Multilingual Datasets for Custom Input Extraction and Explanation Requests Parsing in Conversational XAI Systems

Qianli Wang[1,2] **Tatiana Anikina**[2,3] **Nils Feldhus**[1,2,5] **Simon Ostermann**[2,3,4]
Fedor Splitt[1] **Jiaao Li**[1] **Yoana Tsoneva**[1] **Sebastian Möller**[1,2] **Vera Schmitt**[1,2]
[1]Technische Universität Berlin [2]German Research Center for Artificial Intelligence (DFKI)
[3]Saarland Informatics Campus [4]Centre for European Research in Trusted AI (CERTAIN)
[5]BIFOLD – Berlin Institute for the Foundations of Learning and Data
Correspondence: qianli.wang@tu-berlin.de

Abstract

Conversational explainable artificial intelligence (ConvXAI) systems based on large language models (LLMs) have garnered considerable attention for their ability to enhance user comprehension through dialogue-based explanations. Current ConvXAI systems are often based on intent recognition to accurately identify the user's desired intention and map it to an explainability method. While such methods offer great precision and reliability in discerning users' underlying intentions for English, a significant challenge in the scarcity of training data persists, which impedes multilingual generalization. Besides, the support for free-form custom inputs, which are user-defined data distinct from pre-configured dataset instances, remains largely limited. To bridge these gaps, we first introduce MultiCoXQL, a multilingual extension of the CoXQL dataset spanning five typologically diverse languages, including one low-resource language. Subsequently, we propose a new parsing approach aimed at enhancing multilingual parsing performance, and evaluate three LLMs on MultiCoXQL using various parsing strategies. Furthermore, we present Compass, a new multilingual dataset designed for custom input extraction in ConvXAI systems, encompassing 11 intents across the same five languages as MultiCoXQL[1]. We conduct monolingual, cross-lingual, and multilingual evaluations on Compass, employing three LLMs of varying sizes alongside BERT-type models.

1 Introduction

To improve the transparency of LLMs while ensuring efficiency and user comprehension, conversational XAI systems have recently emerged (Chromik and Butz, 2021; Lakkaraju et al., 2022; Shen et al., 2023; Bertrand et al., 2023; Mindlin et al., 2024; Feustel et al., 2024; He et al., 2025). Several systems have since been developed, e.g., TALKTOMODEL (Slack et al., 2023), INTERROLANG (Feldhus et al., 2023) and LLMCHECKUP (Wang et al., 2024a). These systems include user interfaces that facilitate users to interact in natural language with a system and rely on intent recognition[2]. Intent recognition is a key upstream component in ConvXAI systems, focusing on accurately interpreting user inputs from multiple perspectives (Chen et al., 2022) and mapping user intents to the corresponding explainability methods, which enables explanations that are as faithful as the underlying method allows (Wang et al., 2024b). Nevertheless, intent recognition remains challenging for ConvXAI, due to the scarcity of training data, particularly multilingual data, and the specialized nature of the XAI domain, which involves mapping requests across a diverse range of XAI methods. Wang et al. (2024b) presents the first and, to-date, largest dataset, CoXQL, for intent recognition in ConvXAI, but it is limited to English. Entering queries in other languages may either result in undesired explanations or prevent users from fully accessing the ConvXAI capabilities, highlighting the need to extend intent recognition to multiple languages for effective use in multilingual scenarios (Gerz et al., 2021; Shi et al., 2022).

Furthermore, CoXQL is restricted to an existing dataset with fixed data points (§4) that can be queried by the user and explained, as in TALKTOMODEL, INTERROLANG, and LLMCHECKUP. The deficiency is that users are unable to freely explore custom input based on their preferences with the explained LLMs, which hinders the generalizability of ConvXAI systems. One goal of ConvXAI systems is to support more adaptive usage (§4), allowing personalization (Orji et al., 2017), user engagement (Irfan et al., 2019), and

[1]Dataset and code are available at: https://github.com/qiaw99/compass

[2]Intent recognition and parsing are used interchangeably in the scope of this work.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 534–555
November 4-9, 2025

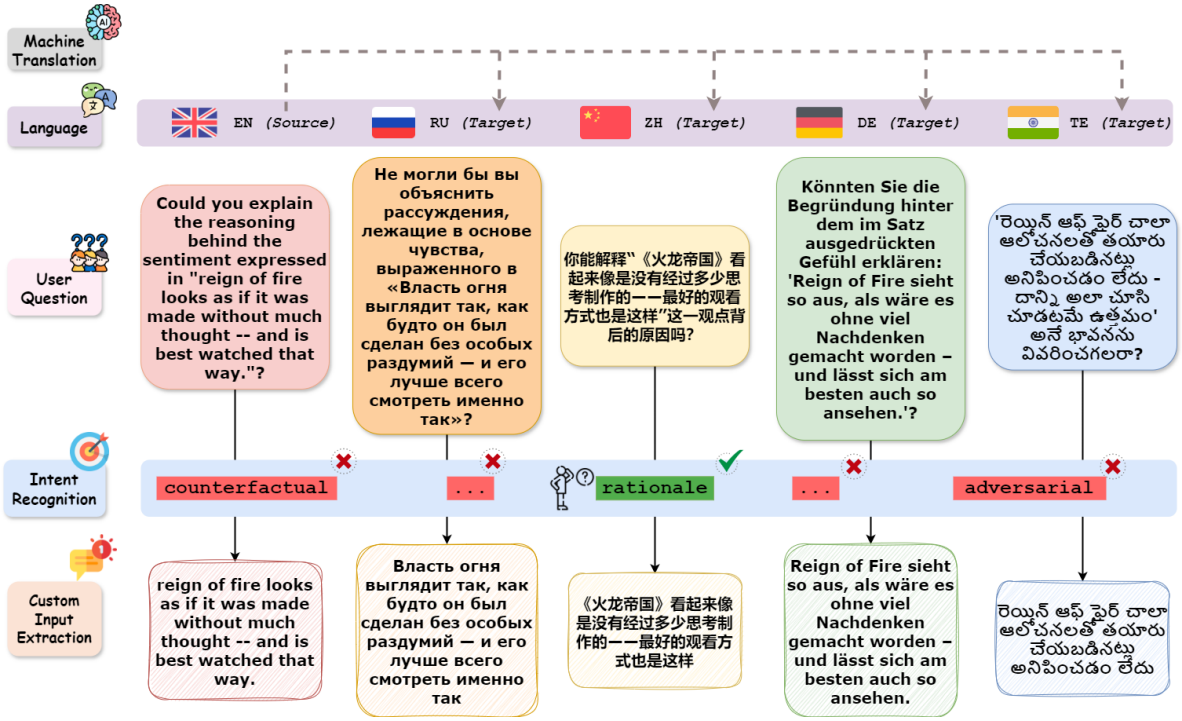

Figure 1: Example parallel utterances from Compass, including the user question, corresponding intent, and extracted custom input, are provided in *English* (EN), *Russian* (RU), *Chinese* (ZH), *German* (DE), and *Telugu* (TE). The example question is requesting reasoning (parsed as "*rationale*", which should provide a natural language explanation) and the corresponding extracted custom input comes from the SST2 dataset.

efficient use of systems (Burkolter et al., 2014). This can be realized by allowing custom input (Figure 1). Nonetheless, the current custom input extraction in the context of ConvXAI has been largely constrained by the lack of suitable datasets.

To address these gaps, we **first** extend CoXQL (Wang et al., 2024b) to support multiple languages, called MultiCoXQL (Figure 2). MultiCoXQL is created by machine translating CoXQL instances, while preserving their intent annotations, covering 5 languages: *German*, *Chinese*, *Russian*, *Telugu* and *English*. We assess the quality of machine translation through human annotators who assess meaningfulness and correctness. The Chinese and German translations are of high quality (Figure 5), and the similarity between the original English text and the translated text improves after human correction. **Secondly**, we evaluate the effectiveness of one baseline and three state-of-the-art parsing approaches in ConvXAI on MultiCoXQL. To improve upon the limited cross-lingual generalization of existing approaches, we propose Guided Multi-prompt Parsing (GMP), which combines existing methods and noticeably enhances multilingual parsing accuracy. **Thirdly**, we present the Compass dataset[3] (Figure 1, Figure 3) for enabling custom input in ConvXAI. It includes user questions, extracted custom inputs, and corresponding intents across the five aforementioned languages. We conduct monolingual, cross-lingual, and multilingual evaluations on Compass using (m)BERT and three decoder-only LLMs. For intent recognition, fine-tuned BERT performs comparably to that of the LLMs and outperforms them in Chinese, German, and Telugu. For custom input extraction, out of four approaches, GOLLIE (Sainz et al., 2024) performs best with smaller LLMs, while naïve few-shot prompting presents the best results with larger LLMs.

