how far each of them jumped in a long jump competition. The task then consisted of relaying this information to a friend who had missed the events by answering questions using comparatives. The truthfulness ("factuality") of the learner utterance can thus be checked against the ground truth provided by the instructions.

On the other hand, the second task, "Buying a Birthday Gift," was more open-ended, allowing students greater freedom in their responses at most stages of the conversation. In this task the students were instructed to buy a birthday present for their friend Max, along with a short description of his tastes (e.g., music, video games, movies). The interaction follows what is to be expected from a real-life conversation in a department store, with questions like "What can I help you with?" or "What does your friend like?". The students were able to decide on a present from a limited set of options and to state the means of payment.

3.1 Annotation

Each student input was annotated for human judgement according to three criteria: relevance to the question being asked, presence of the grammatical construct (if any was elicited), and factuality according to the task instructions (when a ground truth existed). Construct presence and relevance were annotated with a set of binary labels (true/-false), whereas factuality used a ternary classification (true/false/None). The "None" label was introduced for such cases where the student utterance lacked a truth value and therefore could not be said to be true or false in any given context.

Relevance annotation posed the question of what exactly should be counted as "relevant" within this specific LLM application. Candidate definitions included utterances that were plausible given the conversation history and utterances that did not deviate from the task at hand. However, given the ultimate intention to integrate the LLM into the application during production, and taking into account that Lex requires the slots be filled with a value, we decided that a given user utterance is relevant, according to our definition, if and only if it is a plausible utterance given the context, and it provides the necessary information for a human interlocutor to create a mental representation of the facts such that the initial question is fully answered. Should a learner input not fulfil the criteria for relevance, it would be desirable for the chatbot to re-elicit an answer until it can satisfy the imaginary interlocutor's curiosity.

To obtain the current baseline for relevance judgements, we annotated each learner utterance for "Lex-relevance". This was based on whether the input was accepted by Amazon Lex v2's rule-based chatbot at the time of the pilot study, allowing the conversation to advance to the next step. Using this annotation scheme, Lex-relevance was directly comparable to "human-relevance", making the latter the ground truth by which to evaluate both the Lex behavior and the LLM judgements during the experiment.

In terms of the target construct, different grammatical features were expected depending on the task. For example, for "Comparing Athletes," students were supposed to use comparative adverbs at each turn, while for "Buying a Birthday Gift," only two turns expected a specific grammatical structure, namely, comparative adjectives and present progressive tense. Each input with an elicited construct was checked for the presence of the corresponding grammatical structure, and its presence or absence was annotated accordingly.

Additionally, the inputs were evaluated for their alignment with ground truth where applicable. For more closed-ended tasks like "Comparing Athletes," this involved checking whether the student's response correctly reflected the expected factual information (e.g., identifying the faster athlete). In contrast, for the open-ended "Buying a Birthday Gift" task, the emphasis was more on whether the input followed a logical and relevant flow in the conversation rather than strictly factual accuracy.

Take the following conversation extract as an example:

> **Bot**: Did Peter jump as far as Andy?
> **User**: *yes he jumped as far as Andy*
> **Bot**: Wow! Who did better in the shot put?
> **User**: *what ist the Shot Put?*
> **Bot**: Peter threw ...?

The user utterance *yes he jumped as far as Andy* contains a comparative adverb, is relevant yet not factual, since Andy won the long jump according to the instructions. As the bot moves on to elicit the next slot, it is clear that Lex accepted the input. On the other hand, *what ist the Shot put* does not contain the construct, is not relevant as per our definition above and, being a question, has no truth value. Finally, given that the chatbot re-elicits the response, we know that the utterance was rejected by Lex.

Finally, all separate dialogues were given a unique identification number to make sure that messages belonging to the same conversation could be identified as such. This annotation process ensured a comprehensive evaluation of both system and human perspectives on language use, aiming to facilitate the development of more robust conversational AI models.

4 Methods

4.1 The model

For this project, the *Mixtral-8x7B-Instruct-v0.1* model by Mistral (Mistral AI, 2023) was selected. Built upon an advanced transformer architecture, the *Mixtral-8x7B-Instruct-v0.1* model is specifically fine-tuned to excel at instruction-following tasks and has been optimized for high-quality, task-oriented interactions, which makes it suitable for scenarios requiring precise control over the format and content of its responses.

It is designed to operate effectively in environments where generating structured, reliable, and accurate outputs is important. Among its strengths are its capabilities in generating structured data formats, which are essential for integration into automated pipelines or systems that depend on precise data formats. The model has been trained on an extensive corpus of text, ensuring its ability to handle both general-purpose queries and domain-specific tasks with a high degree of reliability and coherence (Mistral AI, 2023).

The decision to adopt the *Mixtral-8x7B-Instruct-v0.1* model was largely influenced by its open-access nature, which provides the freedom to utilize, modify, and fine-tune the model without the restrictions imposed by proprietary systems. The availability of open weights allows for integration into AISLA's framework and enables further optimization to meet specific application requirements. The choice of *Mixtral-8x7B-Instruct-v0.1* above other open-weight models was informed by our own preliminary trials using the Groq playground API (`https://console.groq.com/playground`), which revealed this model to be the one to most reliably output valid JSON using the API's JSON mode.