2 Related Work

Parsing in ConvXAI Systems In most prior ConvXAI systems (Werner, 2020; Nguyen et al., 2023; Shen et al., 2023), parsing is achieved by comparing the semantic similarity between user queries and a predefined set of example utterances, often resulting in relatively low parsing accuracy. In contrast, TALKTOMODEL (Slack et al., 2023) converts user questions into SQL-like queries for parsing

[3]Abbreviation of "Custom Input Extraction and Explanation Requests Parsing in ConvXAI Systems" (Compass).

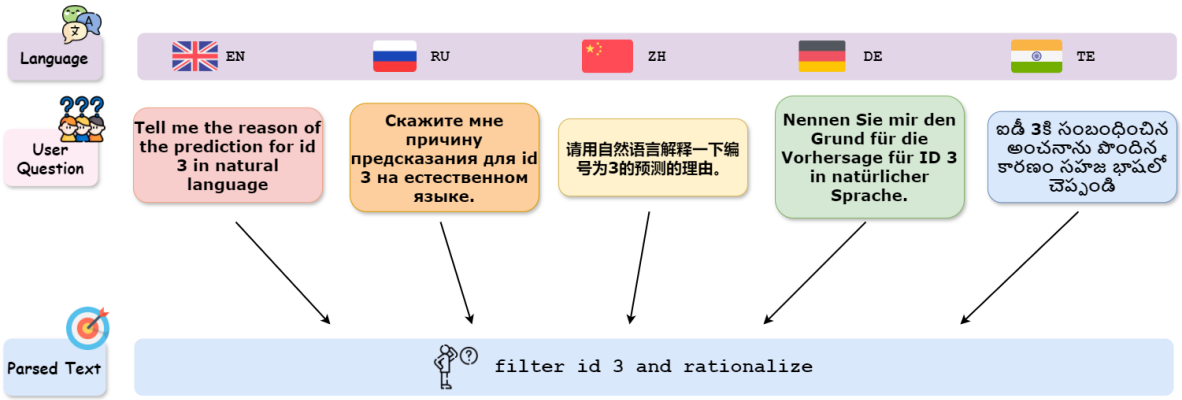

Figure 2: Example parallel utterances from MultiCoXQL, including the user question, and corresponding parsed texts, are provided in *English* (EN), *Russian* (RU), *Chinese* (ZH), *German* (DE), and *Telugu* (TE). The example question seeks reasoning (parsed as "*rationale*") for a specific instance with the id 3, which should return a natural language explanation.

and employs guided decoding (GD) by defining a grammar to constrain the output vocabulary. Similarly, INTERROLANG (Feldhus et al., 2023) uses fine-tuned models for slot tagging to perform parsing, while Wang et al. (2024a) introduces multi-prompt parsing (MP), which hierarchically parses user intent from coarse-grained to fine-grained slots. To ensure compatibility with various XAI operations, MP with template checking (MP+) (Wang et al., 2024b) is proposed to validate the output and fulfill operational requirements. Our work further investigates the effectiveness of GD, MP, and MP+ in multilingual settings and proposes a new approach, Guided Multi-prompt Parsing, which substantially enhances multilingual parsing performance.

Multilingual Dataset Annotation XNLI (Conneau et al., 2018) extend MultiNLI (Williams et al., 2018) to 15 languages, including low-resource languages, to facilitate cross-lingual natural language inference. Min et al. (2019), Tuan Nguyen et al. (2020), and Bakshandaeva et al. (2022) broaden Spider dataset (Yu et al., 2018), a widely recognized text-to-SQL dataset in English that encompasses queries of varying complexity, by translating it into target languages. MultiSpider is subsequently created as a multilingual Text-to-SQL dataset, covering seven popularly used languages (Dou et al., 2023). Hennig et al. (2023) introduce the MultiTACRED dataset, created by translating TACRED (Zhang et al., 2017), a dataset for information extraction, into 12 typologically diverse languages from nine language families. Our work is closely aligned with prior research on multilingual dataset creation and annotation, and adheres to established best practices in the field.

Information Extraction Information extraction can be tackled using in-context learning, which leverages the emergent capability of LLMs (Han et al., 2024). We rely on information extraction approaches to identify custom user inputs in ConvXAI systems. TANL translates between input and output text using an augmented natural language format, with the output later decoded into structured objects (Paolini et al., 2021). GPT-NER reformulate information extraction as a sequence-to-sequence task and special tokens are used to demarcate the boundaries of extracted entities (Wang et al., 2023). GOLLIE employs annotation guidelines represented in code snippet for both input and output (Sainz et al., 2024). This approach is effective when the extracted entities can be represented in a structured or code-like format. In our paper, we employ these approaches to evaluate their efficacy in capturing custom input within ConvXAI systems and across multilingual settings.

3 The MultiCoXQL Dataset

CoXQL (Wang et al., 2024b) is a text-to-SQL dataset for intent recognition in ConvXAI systems and comprises *user questions* and *gold labels* (SQL-like queries) in English. CoXQL covers 31 operations, including explainability and supplementary operations[4] (Table 6). Some operations

[4]Appendix A includes details on operations and examples.

involve additional fine-grained slots[5] and multiple interpretations of the same request, rendering intent recognition in ConvXAI particularly challenging.

The MultiCoXQL dataset introduced in this work encompasses five languages: *English*, *German*, *Russian*, *Chinese*, and *Telugu* (Figure 2). These languages are selected for their typological diversity, representing a spectrum from widely spoken to low-resource languages that use different scripts. Following Hennig et al. (2023) and Popov et al. (2024), we translate the entire train and test splits of CoXQL into the target languages using `Gemini-1.5-pro`[6] (Team, 2024) and name it MultiCoXQL (Figure 2), whose translation quality is evaluated in §7.1. Note we only translate the user questions, not the gold labels, in order to maintain consistency in the label space. Finally, we store the translated instances in the same JSON format as the original CoXQL English dataset.

4 The Compass Dataset

Custom input (Figure 1) refers to user-defined or task-specific data provided to ConvXAI, distinct from instances found in pre-configured datasets (Figure 2). In previous ConvXAI systems, users can only query instances from pre-configured datasets using their dataset ID, while the main challenge in custom input is for LLMs to explicitly interpret and extract relevant information from the user's question. Custom input allows users to explore ConvXAI systems according to their individual preferences, thereby enhancing system *flexibility*, *generalizability*, and *extensibility* (Burkolter et al., 2014; Orji et al., 2017). However, no publicly available dataset currently addresses this type of input for information extraction tasks within ConvXAI systems. Compass is therefore constructed to address this gap and serves as a **synergistic complement** to (Multi)CoXQL, offering users diverse input text formats. To maintain consistent language selection, Compass adopts the same five languages as MultiCoXQL (§3).

[5]E.g., the feature importance method can support various approaches, including *LIME*, *Input x Gradient*, *Integrated Gradients*, and *attention* in CoXQL. The number of data points for each operation depends on whether it involves additional slots.

[6]The prompt instruction to translate user texts into target languages is provided in Appendix B. `Gemini-1.5-pro` is selected, as it supports all target languages that we determine: `https://ai.google.dev/gemini-api/docs/models/gemini#available-languages`.

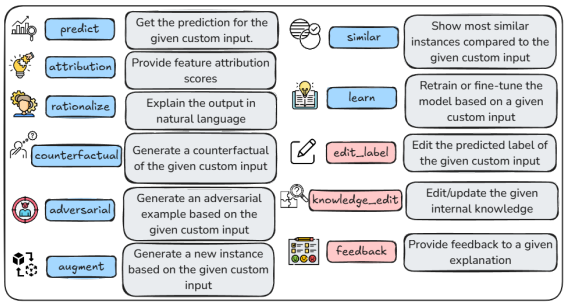

Figure 3: Main operations in the Compass dataset capable of receiving custom input. Operations highlighted in *blue* are collected from the CoXQL dataset. Operations highlighted in *red* are not yet implemented in any ConvXAI system and have been identified from the literature.

4.1 Operations

As shown in Figure 3, we first identify eight operations within CoXQL that should accommodate custom user input to enhance user experience and these operations are highlighted in blue. Additionally, we include three new operations (`edit_label`, `knowledge_edit`, `feedback`), highlighted in red, drawn from the literature (Li et al., 2022; Zhang et al., 2024; Wang et al., 2025). These operations provide users with deeper insights into model behavior by enabling more interactive engagement with the underlying explained model.