Calls to the model were made using the Groq Playground API, for which free keys can be obtained. Given the need for accurate, reproducible predictions above creative and varied text output, the temperature value used was zero. Finally, the calls were made using the API's JSON mode, which enforces the validity of the output according to the standard JSON format, necessary to facilitate data extraction, post-processing, and analysis.

4.2 Prompting the model

The first step to obtaining the output was designing a prompt template with the necessary information and a dynamic prompt builder that took into account the task and its instructions, the message along with its previous conversation history, the construct being elicited at the present turn in conversation (if any), the prompting strategy and the number of examples to be passed into the model at each call. The prompt template used can be found in Appendix A

The user messages for the few-shot examples and their corresponding conversation histories were taken from the unused chat logs of the same task and annotated for the three criteria.

In the case of the k-shot prompting strategy without CoT, the prompt asked the model to output a JSON-formatted string containing a Boolean value for relevance, a Boolean value for the presence of the elicited construct if one existed[1] and, in the case of the Comparing Athletes task, a ternary factuality judgement (True/False/None) based on the provided task instructions. Then, k sample JSON outputs with the correct labels were provided. Values of k ranged from 0 to 5. In some cases, the API would fail to validate the JSON output by the model, such that it was necessary to repeat the call with minor adjustments to the prompt.

For k-shot with CoT prompting, the prompt included a request to output a key-value pair with a "thought" before each judgement, which was also demonstrated in the sample JSON outputs. Once again, the number of examples for the CoT prompting strategy were between 0 and 5. Appendix A.3 shows an instance of a CoT output provided as an example to the model.

Spelling and grammatical mistakes in the students' utterances were not corrected, and the LLM was instructed to interpret the intent behind the users' responses, allowing for natural language errors common in learning contexts.

4.3 Data analysis

The JSON-formatted strings were parsed using standard JSON libraries. However, with chain-of-thought prompting, the model's output often failed to be parsed. In these instances, the model was reprompted until a valid, parseable JSON string was produced. The classification labels generated by the model were then added to a dataframe, where each row represented an individual user message, and the columns corresponded to the various experimental conditions: the task, the prompting strategy, and the number of examples in the prompt that led to the output.

Some data-cleaning and post-processing steps were necessary before carrying out the data analysis.[2] On inspection, we realized that the model did not always produce solely Boolean values and Nones, but rather there existed some JSON objects with values such as "partially true" or "not defined". These were mapped to the closest Boolean (or, in the case of factuality, Boolean or None) value. Additionally, in certain cases, the CoT prompting strategies led to the "construct presence" property being assigned a value of None, even when a specific construct was elicited at the conversation turn. Further inspection of the reasoning chains revealed that this happened only when the grammatical structure was missing in the user's input. We corrected this error by mapping those instances to False.

For the subsequent analysis, we distinguished between binary and multi-class criteria. The former included the relevance and construct presence properties, whereas factuality was the only instance of proper multi-class classification.

In the case of the binary criteria, we computed standard classification metrics: accuracy, precision, recall and F1-score. True positives and true negatives were calculated in the usual manner. On the other hand, multi-class precision, recall and F1 were computed for factuality. This meant that for each label —True, False and None—, precision, recall and F1-score were calculated. After observing the results for this property, we explored the possibility of a binary factuality criterion by unifying the False and None labels that were output by the model. Finally, as a means of comparison, we also calculated the binary metrics for the system's relevance judgement without LLM integration.

5 Results

The following section presents both results obtained from the quantitative analysis of the model-generated responses (Section 5.1), and qualitative

[1] In order to make the prompt-building simpler, we asked the model to output None if no target construct was specified. However, for our purposes we only needed to analyze Boolean values from those conversation turns where a target construct is expected.

[2] All scripts used for inference and analysis are provided in the following repo: `https://github.com/meghorikawa/ULLM/`. The conversation data has been kept private for the purpose of complying with ethical standards.

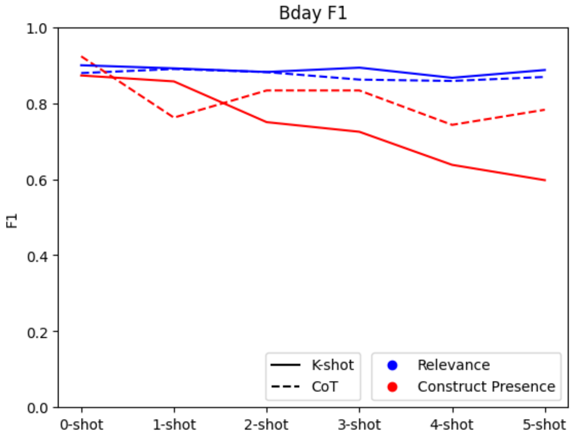

Figure 1: F1 scores for predictions of relevance and construct presence for the birthday present task

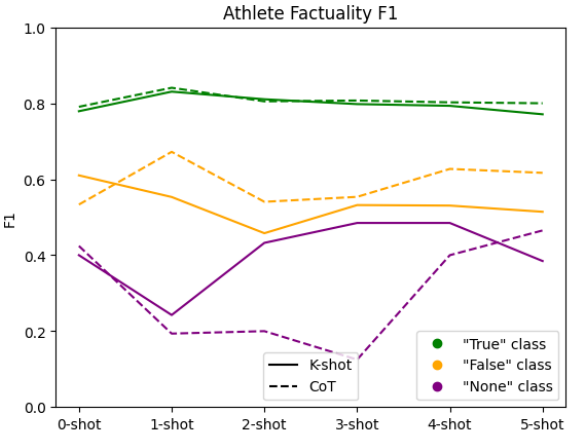

Figure 3: F1-scores for multi-class predictions of factuality for the Comparing Athletes task.

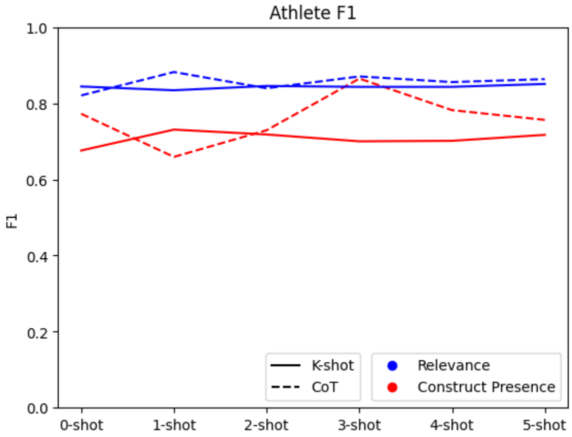

Figure 2: F1 scores for predictions of relevance and construct presence for the comparing athletes task

observations of the model's behavior in the classification of the user inputs (Section 5.2).

5.1 Quantitative analysis

5.1.1 Classifying relevant input

As shown in Table 1 and Table 2, the LLM presented an over-all improved performance in classifying relevant student input over the Lex system. Although, in all cases, the Amazon Lex showed higher precision scores, the LLM substantially outperformed it in all other metrics.

With regard to the influence of the two different prompting strategies on the classification, Figures 1 and 2 illustrate that both the number of examples and the prompting strategy had little effect on the model's performance. All in all, the LLM was more successful in classifying relevant input than in the remaining two criteria.

5.1.2 Grammatical construct detection

Figures 1 and 2 also display the F1-score of the model's answers for the construct presence criterion. Judging by these scores, the model achieves moderate yet inconsistent success when determining whether an utterance contains comparative adverbs (in the case of the Comparing Athletes task) or comparative adjectives and the present progressive tense (in the Birthday Present task).

For the "Buying a Birthday Present" task, the F1-score of the K-shot strategy without CoT shows a steady decline as the number of examples increases. The results generated with the CoT strategy presents a similar trajectory, but the decrease is not so consistent. On the other hand, the scores for the "Comparing Athletes" task using the pure K-shot strategy remain relatively constant across the values of *k*, while CoT is once again unstable, peaking at 3-shot – ultimately, the 5-shot CoT prompt for this task performed slightly worse than the 0-shot

Such inconsistent results warrant a closer look into the model's performance for this criterion. Tables 3 and 4 provide an overview of the precision and recall achieved by the model for both tasks. Noticeably, CoT performs better than pure K-shot prompting in all cases, except for the recall score corresponding to the "Comparing Athletes" task, where there is a substantial difference in favor of the few-shot strategy without chains of thought.

Generally speaking, there was no clear overarching trend with regard to the increase or decrease in performance as a function of the number of examples and prompting strategy.

	Lex	**CoT (Mean ± SD)**	**K-shot (Mean ± SD)**
Accuracy	0.624	0.820 ± 0.029	0.776 ± 0.009
Precision	0.867	0.852 ± 0.034	0.739 ± 0.008
Recall	0.411	0.860 ± 0.029	0.982 ± 0.009
F1-Score	0.557	0.855 ± 0.022	0.843 ± 0.005

Table 1: Relevance judgement metrics for the Comparing Athletes task. Lex performance compared to CoT and K-shot. For CoT and K-shot, values are presented as the mean of the metric across *k* values ± Standard Deviation.

	Lex	**CoT (Mean ± SD)**	**K-shot (Mean ± SD)**
Accuracy	0.611	0.792 ± 0.016	0.810 ± 0.016
Precision	0.986	0.892 ± 0.010	0.886 ± 0.017
Recall	0.544	0.856 ± 0.031	0.888 ± 0.030
F1-Score	0.701	0.873 ± 0.012	0.887 ± 0.011

Table 2: Relevance judgement metrics for the Birthday Present task. Lex performance compared to CoT and K-shot. For CoT and K-shot, values are presented as the mean of the metric for each *k* value ± Standard Deviation.

5.1.3 Classification of input factuality

Factuality of the input was only taken into account in the athlete comparison conversations, since an objective truth was present in the instructions of this specific task. Figure 3 shows the F1-scores obtained in the multi-class classification of user input into the classes “True”, “False” and “None”. Visualizing this metric, it is transparent that the model had little difficulty in determining when a user sentence contained true information, whereas distinguishing utterances with untrue information and, especially, those without a truth value proved to have increased complexity. In terms of the relationship between the number of examples given to the model and the resulting F1-score, the value of *k* appears to have little effect on the performance for the True class, which displays relatively stable behavior. On the other hand, the False and None classes do not show a clear positive or negative trend.