4.2 Dataset Construction

Source of Custom Input To preserve the naturalness and usability, we curate custom input from three core NLP tasks - fact-checking, commonsense question answering, sentiment analysis - ensuring all examples remain self-contained within the NLP domain. This approach endows Compass with diverse topics and texts of different complexity and length, offering varying levels of difficulty for LLMs in custom input extraction. The following datasets are selected for each use case: COVID-Fact (Saakyan et al., 2021), ECQA (Aggarwal et al., 2021), and SST2 (Socher et al., 2013)[7].

Dataset Creation Data points in Compass consist of a *user question*, *custom input* and the *corresponding intent*. As a first step, we manually create

[7]COVID-Fact is a fact-checking dataset consisting of claims and evidence, with labels indicating whether a claim is *supported* or *refuted*. ECQA is commonsense question answering dataset encompassing commonsense questions with multiple-choice answers. SST2 is a sentiment analysis dataset which provides movie reviews and corresponding sentiment labels. Examples from each dataset are shown in Figure 9.

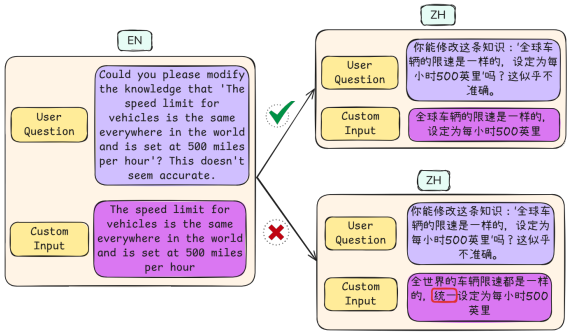

Figure 4: Validation of whether the translated custom input (e.g., in *Chinese*) is fully contained within the translated user question. The words marked in the red box are not included in the translated user question, and thus the translation in the bottom right is invalid.

10 examples per operation in English, similar to the ones in Figure 1, following the approach used in Feldhus et al.'s (2023) study to simulate how questions may naturally arise. This ensures alignment with realistic scenarios while maintaining structural and stylistic diversity across examples. These manually created examples are then used as demonstrations to prompt `Gemini`, guiding it to generate new data points that conform to the desired question formulations and linguistic patterns. The resulting user questions with the custom input are evaluated by checking if they directly pertain to the specified operation and discarded if not applicable. Furthermore, we manually develop a test set adhering to the guidelines (Figure 13), incorporating questions for each included operation (Figure 3). Ultimately, we acquire a **training set** comprising 1089 instances and a **test set** consisting of 109 instances.

4.3 Automatic Translation

Consistent with the translation process for MultiCoXQL (§3), we prompt `Gemini` (Figure 14) to translate both the *user question* and the *custom input* into the target language, with the translation quality subsequently evaluated in §7.1. We then verify that the translated custom input remains fully embedded within the translated user question (Figure 4); if not, the translation process is repeated until a valid result is obtained.

5 Methodology

5.1 MultiCoXQL

In CoXQL, the recognition of XAI intents is treated as a task similar to text-to-SQL (Figure 6), which can be represented and processed as a sequence-to-sequence task (Sutskever et al., 2014). In this work, we benchmark one baseline and three state-of-the-art parsing approaches (§5.1.1) on MultiCoXQL, and propose a new method (§5.1.2) for explanation request parsing in multilingual settings.

5.1.1 Parsing Approaches Selection

Nearest neighbor (NN) determines intents based on the semantic similarity between the user query and existing training samples, using a multilingual `SBERT` model[8], which is trained on all target languages (§3). **Guided decoding** (GD) ensures that the output conforms to predefined grammatical rules and constraints (Shin et al., 2021). **Multi-prompt parsing** (MP) (Wang et al., 2024a) comprises two stages: first, the model is presented with all possible operations in a simplified format to identify the main operation; subsequently, the model is further prompted to populate fine-grained attributes (§3). Unlike GD, MP is not constrained by grammatical rules and thus tends to deviate from the predefined templates for each operation. **Multi-prompt parsing with template checking** (MP+) addresses this issue to some extent by incorporating template validation (Wang et al., 2024b).

5.1.2 Guided Multi-prompt Parsing

To leverage and integrate the strengths of GD and MP, particularly in multilingual settings where existing methods often yield suboptimal performance (Table 1), we propose a simple yet effective approach: Guided Multi-prompt Parsing (GMP) (Figure 8). First, we employ `SBERT` to compute intent centroid embeddings by averaging the embeddings of training examples that share the same intent. We then find which intents are most similar to the user's query using cosine similarity between the intent centroid embeddings and the user query embedding, and retrieve the top-k most similar training examples for each candidate intent based on their similarity to the user input. These retrieved examples are then used to dynamically construct a prompt for generating a coarse-grained intent (e.g., *learn* or *augment*) that corresponds to the supported XAI operations. Next, GMP uses prompting with an intent-specific grammar, guided decoding and additional demonstrations to generate a fine-grained intent with relevant attributes (Table 6). The multi-stage prompting with multiple intent options provides greater flexibility, while guided de-

[8] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

coding ensures that the final structured parse includes the only correct associated attributes[9].

5.2 Compass

5.2.1 Intent Recognition

Compass embodies coarser-grained intents compared to (Multi)CoXQL, with a pronounced focus on custom input extraction (Figure 1). For (m)BERT, we frame intent recognition as a multi-class classification task and fine-tune (m)BERT on the training dataset for a given language. In addition, in-context learning is employed for decoder-only LLMs (§6) to perform intent recognition. Suitable demonstrations are selected based on semantic similarity, measured using a multilingual SBERT, followed by the application of few-shot prompting.

5.2.2 Information Extraction

To facilitate custom input extraction from user requests, we formulate the task as a *sequence labeling* problem, where the custom input embedded within the user request is treated as the target output. We consider four distinct information extraction approaches for identifying custom inputs[10].

Naïve For (m)BERT, custom input extraction is framed as a token-level classification task, whereas for decoder-only LLMs (§6), few-shot prompting is performed using $n = 10$ demonstrations (§5.2.1).

TANL TANL (Paolini et al., 2021) employs predefined inline tagging to annotate entities, thereby capturing structural information within the text.

GPT-NER GPT-NER (Wang et al., 2023) reformulates sequence labeling as a text generation problem, where the model generates augmented text with information marked with special tokens.

GOLLIE GOLLIE (Sainz et al., 2024) leverages annotation guidelines to guide the model, providing detailed instructions on how to annotate specific types of information.

6 Models

We select three open-source, state-of-the-art decoder-only LLMs with increasing parameter sizes from distinct model families: Llama3-8B (AI@Meta, 2024), Phi4-14B (Abdin et al., 2024), and Qwen2.5-72B (Qwen, 2024) to evaluate the

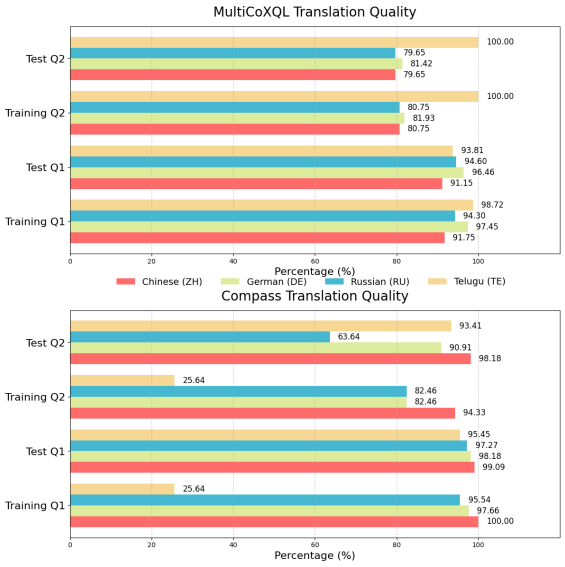

Figure 5: Translation quality of texts in Chinese (ZH), German (DE), Telugu (TE) and Russian (RU) from the training and test sets of MultiCoXQL and Compass, as judged by native speakers.

MultiCoXQL and Compass datasets. These models are selected because they have been trained on at least one of the non-English target languages we chose (§3). In contrast to few-shot prompting, for fine-tuning, we employ BERT and mBERT models (Devlin et al., 2019) to conduct monolingual, cross-lingual, and multilingual evaluations (§7.2)[11].

7 Evaluation

7.1 Machine Translation Human Evaluation

To evaluate the quality of machine translation, we engage 8 in-house native speakers of each target language to meticulously review the translations[12], rectify the translations if necessary, and assess their quality by answering two questions:

- (Q1) *"Does the translated text effectively convey the semantic meaning of the English original, despite minor translation errors?"*
- (Q2) *"Is the overall translation grammatically correct?"*

In addition, we leverage the multilingual SBERT (§5.1), to measure the semantic similarity between the original input and its translations before and after human evaluation.

[9] Further details on GMP can be found in Appendix C.