Zooming in on the precision and recall scores (Tables 5 and 6, respectively) reveals that both metrics present similarly low values for the False and None classes, suggesting that the issue is not that the model constantly misidentifies the same class, but rather that it is inconsistent in finding both classes.

Given the clear distinction in performance across classes, we thought it sensible to explore the possibility of turning factuality into another binary property. We did so by merging the None and False classes, such that all None labels, gold-standard and predicted, were changed to False. Unsurprisingly, these results proved much more promising and consistent across numbers of examples, as summarized in the F1-score graph in Figure 4.

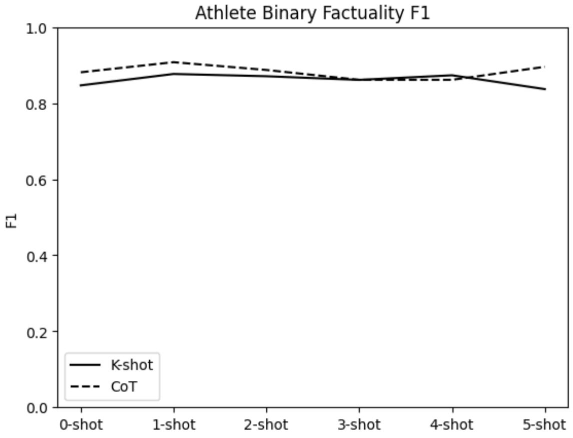

Figure 4: F1-scores for the factuality property in the Comparing Athletes task after merging the False and None classes.

5.2 Qualitative analysis of the model ouput

In the following paragraphs, we describe our qualitative observations of the model output to provide an account of the patterns that emerged as a result of specific kinds of input and how they affected the performance of the LLM.

In the birthday task, the model tended to make mistakes in the relevance judgement when user inputs are unrelated to the task or conversation history. For example, inputs like “The colour green” or “Do you have pets” are problematic — the first is misleading in relation to the task, and the second being completely irrelevant - resulting in the model oscillating between classifying them as true

	Athletes		Birthday	
	CoT	**Few-shot**	**CoT**	**Few-shot**
0-shot	0.907	0.563	0.960	0.857
1-shot	0.967	0.582	0.667	0.828
2-shot	0.921	0.571	0.758	0.649
3-shot	0.906	0.549	0.758	0.595
4-shot	0.827	0.563	0.605	0.524
5-shot	0.738	0.564	0.643	0.460
mean	0.877	0.565	0.732	0.652

Table 3: Precision scores for the construct-presence criterion in the athlete comparison and birthday present tasks across values of *k*.

	Athletes		Birthday	
	CoT	**Few-shot**	**CoT**	**Few-shot**
0-shot	0.672	0.845	0.889	0.889
1-shot	0.500	0.983	0.889	0.889
2-shot	0.603	0.966	0.926	0.889
3-shot	0.828	0.966	0.926	0.926
4-shot	0.741	0.931	0.963	0.815
5-shot	0.776	0.983	1.000	0.852
mean	0.687	0.945	0.932	0.877

Table 4: Recall scores for the construct presence criterion in the Athlete and Birthday tasks across values of *k*.

	True		False		None	
	CoT	**K-shot**	**CoT**	**K-shot**	**CoT**	**K-shot**
0-shot	0.773	0.761	0.560	0.571	0.412	1.000
1-shot	0.904	0.768	0.612	0.667	0.200	0.235
2-shot	0.853	0.726	0.536	0.679	0.167	0.381
3-shot	0.817	0.737	0.544	0.641	0.125	0.471
4-shot	0.833	0.747	0.587	0.605	0.429	0.471
5-shot	0.842	0.726	0.635	0.540	0.370	0.500

Table 5: Multi-class precision scores for the factuality criterion across CoT and K-shot for varying values of *k*.

and false with no clear pattern with regard to the number of examples. Further, it also struggled with short, vague or open-ended responses, such as “Nothing,” “Birthday Gift,” or “No,” as well as inputs that deviated from expected content, like the reply to a list of suggestions: “I don’t like those things.” Even though this is a valid answer to the question, this kind of uncooperative response seems to confuse the model, leading to a similar unstable behavior as mentioned before.

Additionally, relevance judgements were affected by unclear or grammatically incorrect language, which also appeared to cause difficulties in making accurate judgements. An example of this is nonsense input, such as “Yes I would have to do it for you to do it for me to do with the fact that I am not sure if you are interested in the position of the position and the position5 of the company.” The model found this input difficult to process, leading it to accept and also decline the input in the relevance category and also for the presence of the construct. These types of inputs cause issues across both FS and CoT. Moreover, with CoT, the model struggles with responses that omit key specifics, such as the fact that the gift referenced in the task is for a friend or for a birthday. For instance, inputs such as “I need a present” often result in errors in the relevance judgement, deeming it not a relevant response, when using the CoT prompting strategy.

The model also struggled with correctly recognizing the absence of the target construct “present progressive tense” when the user’s input was either too short (e.g., “Birthday gift” instead of “I’m looking for a birthday gift”) or when the structure was completely omitted or replaced by a different construct, as in “I would buy a video game console.” In both instances, the model sometimes incorrectly marked the construct as present, even when it was absent.

Similarly, the model faced challenges with the second target construct, “comparative adjectives,” particularly with inputs like “It’s too expensive” or the misspelled version “It’s to expensive.” While using FS, the model oscillated between detecting a comparative and deeming it absent. However, with CoT, it consistently identifies the construct as being present, with a limited number of exceptions. Nevertheless, the model handled other minor misspellings well, since they did not seem to significantly impact performance.

In the athlete comparison task, the model faced difficulties when dealing with inputs that were either too short or ambiguous. For instance, when the input is just a name like “Peter” in response

	True		False		None	
	CoT	**K-shot**	**CoT**	**K-shot**	**CoT**	**K-shot**
0-shot	0.810	0.798	0.509	0.655	0.438	0.250
1-shot	0.786	0.905	0.745	0.473	0.188	0.250
2-shot	0.762	0.917	0.545	0.345	0.250	0.500
3-shot	0.798	0.869	0.564	0.455	0.125	0.500
4-shot	0.774	0.845	0.673	0.473	0.375	0.500
5-shot	0.762	0.821	0.600	0.491	0.625	0.313

Table 6: Multi-class recall scores for the factuality criterion across CoT and K-shot for varying values of *k*.

to a comparative question (e.g., "Who ran faster, Peter or Andy?"), the model struggled to identify the absence of a comparative. In FS, the model tended to classify such responses as containing the target construct, while CoT often marks them as "None," mistaking it for a conversation turn without an elicited grammar structure. As previously stated, such behavior needed to be corrected during the data analysis.

Moreover, the model showed confusion with sentences like "Andy was better," where a comparative adjective was present with the same form as a comparative adverb (note that *better* can also be an adverb, as in the sentence "Andy ran *better*"). The LLM oscillated between identifying the presence and absence of the target construct "comparative adverbs". This inconsistency suggests that the model had difficulties with subtle grammatical distinctions.

One-word responses also posed a challenge for CoT prompting, as it sometimes failed to classify words like "Faster" correctly, despite being a comparative adverb. FS, however, showed better generalization in handling spelling errors, while COT was more prone to confusion when faced with misspellings or ungrammaticalities (e.g., "Peter jump longer"). Additionally, when asked comparative questions (e.g., "Did Peter throw farther?"), vague responses like "Peter did it" lead to confusion. In some cases, CoT identified the construct as present, but both FS and COT struggled to classify the response correctly.

In the case of relevance, the model often misinterpreted sentences like "Peter ran faster than Peter" or "Andy jumped as far as Andy," treating them as relevant despite the logical error, which causes it not to fulfil our criteria for relevance. CoT eventually learned to classify such inputs as false, but only after multiple examples.

With regard to factuality, the model occasionally demonstrates flawed reasoning. For instance, it mistakenly interpreted a slower time as faster (e.g., 13.5 seconds as being faster than 13 seconds), reflecting a misjudgment of numerical values. Similarly, the absence of measurement units or grammatically incorrect statements can lead to factual errors, like the model sometimes not accepting these as true even though they might match the ground truth provided in the task instructions.

6 Discussion

In this term paper, we focused on evaluating the feasibility of integrating LLMs into rule-based chatbots for TBLT. The goal is to enhance the interaction capabilities of the chatbot used in the AISLA system, an English learning application, by addressing its limitations in handling student responses that deviate to predefined formats. By incorporating LLMs, the study aims to evaluate whether LLMs can recognize relevant information in user responses, detect specific grammatical structures, and assess the factual accuracy of learner inputs. Furthermore, the study explores various prompting strategies to determine how well LLMs perform in these tasks. The chatbot's effectiveness is evaluated through two specified tasks, which are also used in the AISLA system. The ultimate objective is to blend the robust, rule-based structure with the flexibility of LLMs, improving chatbot interactions for language learners.

In this section, we will address the research questions based on the results obtained in our experiments, as described in Section 5. Additionally, we outline the limitations of the current project and suggest further directions to improve and expand on our findings. We also discuss how the insights gained from the present work can be applied to enhance chatbot interactions in the AISLA app and guide the implementation of new features.

6.1 RQ1: Relevance

The first research question concerns the LLM's ability to determine whether a learner's input is relevant in the context of a TBLT-based conversation task. For this purpose, we defined "relevance" as the property of providing an answer that is both within the task's context and clearly provides the necessary information to answer the chatbot's question.