[10] Prompts for each of the following four approaches are provided in Appendix E.

[11] The details regarding LLMs and language-specific pre-trained (m)BERT models are listed in Appendix F.

[12] Instructions for human evaluation and details about the annotators' background are provided in Appendix G.

7.2 Automatic Evaluation

To evaluate the capability of models in interpreting user intents, we measure the performance of three LLMs (§6), ranging in size from 8B to 72B, using five approaches: NN, GD, MP, MP+, and GMP (§5.1). Intent recognition performance is evaluated by F_1 score on MultiCoXQL and on Compass. In parallel, we evaluate the same LLMs for custom input extraction on Compass using four approaches: Naïve, TANL, GPT-NER, and GOLLIE (§5.2.2). Information extraction performance is likewise reported using F_1 scores. Additionally, we employ BERT and mBERT (Table 7) for monolingual, cross-lingual, and multilingual evaluation of both intent recognition and information extraction on the Compass dataset.

Monolingual We utilize a BERT model pre-trained on the target language and fine-tune it using the training set in the given language.

Cross-lingual We evaluate the performance of a multilingual mBERT model on the test set of each of the four target languages (§3), as well as English, after training it only on the English training set.

Multilingual Simultaneously, we train a multilingual mBERT model on a mixed dataset comprising all languages. mBERT is trained on the full English training split along with a variable proportion of the target language's training split, as proposed by Nag et al. (2021). We vary the amount of target language data used to {10%, 25%, 50%, 75%, 100%} of the available training set.

8 Results and Analysis

8.1 Machine Translation Evaluation

Figure 5 illustrates the translation quality for MultiCoXQL and Compass across all selected languages, highlighting that Gemini performs well overall. Notably, translations into Chinese and German are of relatively high quality compared to Telugu, particularly on Compass. Telugu translations occasionally pose challenges for Gemini, largely due to the semantic complexity of the custom input and language's low-resource status. In addition, as shown in Table 9, the translated texts generally exhibit a high degree of similarity to the original English input. Among the target languages, the German texts have the highest similarity scores, whereas the Telugu texts demonstrate the lowest.

Approach	EN	ZH	DE	RU	TE
NN $_{(Baseline)}$	44.25	44.25	40.71	42.48	25.66

Model	Language	GD	MP	MP+	GMP $_{(Ours)}$
Llama3	EN	63.72	**88.50**	71.68	69.03
	ZH	58.41	43.36	51.33	**72.57**
	DE	44.25	30.97	46.90	**64.60**
	RU	48.67	43.46	52.21	**71.68**
	TE	47.79	27.43	33.63	**51.33**
Phi4	EN	46.02	75.22	61.06	**85.84**
	ZH	48.67	38.94	42.48	**88.50**
	DE	38.94	30.97	30.97	**78.76**
	RU	40.71	26.55	39.20	**84.96**
	TE	53.98	14.16	14.16	**77.88**
Qwen2.5	EN	63.71	91.14	**94.69**	88.50
	ZH	68.14	55.75	57.52	**88.50**
	DE	59.29	46.02	48.67	**77.88**
	RU	69.03	58.41	64.60	**85.84**
	TE	63.72	40.71	42.48	**77.88**

Table 1: Micro-F_1 scores (in %) for different models on MultiCoXQL test set. NN = Nearest Neighbor; GD = Guided Decoding prompted by 20-shots; MP = Multi-prompt Parsing; MP+ = MP with template checks; GMP = Guided Multi-prompt Parsing. **Bold-faced values** indicate the best-performing approach for a given language.

Model	EN	ZH	DE	RU	TE
BERT	87.27	**85.45**	**86.36**	67.27	70.00
Llama3	84.55	50.91	65.45	60.00	53.64
Phi4	88.18	69.09	60.90	70.00	18.18
Qwen2.5	**93.63**	54.55	80.91	**86.36**	**77.27**

Table 2: Micro-F_1 scores (in %) on the Compass dataset are reported for the monolingual setting.

Moreover, after human annotators revise the translations, the similarity improves by up to 9%[13].

8.2 MultiCoXQL

Table 1 reveals that MP and MP+ outperform GD on the English subset of MultiCoXQL, consistent with the findings of Wang et al. (2024a,b). All three approaches significantly surpass the baseline. However, GD generally exhibits superior parsing performance in other languages, especially in Chinese and Telugu, compared to MP and MP+. This discrepancy can be attributed to the limited cross-

[13]Further quality analysis and common error patterns are detailed in Appendix I. Given the challenges associated with recruiting multiple annotators—particularly for low-resource languages—and the relatively straightforward nature of the task for human annotators, we report inter-annotator agreement (IAA) only for the German and Chinese test sets, achieving Krippendorff's α scores of 0.89 and 0.94, respectively.

Data (%)	EN	ZH	DE	RU	TE	Δ
-	87.27	60.91	63.64	52.73	41.82	-
10%	90.00	76.36	73.64	80.00	67.27	16.18
25%	91.82	83.64	87.27	78.18	71.82	21.27
50%	90.00	82.73	83.64	85.45	77.27	22.54
75%	89.09	85.45	83.64	83.64	78.18	22.73
100%	91.82	86.36	92.73	86.36	84.55	27.09

Table 3: Micro-F_1 scores (in %) on Compass for *intent recognition* are reported in the multilingual setting, which are achieved by fine-tuning mBERT on the complete English training split combined with different proportions of the translated target language training split, ranging from 10% to 100%. The final column shows the averaged improvement across languages compared to the cross-lingual evaluation.

Language	Model	Approaches			
		Naïve	GPT-NER	TANL	GOLLIE
EN	BERT	71.96	-	-	-
	Llama3-8B	64.55	58.18	60.91	**66.36**
	Phi4-14B	**85.45**	44.55	44.93	77.27
	Qwen2.5-72B	**89.09**	77.27	68.18	80.91
ZH	BERT	69.82	-	-	-
	Llama3-8B	20.91	60.91	48.18	**62.73**
	Phi4-14B	26.36	50.00	32.73	**73.63**
	Qwen2.5-72B	45.45	**70.00**	50.91	68.18
DE	BERT	76.69	-	-	-
	Llama3-8B	49.09	57.27	52.73	**67.27**
	Phi4-14B	52.73	40.00	38.18	**72.73**
	Qwen2.5-72B	**77.27**	60.91	48.18	70.91
RU	BERT	74.95	-	-	-
	Llama3-8B	54.54	62.73	60.91	**73.64**
	Phi4-14B	63.64	44.55	29.09	**75.45**
	Qwen2.5-72B	**86.36**	68.18	60.91	82.73
TE	BERT	78.38	-	-	-
	Llama3-8B	**22.73**	16.36	17.27	20.91
	Phi4-14B	12.73	2.72	5.00	**21.82**
	Qwen2.5-72B	**36.36**	25.45	18.18	34.55

Table 4: Custom input extraction results (Micro-F_1 scores in %) obtained using Naïve, GPT-NER, TANL, GOLLIE on Compass, with BERT, Llama3-8B, Phi4-14B, and Qwen2.5-72B. Bold-faced values indicate the best-performing approach for a given LLM.

lingual generalizability of current methods (§5.1). For Telugu, GD achieves performance on par with other languages, whereas MP and MP+ exhibit a marked performance decline. Moreover, due to the hierarchical, two-stage parsing nature of MP and MP+, coupled with their lack of grammatical constraints compared to GD, they are more prone to misidentifying the main operation or generating output that fall outside the predefined operation set, thereby hampering further parsing. This issue is partially addressed by our proposed approach, GMP, which performs two-stage parsing similar to MP(+), while constraining the outputs using predefined grammars. As shown in Table 1, GMP consistently outperforms existing methods by an average of 28.31%, particularly demonstrating substantial performance gains across non-English languages and achieves comparable performance on Qwen2.5-72B and Phi4-14B, both of which significantly outperform Llama3-8B. Meanwhile, in English, GMP occasionally underperforms MP(+). This can be attributed to the application of grammars, which limit the flexibility of generation, while ensuring the outputs conforms to predefined grammatical structures.

8.3 Compass

8.3.1 Intent Recognition

Monolingual Evaluation Table 2 illustrates that while LLMs achieve satisfactory accuracy on English data, they struggle to recognize user intents in the Chinese and Telugu subsets of Compass. Model performance generally improves with increasing model size. Furthermore, fine-tuned BERT achieves performance comparable to Qwen2.5-72B, and consistently outperforms Llama3-8B and Phi4-14B in Chinese, German and Telugu, offering an efficient solution for intent recognition. We observe that LLMs occasionally generate labels in the target language instead of English (Figure 15).