As shown above, the model consistently performed well across varying numbers of examples and outperformed the baseline provided by Amazon Lex v2 in nearly all metrics. Although Lex is less prone to generating false positives for relevance compared to the LLM—resulting in slightly better precision—it has a much higher false negative rate. This results in the bot repeatedly eliciting the same information, disrupting the flow of conversation and potentially causing user frustration or confusion. This high precision and low recall of Lex is to be expected, given that the system accepts utterances based on a predefined set of sample answers provided at the time of creation and rejects those utterances that do not match according to its algorithm. In contrast, the LLM's relevance judgements aligned with the humans' roughly 80% of the time, with an acceptable degree of precision and recall. These results suggest that an LLM can effectively classify user input as relevant or irrelevant in the context of a TBLT language learning conversation task.

6.2 RQ2: Construct presence

The second research question aimed to evaluate the LLMs ability to correctly identify sentences containing a pre-specified target construct. The results indicate limited success in detecting the presence and absence of the grammatical structures elicited in the two tasks: the present progressive tense, comparative adjectives and comparative adverbs.

It is worth noting the fact that the prompt to the LM included instructions to output "None" when, according to the annotations, the conversation turn was not intended to elicit specific structure. This led to unexpected and incorrect behavior: conversation turns where a construct was elicited but absent from the learner's input were mistakenly marked as "None" instead of "False". This mistake was corrected in the script during the data analysis process. As a result, we cannot fully assert that the values after the post-processing reflect what would have been obtained had the model only been prompted for construct presence in the pertinent conversation turns. For future research or production endeavors using an LLM for the purpose of target grammatical construct detection, we recommend tailoring the prompt to each row in the dataset so that the construct presence property only appears when required.

Overall, the model achieved moderate success when classifying learner inputs for construct presence. Consequently, caution is advised when using Mixtral-8x7B-Instruct-v0.1 for this purpose and to potentially explore tools for extracting grammatical constructs from learner language.

6.3 RQ3: Factual correctness

Factual correctness was the only property in our dataset where we required a multi-class output. A learner utterance is considered factual (*True*) when it matches the state of affairs defined by the task instructions, and non-factual (*False*) when it contradicts them. Moreover, some utterances might not represent a proposition, meaning they are neither true nor false, for which the *None* class was introduced.

The model demonstrated high precision and recall for the *True* class, while they were moderate and low for the *False* and *None* classes, respectively. This, along with the issue related to the wrongly-placed "None" values mentioned in Section 6.2, suggests that the model has trouble telling apart the meaning of False from that of None in the context of classification tasks, while these two are more clearly distinguishable from *True*. This likely occurs because both terms in question are used in similar negative contexts, causing their embeddings to be closer to each other in vector space.

In view of the above, we conducted a *post-hoc* exploration of the viability of using the LM to assess truthfulness as a binary category. Hence, the previous *False* and *None* classes were merged into a single class, *False* and the results improved considerably. As before, it is not possible to ascertain that this would have been the outcome had the model been prompted to do this from the start. However, we believe this a reasonable approximation.

Overall, the LM only provided reliable results for factuality judgements once the distinction between untrue claims and non-statements was eliminated. While a binary conception of factuality in the context of an LLM-enhanced TBLT conversational agent might still be worth pursuing, losing the *None* class has some implications for the potential use of this property in the implementation of new features. Further experimentation with larger models to preserve this distinction is advisable.

6.4 RQ4: The effect of prompting strategies

In our work, we evaluated the performance of pure K-shot prompting and of CoT prompting, across a range of zero to five examples. Interestingly, for the properties for which the best performance — namely, relevance and binary factuality, and the *True* class in multi-class factuality—, the corresponding F1-scores for both prompting strategies did not considerably diverge. This could mean that when the model reaches a certain level of certainty in the classification, a ceiling effect may be observed, such that more powerful prompting strategies like CoT offer no significant improvements beyond simpler strategies like few-shot prompting. Similarly, higher numbers of examples did not significantly enhance nor impair model performance in these cases, showing relatively stable behavior across all values of k. Finally, for these specific properties, CoT demonstrated a slight advantage over pure k-shot for a higher number of examples. However, significance testing would be necessary to draw strong conclusions from this observation.

In contrast, the less successful experiments displayed unpredictable behavior, with the top-performing strategy varying based on the number of examples. Despite this, CoT consistently yielded better overall results when provided with all five examples. Surprisingly, there was no straightforward relation between the number of examples and performance in terms of F1-scores. For construct presence in the Comparing Athletes task, a negative trend was observed for the pure few-shot prompts, whereas in the other cases, the scores fluctuate without a clear pattern.

Overall, there is some evidence, albeit weak, that CoT performs better than few-shot prompting for classifying the properties discussed in this paper, particularly when more examples are provided. A larger range of examples and a more extensive dataset would prove beneficial for corroborating these findings.

6.5 Limitations and outlook

This work has a number of limitations to be acknowledged. Firstly, the dataset is limited to a number of interactions, covering only two tasks. As of now, around one hundred tasks have been designed for the AISLA app, so more extensive testing with a wider variety of tasks is recommended.

Secondly, we only used one model for inference, limiting the generalizability of the insights obtained in this project. Furthermore, we only analyzed the final output of the model and did not look at any intermediate calculations. For example, examining the log-probabilities assigned by the LM for each of the candidate classes would have eliminated the issues with outputs falling outside of the binary or ternary choice for each property. This would have eliminated the need for post-processing that required us to manually assign the closest Boolean or None value to the model's output of some rows.

Moreover, the numbers of examples used spanned from zero to five, which is a limited range in comparison to previous works that have contained k values of up to 32 (Min et al., 2022), 64 (Reynolds and McDonell, 2021) and even 256 (Webson and Pavlick, 2022) in their analyses. A broader range of examples might have revealed clearer trends in the LLM's performance across values of k for K-shot prompting with and without CoT.

In addition, for K-shot and CoT, the same five examples in the same order were used when building the prompts. Since the order was not randomized, it cannot be discarded that a specific example or the order of them might have nudged the model towards the correct class or away from it.

Future work could expand our findings by creating and analyzing a more extensive dataset covering a wider variety of tasks and using a broader range of k-shot examples. Further, comparing the performance of differently-sized models for this task would allow for greater generalizability and shed light on how LLMs' abilities to detect the different properties emerge or stagnate with larger numbers of model parameters. Furthermore, slight modifications, such as adding a classification head to the LLM, might result in higher overall scores.

Additionally, while this study prompted for all properties at the same time, it would be of interest to explore the model's behavior when separating the prompts for each property into different LM calls and compare it to the results obtained here.

Finally, the ultimate goal of this study was to explore the feasibility of integrating an LLM into the current architecture of the AISLA app. The three properties studied here —factuality, relevance and construct presence— could be used to inform both the conversation flow and for providing the teacher with suggestions for grading the task submission. In the first place, LLM relevance judgements might be used to enhance the naturalness of the chatbot interactions. They could, for instance, determine whether the Amazon Lex v2 has received the necessary information to move on in the conversation, or whether the slot should be re-elicited. The construct presence criterion would serve the purpose of informing the teacher whether the learner is actively practicing the grammatical structures that the task targets. Finally, the factuality property can serve as an indicator of how well the student has understood the task instructions. In summary, a higher number of relevant and factual responses containing the constructs would indicate that the student is successfully engaging with the task. This has the advantage of providing the teacher with an overview of the learner's interaction with the app, grounded on explainable and observable features reflecting their performance.

The three-way distinction of the factuality property could be helpful in distinguishing two kinds of user answers. First, cases where the student is simply wrong about the objective truth provided, in which case the message gets the *False* label. Second, those situations where a student might ask the conversational agent a question in a good-faith attempt to better understand the context or the vocabulary in the task; this would ideally be labeled with *None*. Furthermore, if combined with judgements of relatedness to the task, the latter messages could be distinguished from deliberate attempts to derail the conversation.

In summary, this term paper provides a data-informed overview of the capabilities of an LLM to process learner input in the context of a TBLT-based conversation system for English as a Foreign Language. We used Mixtral-8x7B-Instruct-v0.1 to classify learner utterances in terms of their relevance to the task, the presence of a target construct and their truthfulness in relation to the task instructions. We report promising results for the relevance criterion using both few-shot and CoT prompting, as well as moderate success for identifying utterances with the presence of pre-specified grammatical structures with the CoT prompting strategy. Furthermore, the model was able to distinguish truthful learner utterances from untruthful ones, yet it failed to reliably discriminate between false statements and utterances without a truth value. Finally, we discussed the implications of these results, recommended directions for future work and proposed how the integration of an LLM into the AISLA system could inform the implementation of a formative assessment feature for the app.

Acknowledgments

We would like to thank Polina Tsvilodub for her guidance in helping us organize our project and its methodology.

We would also like to express our sincere gratitude to Prof. Xiaobin Chen, Elizabeth Bear, and other members of the KiBi (Künstliche Intelligenz in der Bildung) group at the Hector-Institut für Empirische Bildungsforschung for allowing us to use the AISLA pilot data for our analysis.

Additionally, we acknowledge the use of the Gemini assistant integrated into Google Colab to assist in debugging our code, as well as automated spell checkers in helping fix any spelling and grammar mistakes in our writing. We also acknowledge the use of OpenAI's ChatGPT as a writing assistant for the purposes of rephrasing our own ideas where necessary to better conform to an appropriate academic register.

References

Elizabeth Bear, Xiaobin Chen, Daniela Verratti Souto, Luisa Ribeiro-Flucht, Björn Rudzewitz, and Detmar Meurers. 2024. Designing a task-based conversational agent for EFL in German schools: Student needs, actions, and perceptions. *System*, 126:103460.

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.

Andrew Caines, Luca Benedetto, Shiva Taslimipoor, Christopher Davis, Yuan Gao, Oeistein Andersen, Zheng Yuan, Mark Elliott, Russell Moore, Christopher Bryant, Marek Rei, Helen Yannakoudakis, Andrew Mullooly, Diane Nicholls, and Paula Buttery. 2023. On the application of Large Language Models for language teaching and assessment technology. *Preprint*, arXiv:2307.08393.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias

Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Amey Hengle, Prasoon Bajpai, Soham Dan, and Tanmoy Chakraborty. 2024. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models. *Preprint*, arXiv:2408.10151.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mistral AI. 2023. Mixtral-8x7b-instruct-v0.1. Accessed: 2024-09-12.