Cross-lingual & Multilingual Evaluation Table 3 shows cross-lingual mBERT yields lower performance compared to monolingual BERT, whereas multilingual mBERT consistently outperforms both. Moreover, in the multilingual setting, performance improves as the proportion of non-English training data increases (Δ), with especially compelling performance gains observed for Telugu.

8.3.2 Custom Input Extraction

Monolingual Evaluation Table 4 unveils that extracting custom input in Telugu poses rigorous challenges, with none of the evaluated approaches or models achieving adequate results. For Llama3-8B and Phi4-14B, GOLLIE generally outperforms the naïve approach, GPT-NER, and TANL, most notably in German and Russian, where the performance margin is substantial (with Llama3-8B on Chinese, performance improves by up to 200% when comparing GOLLIE to naïve prompting). In contrast, for Qwen2.5-72B, the naïve approach yields the best results among all considered methods. On Compass, smaller LLMs benefit from GOLLIE, which reformulates the task into structured code snippets, making it more interpretable

Data (%)	EN	ZH	DE	RU	TE	Δ
-	79.74	62.22	68.82	64.54	40.38	-
10%	80.13	80.32	75.49	78.32	80.72	15.86
25%	84.01	91.15	78.94	82.40	80.23	20.21
50%	83.19	92.00	79.23	83.56	82.06	20.87
75%	84.15	92.87	80.18	84.16	83.42	21.82
100%	85.11	92.99	79.35	83.65	85.16	22.11

Table 5: Micro-F_1 scores (in %) on Compass for *custom input extraction* are reported in the multilingual setting, which are achieved by fine-tuning `mBERT` on the complete English training split combined with different proportions of the translated target language training split, ranging from 10% to 100%. The final column shows the averaged improvement across languages compared to the cross-lingual evaluation.

for models (Sainz et al., 2024). Conversely, larger models appear more susceptible to distraction from newly introduced patterns (Figure 10, Figure 11). In addition, fine-tuned `BERT` exhibits competitive performance across all target languages. For non-English subsets, `BERT` generally outperforms LLMs across most approaches, particularly in Telugu, where it achieves more than double the performance of `Qwen2.5-72B` using the naïve approach.

Cross-lingual & Multilingual Evaluation As shown in Table 5, cross-lingual `mBERT` exhibits lower performance compared to monolingual `BERT`, aligned with the results observed in the intent recognition task, in particular with a pronounced performance gap in Telugu. As the proportion of non-English data increases, the performance improvement trend is similar to that shown in Table 3, with Telugu benefiting the most. Besides, multilingual `mBERT` consistently outperforms all LLMs across nearly all languages, with the exception of English.

Error Analysis Figure 16 illustrates common error patterns observed in the custom input extraction outputs from LLMs. In some cases, LLMs tend to generate or substitute words that do not appear in the original user question, extract only part of the intended custom input, or inadvertently include parts of artifacts from extraction methods in the final output. Additionally, there are instances where LLMs fail to solve the task altogether due to the task's inherent difficulties for LLMs.

9 Conclusion

In this work, we first extend the CoXQL dataset for intent recognition in ConvXAI to a multilingual version, MultiCoXQL, covering five languages, including one low-resource language, using machine translation followed by human evaluation and correction. We benchmark state-of-the-art explanation request parsing approaches on MultiCoXQL using three different LLMs. Second, we propose a new approach, Guided Multi-Prompt Parsing, which integrates the strengths of existing methods and substantially improves parsing accuracy in multilingual settings. Third, we introduce the Compass dataset for coarse-grained intent recognition and custom input extraction in ConvXAI, incorporating the same five languages as MultiCoXQL. We conduct comprehensive experiments using three LLMs, along with `(m)BERT`, on Compass, to evaluate performance in monolingual, cross-lingual, and multilingual scenarios. We observe that cross-lingual `mBERT` underperforms compared to monolingual `mBERT`, whereas multilingual `mBERT` outperforms both. For the task of custom input extraction, GOLLIE proves to be more effective for smaller LLMs, while naïve few-shot prompting yields better results with larger LLMs.

Limitations

A key limitation of this work is its dependence on a machine translation (MT) system, i.e., `Gemini-1.5-pro`, to obtain high-quality translations for the MultiCoXQL and Compass datasets. Depending on the availability of linguistic resources and the quality of the MT model for a specific language pair, the translations used for training and evaluation may contain inaccuracies, although these translations have been assessed and rectified if necessary by human annotators.

We do not implement the operations highlighted in red, introduced in Section 4.1; their actual implementation and integration into ConvXAI systems are left for future work.

Given the difficulties involved in recruiting multiple annotators - especially for low-resource languages - and considering the relatively straightforward nature of the annotation task, we limit our reporting of inter-annotator agreement (IAA) to only the German and Chinese test sets (§4.2).

We do not extensively experiment with every model from different model families; rather, we select three widely used models of varying sizes (§6).

The current state-of-the-art ConvXAI systems are typically designed to support a set of representative and widely used XAI methods, from which we

determined the current set of 11 XAI approaches. Furthermore, extending the set of XAI operations is a highly involved process – for example, it requires collecting user questions, translating them from English into all target languages, conducting user studies to evaluate translation quality, and recruiting annotators (which is particularly challenging for low-resource languages, such as Telugu in our case).

Ethics Statement

The participants in the machine translation evaluation were compensated at or above the minimum wage, in accordance with the standards of our host institutions' regions. The annotation took each annotator approximately 8 hours on average.

Acknowledgment

We thank Alon Drobickij and Selin Yeginer for reviewing the German translation, Polina Danilovskaia for reviewing the Russian translation and Ravi Kiran Chikkala for reviewing the Telugu translation. We would extend our gratitude to Lisa Raithel for setting up and organizing the translation quality check for the German translation.

Additionally, we are indebted to the anonymous reviewers of EMNLP 2025 for their helpful and rigorous feedback. This work has been supported by the Federal Ministry of Research, Technology and Space (BMFTR) as part of the projects BIFOLD 24B, TRAILS (01IW24005), VERANDA (16KIS2047) and newspolygraph (03RU2U151C).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

AI@Meta. 2024. Llama 3 model card.

Daria Bakshandaeva, Oleg Somov, Ekaterina Dmitrieva, Vera Davydova, and Elena Tutubalina. 2022. PAUQ: Text-to-SQL in Russian. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2355–2376, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Astrid Bertrand, Tiphaine Viard, Rafik Belloum, James R. Eagan, and Winston Maxwell. 2023. On selective, mutable and dialogic xai: A review of what users say about different types of interactive explanations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Dina Burkolter, Benjamin Weyers, Annette Kluge, and Wolfram Luther. 2014. Customization of user interfaces to reduce errors and enhance user acceptance. *Applied Ergonomics*, 45(2, Part B):346–353.

Zhi Chen, Lu Chen, Bei Chen, Libo Qin, Yuncong Liu, Su Zhu, Jian-Guang Lou, and Kai Yu. 2022. UniDU: Towards a unified generative dialogue understanding framework. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 442–455, Edinburgh, UK. Association for Computational Linguistics.

Michael Chromik and Andreas Butz. 2021. Human-XAI interaction: a review and design principles for explanation user interfaces. In *Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part II 18*, pages 619–640. Springer.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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.

Longxu Dou, Yan Gao, Mingyang Pan, Dingzirui Wang, Wanxiang Che, Dechen Zhan, and Jian-Guang Lou. 2023. Multispider: Towards benchmarking multilingual text-to-sql semantic parsing. In *AAAI Conference on Artificial Intelligence*.

Nils Feldhus, Qianli Wang, Tatiana Anikina, Sahil Chopra, Cennet Oguz, and Sebastian Möller. 2023.

InterroLang: Exploring NLP models and datasets through dialogue-based explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5399–5421, Singapore. Association for Computational Linguistics.

Isabel Feustel, Niklas Rach, Wolfgang Minker, and Stefan Ultes. 2024. Enhancing model transparency: A dialogue system approach to XAI with domain knowledge. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 248–258, Kyoto, Japan. Association for Computational Linguistics.

Daniela Gerz, Pei-Hao Su, Razvan Kusztos, Avishek Mondal, Michał Lis, Eshan Singhal, Nikola Mrkšić, Tsung-Hsien Wen, and Ivan Vulić. 2021. Multilingual and cross-lingual intent detection from spoken data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7468–7475, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. 2024. An empirical study on information extraction using large language models. *Preprint*, arXiv:2305.14450.

Gaole He, Nilay Aishwarya, and Ujwal Gadiraju. 2025. Is conversational xai all you need? human-ai decision making with a conversational xai assistant. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, page 907–924, New York, NY, USA. Association for Computing Machinery.

Leonhard Hennig, Philippe Thomas, and Sebastian Möller. 2023. MultiTACRED: A multilingual version of the TAC relation extraction dataset. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Toronto, Canada. Association for Computational Linguistics.

Bahar Irfan, Aditi Ramachandran, Samuel Spaulding, Dylan F. Glas, Iolanda Leite, and Kheng Lee Koay. 2019. Personalization in long-term human-robot interaction. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 685–686.

Raviraj Joshi. 2023. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *Preprint*, arXiv:2211.11418.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *Preprint*, arXiv:1905.07213.

Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chenhao Tan, and Sameer Singh. 2022. Rethinking explainability as a dialogue: A practitioner's perspective. *HCAI @ NeurIPS 2022*.

Zichao Li, Prakhar Sharma, Xing Han Lu, Jackie Cheung, and Siva Reddy. 2022. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 926–937, Dublin, Ireland. Association for Computational Linguistics.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for Chinese SQL semantic parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3652–3658, Hong Kong, China. Association for Computational Linguistics.

Dimitry Mindlin, Amelie Sophie Robrecht, Michael Morasch, and Philipp Cimiano. 2024. Measuring user understanding in dialogue-based xai systems. In *ECAI 2024*, pages 1148–1155. IOS Press.

Arijit Nag, Bidisha Samanta, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2021. A data bootstrapping recipe for low-resource multilingual relation classification. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 575–587, Online. Association for Computational Linguistics.

Van Bach Nguyen, Jörg Schlötterer, and Christin Seifert. 2023. From black boxes to conversations: Incorporating XAI in a conversational agent. In *Explainable Artificial Intelligence*, pages 71–96, Cham. Springer Nature Switzerland.

Rita Orji, Kiemute Oyibo, and Gustavo F. Tondello. 2017. A comparison of system-controlled and user-controlled personalization approaches. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, page 413–418, New York, NY, USA. Association for Computing Machinery.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Dmitrii Popov, Egor Terentev, and Igor Buyanov. 2024. Be my donor. transfer the nlp datasets between the languages using llm. *Preprint*, arXiv:2410.14074.

Qwen. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*.

Hua Shen, Chieh-Yang Huang, Tongshuang Wu, and Ting-Hao Kenneth Huang. 2023. ConvXAI: Delivering heterogeneous AI explanations via conversations to support human-AI scientific writing. In *Computer Supported Cooperative Work and Social Computing*, CSCW '23 Companion, page 384–387, New York, NY, USA. Association for Computing Machinery.

Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. XRICL: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5248–5259, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7699–7715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dylan Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh. 2023. Explaining machine learning models with interactive natural language conversations using TalkToModel. *Nature Machine Intelligence*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A pilot study of text-to-SQL semantic parsing for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4079–4085, Online. Association for Computational Linguistics.

Qianli Wang, Tatiana Anikina, Nils Feldhus, Josef Genabith, Leonhard Hennig, and Sebastian Möller. 2024a. LLMCheckup: Conversational examination of large language models via interpretability tools and self-explanations. In *Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 89–104, Mexico City, Mexico. Association for Computational Linguistics.

Qianli Wang, Tatiana Anikina, Nils Feldhus, Simon Ostermann, and Sebastian Möller. 2024b. CoXQL: A dataset for parsing explanation requests in conversational XAI systems. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1410–1422, Miami, Florida, USA. Association for Computational Linguistics.

Qianli Wang, Tatiana Anikina, Nils Feldhus, Simon Ostermann, Sebastian Möller, and Vera Schmitt. 2025. Cross-refine: Improving natural language explanation generation by learning in tandem. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1150–1167, Abu Dhabi, UAE. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *Preprint*, arXiv:2304.10428.

Christian Werner. 2020. Explainable ai through rule-based interactive conversation. In *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Ningyu Zhang, Yunzhi Yao, and Shumin Deng. 2024. Knowledge editing for large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 33–41, Torino, Italia. ELRA and ICCL.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling.

In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A The CoXQL Dataset

A.1 Operations

Table 6 demonstrations all operations supported in CoXQL.

A.2 Examples

Figure 6 displays the examples from the CoXQL dataset.

B Prompt Instruction for Machine Translation

Figure 7 shows the prompt instruction used to perform the machine translation with `Gemini-1.5-pro` for the MultiCoXQL dataset.

C Guided Multi-prompt Parsing

Figure 8 illustrates the workflow of the Guided Multi-prompt Parsing (GMP) approach. This method combines the strengths of the Multi-prompt Parsing (Wang et al., 2024a) and Guided Decoding (Slack et al., 2023) that proved to be effective for explanation request parsing in ConvXAI systems on the English data, but achieve substantially worse performance on other languages (Table 1).

First, GMP computes centroid embeddings using a multilingual SentenceTransformer model[14] for each intent (step (1) in Figure 8). Next, the user input is encoded with the same model and GMP retrieves the top k most similar intents based on the cosine similarity between their centroid embeddings and the user query in step (2). In this way, we can have a selection of multiple candidate intents that are similar to the user query, but we are not restricted to a single most similar intent (e.g., *nlpattribute, influence, etc.* can be chosen as candidates for the user question *"Show me 10 most important samples for ID 68."*). While similarity-based intent selection is also used in Guided Decoding (Slack et al., 2023), the key difference is that GMP uses the retrieved candidate intents to dynamically construct a prompt in step (3) that includes demonstrations for each of the candidates. Meanwhile, GMP excludes any dissimilar intents, so that the prompt is more concise and relevant to the input. GMP also uses a simplified intent-only grammar with guided decoding to make sure that the generated labels are from a pre-defined set. Note that the simplified grammar does not specify any attributes, only the main XAI operation (e.g. *influence*).

Finally, in step (4), GMP refines the initial coarse-grained intent annotation and prompts the model with more examples for the selected intent to fill in the missing attributes. This step is similar to Multi-prompt Parsing (Wang et al., 2024a), but instead of relying on a single grammar that covers all operations and their attributes, GMP uses an intent-specific grammar based on the selected intent from step (3) to ensure that we do not generate any attributes that are not valid for the selected XAI operation.

GMP is a flexible approach that leverages the advantages of multi-stage prompting that iteratively refines the predictions based on relevant demonstrations and guided decoding that constrains generated outputs. Thus, GMP generally achieves the best results on the MultiCoXQL dataset in the multilingual setting (Table 1).

D Sample Dataset Examples

Figure 9 presents examples of datasets (§4.2) from which custom inputs are collected.

E Custom Input Extraction

Figure 10 and Figure 11 show the prompt instructions for `Naïve`, `TANL`, `GPT-NER` and `GOLLIE` approaches.

F Models

F.1 Pre-trained BERT-type Models

Table 7 lists detailed information about used `(m)BERT` models in our experiments. Fine-tuning `(m)BERT` models for monolingual, cross-lingual, and multilingual evaluations can be completed within 20 minutes.

F.2 Decoder-only LLMs

Table 8 presents details of the three LLMs used in our experiments (§6), including model sizes and corresponding URLs from the Hugging Face Hub. All models were directly obtained from the Hugging Face repository. All experiments were conducted using A100 or H100 GPUs. For each model, experiments on MultiCoXQL can be completed within 15 minutes. For each model, experiments on Compass can be completed within 20 minutes.

[14] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

	Operation	Description/Request
Loc.Pr.	**predict**(instance)	Get the prediction for the given instance
	likelihood(instance)	Calculate the model's confidence (or likelihood) on the given instance
Glob.Pr.	mistake({sample|count}, subset)	Count or show incorrectly predicted instances
	score(subset, metric)	Determine the relation between prediction and labels
Loc. Expl.	**nlpattribute**(inst., topk, method)	Provide feature attribution scores
	rationalize(inst.)	Explain the output/decision in natural language
	influence(inst., topk)	Provide the most influential training data instances
Pertrb.	**cfe**(instance)	Generate a counterfactual of the given instance
	adversarial(instance)	Generate an adversarial example based on the given instance
	augment(instance)	Generate a new instance based on the given instance
Data	show(instance)	Show the contents of an instance
	countdata(list)	Count instances
	label(dataset)	Describe the label distribution
	keywords(topk)	Show most common words
	similar(instance, topk)	Show most similar instances
Mod.	editlabel(instance)	Change the true/gold label of a given instance
	learn(instance)	Retrain or fine-tune the model based on a given instance
	unlearn(instance)	Remove or unlearn a given instance from the model
Meta	function()	Explain the functionality of the system
	tutorial(op_name)	Provide an explanation of the given operation
	data()	Show the metadata of the dataset
	model()	Show the metadata of the model
	domain(query)	Explain terminology or concepts outside of the system's functionality, but related to the domain
Filter	filter(id)	Access single instance by its ID
	predictfilter(label)	Filter the dataset according to the model's predicted label
	labelfilter(label)	Filter the dataset according to the true/gold label given by the dataset
	lengthfilter(level, len)	Filter the dataset by length of the instance (characters, tokens, ...)
	previousfilter()	Filter the dataset according to outcome of previous operation
	includes(token)	Filter the dataset by token occurrence
Logic	and(op1, op2)	Concatenate multiple operations
	or(op1, op2)	Select multiple filters

Table 6: Main operations in CoXQL, including exlainability (*Local Explanation*, *Perturbation*, *Modification*) and supplementary (*Local Prediction*, *Global Prediction*, *Data*, *Meta*, *Filter*, *Logic*) operations. Operations designated in **bold** should facilitate custom input and, therefore, be selected for integration with Compass dataset.