Jisoo Mok, Mohammad Kachuee, Shuyang Dai, Shayan Ray, Tara Taghavi, and Sungroh Yoon. 2024. LLM-based Frameworks for API Argument Filling in Task-Oriented Conversational Systems. *Preprint*, arXiv:2407.12016.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. SOLOIST: few-shot task-oriented dialog with A single pre-trained auto-regressive model. *CoRR*, abs/2005.05298.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Preprint*, arXiv:2201.11903.

Hanyi Xu, Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Philip S. Yu. 2024. Large Language Models for Education: A Survey. *Preprint*, arXiv:2405.13001.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. *Preprint*, arXiv:1911.00536.

A Prompts

A.1 Prompt template

[INST]

You are a helpful teaching assistant for English as a second language.

ESL students are talking to a chatbot that works with string-matching to complete a task. The chatbot does not always recognize when a user's answer is valid. When it does not, it asks the question again or tries to elicit an answer in a different way.

Based on the conversation history and the instructions of the chatbot task, your job is to output a JSON object, stating whether the last student response is relevant. Importantly, we define "relevant" as providing a plausible answer to the question with all necessary information and which will not interrupt the flow of the conversation. The JSON object should also state whether the message at hand contains a given grammatical construct. If the target construct is "None", the value of this property should be the string "None".

One property in a JSON object must never have more than one value.

Finally, also include information about whether the student's input is factually correct based on the task instructions. (Note that this part of the prompt was only included in the "Comparing Athletes" task).

Ignore typos and grammar issues. Answer based on what you understand the student's answer to be meant to represent.

The JSON object should follow this schema:

```
{
  "relevance": Boolean,
  "factual": Boolean,
  "construct_present": True,
     False or None if the target
     construct is None
}
```

–––––

Here are some examples for how to do this:

`{examples}`

–––––

Now it is your turn:

–––––

TASK INSTRUCTIONS:

`{instructions}`

–––––

CONVERSATION HISTORY:

`{context}`

–––––

STUDENT RESPONSE:

`{student_input}`

–––––

CONSTRUCT:

`{target_construct}`

[/INST]

A.2 K-shot example

Given the following CONTEXT:

BOT: What can I help you with? Student: I would like to buy a gift for my friend

BOT: What does your friend like?

Student: So she likes dresses. Last year. Or

BOT: What does your friend like?

Student: She likes to draw

BOT: What does your friend like?

Student: She likes skirts

BOT: How about this video game console? It is brand new for £400.

Student: Yes that's very good

BOT: How about this video game console? It is brand new for £400

Student: It's very expensive

The OUTPUT should be:

```
{
  "relevance": TRUE,
  "construct_present": FALSE
}
```

A.3 Chain of Thought example

Given the following CONTEXT:

BOT: I heard Andy and Peter are in 1st and 2nd place in the Decathlon! Who ran faster in the 100 meters?

Student: you are Bad in sports

BOT: Was Peter faster or slower than Andy?

Student: idk

BOT: Peter was ....?

Student: faster?

The OUTPUT should be:

```
{
  "relevance_thought": "The bot
     asks how Peter ran compared
     to Andy. The students
     response "faster?" is a
     plausible answer to the
     question. It gives the
     necessary information to the
      bot and allows dialogue to
     continue without
     interruption to the next
     step. Therefore, the answer
     is True.",
  "relevance": TRUE,
  "factual_thought": "Considering
      the data in the
     instructions, peter had a
     time of 13 seconds and andy
     13.5 seconds. 13 seconds is
     the fastest of the two times
     . Peter had the faster time,
      and therefore the student
     is correct that Peter was
     faster.  The answer is True
     .",
  "factual": TRUE,
  "construct_thought": "The
     utterance 'faster?' contains
      the  word 'faster', which
     is a comparative adverb.
     Since this is the target
     construct, the answer is
     True."
  "construct_present": TRUE
}
```

A.4 Comparing Athletes instructions

You and Max are discussing the results of the school Decathlon, a competition with 10 events. You watched Day 1, but Max missed it. Explain what happened on Day 1.
Here is some language that may help you:

1. Peter did better than Andy in the Day 1 events.

2. Peter ran faster than Andy in the 100 meters.

```
{"events": [
    {
      "event": "100 metres",
      "peter": "13 seconds",
      "andy": "13.5 seconds",
    },
    {
      "event": "Long jump",
      "peter": "168 cm",
      "andy": "170 cm"
    },
    {
      "event": "Shot put",
      "peter": "9.1 m",
      "andy": "8.9 m"
    },
    {
      "event": "High jump",
      "peter": "130 m",
      "andy": "150 m"
    },
    {
      "event": "400 metres",
      "peter": "85 seconds",
      "andy": "100 seconds"
    }
  ]
}
```

A.5 Birthday Present task instructions

Your friend Max is having a birthday soon! Let's go shopping to buy him a present. Ask the shop assistant to help you find a gift he will like.Remember that Max likes video games, music, and movies.

Here is some language that may help you: You can use the present progressive form when you are searching for something

1. I am looking for a birthday gift for my friend.

2. I am searching for a birthday present for my friend.

You can use a comparative adjective to ask for more options

1. Do you have something cheaper?