Name	Language	Citation	Size	Link
BERT	English	(Devlin et al., 2019)	110M	https://huggingface.co/google-bert/bert-base-uncased
BERT	German	(Devlin et al., 2019)	110M	https://huggingface.co/google-bert/bert-base-german-cased
BERT	Chinese	(Devlin et al., 2019)	110M	https://huggingface.co/google-bert/bert-base-chinese
BERT	Russian	(Kuratov and Arkhipov, 2019)	110M	https://huggingface.co/DeepPavlov/rubert-base-cased
BERT	Telugu	(Joshi, 2023)	110M	https://huggingface.co/l3cube-pune/telugu-bert
mBERT	Multilingual	(Devlin et al., 2019)	110M	https://huggingface.co/google-bert/bert-base-multilingual-cased

Table 7: Detailed information about used BERT and mBERT models in our experiments.

Name	Citation	Size	Link
Llama3	AI@Meta (2024)	8B	https://huggingface.co/meta-llama/Meta-Llama-3-8B
Phi4	Abdin et al. (2024)	14B	https://huggingface.co/microsoft/phi-4
Qwen2.5	Qwen (2024)	72B	https://huggingface.co/Qwen/Qwen2.5-72B

Table 8: Detailed information about used LLMs in our experiments.

G Human Evaluation Instructions for Translation Quality

Figure 12 presents the instructions for human evaluation, which are used to guide annotators in assessing the quality of translations. All participants have a background in computational linguistics or computer science, hold at least a bachelor's degree, and are proficient in English. In addition, they are native speakers of one of the target languages (§3).

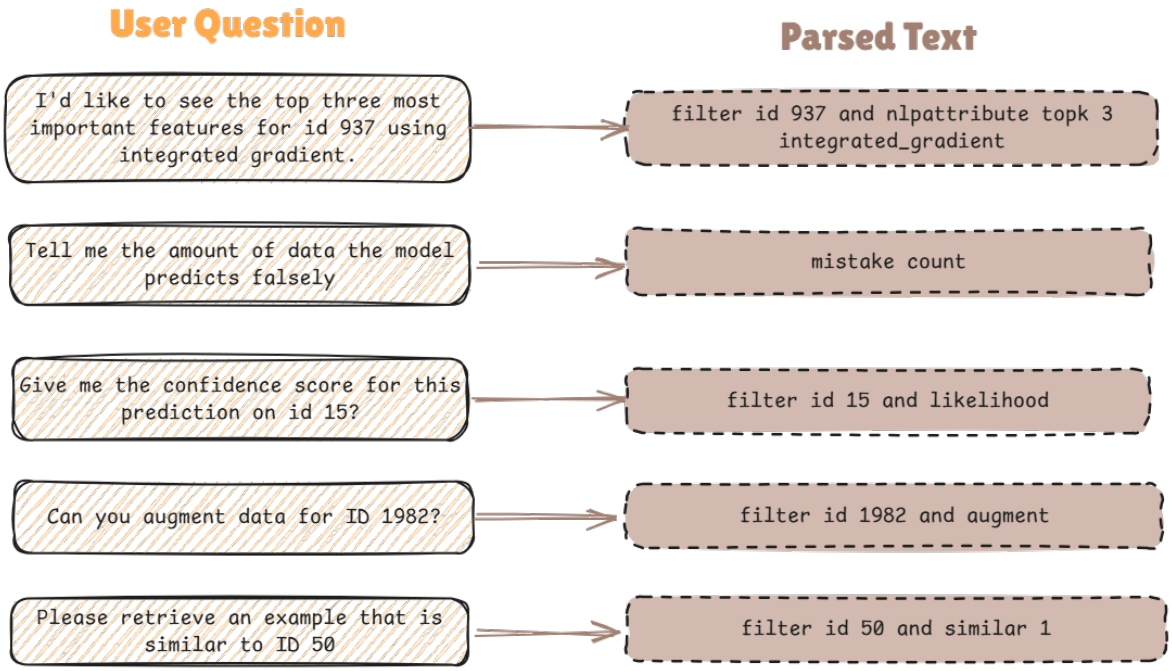

Figure 6: Examples in CoXQL consist of user questions along with their corresponding parsed text, which are used to address various tasks such as feature importance, identifying mistakes, likelihood analysis, data augmentation, and instance similarity.

Prompt Instruction

```
system_prompt = "You are an excellent translator."
task_instruction = f"Please translate the following text into {language}. Provide only the translated texts: {original_input}
prompt = f"{system_prompt} {task_instruction}"
```

Figure 7: Prompt instruction for machine translation.

H Annotation Guideline for Creating the Compass Test Set

Figure 13 shows the annotation instruction for creating the Compass test set.

I Translation Quality Analysis

I.1 Chinese Translation

For the *Chinese* translation, we found that `Gemini-1.5-pro` sometimes omits prepositions. For example, *"For id 9, what are the other 3 instances that are similar to it?"* is translated as "ID 为9 的条目，还有哪3 个类似的实例?", where the preposition *"for"* is missing. Meanwhile, some words, which could have multiple meanings, such as *"item"*, may be translated to a meaning that does not fit our context ("商品" - *"commodity/merchandise"*). Additionally, domain-specific terms, such as *"accuracy score"* are often translated literally ("得分准确率") rather than using the correct predefined terminology ("准确率评分").

I.2 Russian Translation

For the *Russian* translation, we found three different categories of errors. The first category corresponds to the lack of context and ambiguous terms, e.g. *"gold labels"* can be translated into *"gold label stickers"* ("золотые этикетки") and *"item ids"* into *"product identifiers"* which is an acceptable translation but in a different setting. For instance, *"Can you show me the item IDs in the training data?"* was translated into "Вы можете показать мне идентификаторы товаров в обучающих данных?" This category of errors also includes ambiguous terms or terms with multiple possible meanings. E.g., both precision and accu-

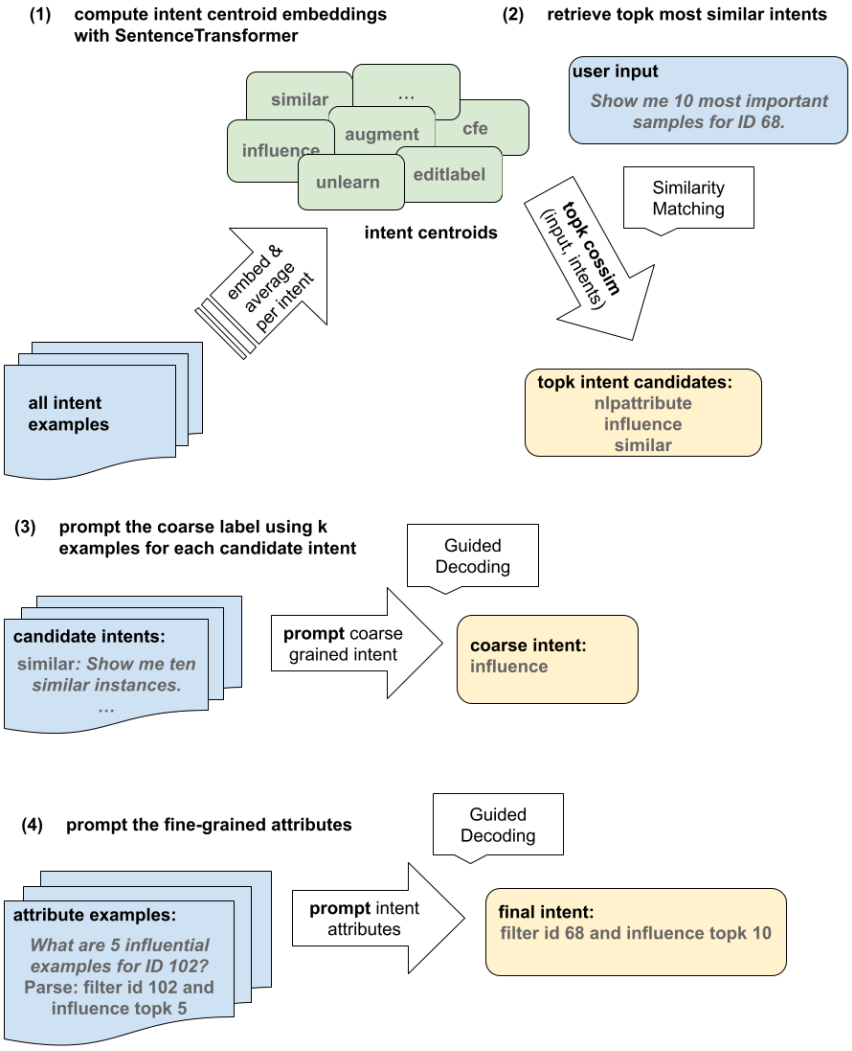

Figure 8: Pipeline of Guided Multi-prompt Parsing approach.

racy can be translated as "точность" in Russian but in our setting they should refer to two different metrics.

The second category relates to the domain-specific terminology and abbreviations. For instance, NLP is sometimes directly transliterated as "НЛП" which is used in Russian as an abbreviation for *"Neural Linguistic Programming"*, not *"Natural Language Processing"*. Also, *"adversarial examples"* are frequently mistranslated as "противоборствующие примеры" instead of a commonly used term "состязательные примеры".

The third category includes all errors caused by the model's failure to correctly interpret the task. Sometimes the output is text in English saying *"Please provide the text..."*. Interestingly, this happens most often for the rationalization operation examples, the model may get confused by the new "instruction" contained in the input and it tries to accomplish the task instead of doing a simple translation. E.g., for *"offer a plain-English interpretation for id 201"* it outputs *"Please provide the text you would like me to translate. I need the text to be able to translate it to Russian and offer an interpretation for id 201."*.

Additionally, the model frequently confuses instrumental and dative cases in Russian. It also misapplies adjective genders, using a single form when different genders are required. Pronoun coreference is often incorrect, leading to misinterpretations. Moreover, the model struggles with voice usage, incorrectly applying passive where active is

ECQA (Commonsense Question Answering)

Question: He had a lot on his plate opening business, this cause a lot of what?
Choices: headaches, making money, success, failure, stress

COVIDFact (Medical Fact Checking)

Claim: Measuring sars-cov-2 neutralizing antibody activity using pseudotyped and chimeric viruses
Evidence: While each surrogate virus exhibited subtle differences in the sensitivity with which neutralizing activity was detected, the neutralizing activity of both convalescent plasma and human monoclonal antibodies measured using each virus correlated quantitatively with neutralizing activity measured using an authentic SARS-CoV-2 neutralization assay. Here, we describe a collection of approaches based on SARS-CoV-2 spike-pseudotyped, single-cycle, replication-defective human immunodeficiency virus type-1 (HIV-1) and vesicular stomatitis virus (VSV), as well as a replication-competent VSV/SARS- CoV-2 chimeric virus.

SST2 (Sentiment Analysis)

Review: Allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.

Figure 9: Examples of ECQA, COVIDFact, and SST2 datasets, from which custom inputs are collected.

Naïve

You will be given a user question related to explainability. Your task is to identify and extract the custom input from this question. The custom input refers to the specific information provided by the user that is necessary to fulfill their request. Extracting this input is crucial for processing user questions and taking appropriate actions. Please return only the custom input as a text string. If no custom input is clearly present, return an empty string. Below are some examples:

[User Question] user question

[Custom Input] custom input

TANL

You will be given a user question related to explainability. Your task is to identify and extract the custom input from this question. The custom input refers to the specific information provided by the user that is necessary to fulfill their request. Extracting this input is crucial for processing user questions and taking appropriate actions. Use the format '[extracted_text | custom_input]' to annotate the custom input in the output. Please return a text string with the custom input marked with [extracted_text | custom_input]. If no custom input is clearly present, return an empty string. Below are some examples:

[User Question] user question

[Custom Input] custom input

Figure 10: The prompt instructions for Naïve and TANL in custom input extraction.

GPT-NER

You are an excellent linguist. You will be given a user question related to explainability. The task is to label the custom input in the given user question. The custom input refers to the specific information provided by the user that is necessary to fulfill their request. Extracting this input is crucial for processing user questions and taking appropriate actions. Use special tokens @@## to mark the extracted phrase in your response. Please return a text string with the custom input marked with @@##. If no custom input is clearly present, return an empty string. Below are some examples:

[User Question] user question

[Custom Input] custom input

GOLLIE

You will be given a user question related to explainability. Your task is to identify and extract the custom input from this question. Please return a list of custom input.If no custom input is clearly present, return an empty list.Below is the schema for the custom input annotation:

@dataclass
class CustomInput(Entity):
"""
The custom input refers to the specific information provided by the user that is necessary to fulfill their request.
Extracting this input is crucial for processing user questions and taking appropriate actions.
"""

[User Question] user question

[Custom Input] custom input

Figure 11: The prompt instructions for GPT-NER and GOLLIE in custom input extraction.

You will find translated CoXQL dataset in JSON format with the naming `{language}_CoXQL_{train/test}.json`, where language could be ***Chinese***, ***English***, ***German***, and ***Russian***.

Your tasks are:

1. *Does the translated text meaningfully express the semantic relation of the English original, regardless of minor translation errors?* If not, set "*meaningful*" to **0**
2. *Check whether the translation done by Gemini 1.5-pro is correct.* If you find any translation is not accurate or incorrect, please change it accordingly, and also set "*correctness*" to **0**.

Please create a new branch and pull request for submission. In addition, please mention the findings you observe, e.g. **the frequent error/incorrect patterns**.

Figure 12: Instructions for human evaluation given to human annotators.

needed and vice versa. It makes errors in verb aspect, confusing perfective and imperfective forms. Some translations sound unnatural due to weak word choices and direct, word-for-word rendering. Furthermore, the model sometimes applies English grammatical structures in a way that is ungrammatical in Russian.

Annotation Guideline

This file provides instructions for creating a test set for Compass. Assume you are an end-user interacting with a conversational explainable AI system that incorporates various explainability and supplementary methods (e.g., adversarial examples, data augmentation, counterfactual examples, edit prediction, feature importance, feedback, knowledge editing, learning from data points, prediction, rationale generation, and identifying similar examples). You may pose questions to verify facts or request explanations. Record your requests as custom input, specifying the operation_name for your intent and the specific question as custom_input, avoiding unnecessary words (e.g., interjections or greetings).

The data should have the same structure as it in `Compass/data`.

```
[
  {
    "idx": 0,
    "operation_name": operation_name,
    "custom_input": custom_input,
    "user_question": user_question
  }
]
```

Note that `edit_prediction` has one additional field `content`:

```
[
  {
    "idx": 0,
    "operation_name": operation_name,
    "custom_input": custom_input,
    "user_question": user_question,
    "content": content
  }
]
```

Figure 13: The annotation guideline for human annotators in creating the Compass test set.

I.3 Telugu Translation

In some instances, the system prompt translated into Telugu is included in the final translation, and there are cases where no direct equivalent exists for certain English words. For example, the word “unlearn” does not have an equivalent term in Telugu and is therefore replaced with a word meaning “forget”. In other cases, the translation may be entirely different, with English words being substituted by phrases or terms that carry a similar meaning but are distinct in structure.

I.4 German Translation

Translation errors can occur in various forms, such as *incorrect use of articles*, *noun gender*, and *case declination*, particularly when shorter substrings could be valid in different forms. Entire sentences may be omitted, especially if they resemble others in meaning within the same text, while idioms like "edge of your seat" or "keeping me guessing" are often translated literally instead of using equivalent expressions, leading to awkward or incorrect phrases in German (e.g.,“der Humor war flach” instead of “der Humor zündete nicht”). Other issues include splitting German compound verbs like “angeben” into “geben ... an” or omitting verbs in complex sentences, failing to adapt English words like “all” when a similar German word doesn’t fit the context, and even translating terms into antonyms, such as “forgettable” becoming “unvergesslich”. Additionally, noun combinations may be mishandled (e.g., “Sicherheits Ergebnisse” instead of “Sicherheitsergebnisse”), and while not outright wrong, translations can suffer from poor style—marked by excessive comma use that hampers readability and simpler, less elegant phrasing like “von wo” instead of “woher”. These flaws often result in text that feels unnatural or unclear to native speakers, despite conveying the intended meaning.

I.5 Semantic Similarity Comparison

Table 9 shows the semantic similarity between the original input in English and the translated text in target languages.

J Compass Dataset Translation

The prompt used for `Gemini-1.5-pro` to translate texts from English to target languages is demonstrated in Figure 14.

Prompt for Machine Translation

The uploaded JSON consists of three fields: user_question, operation_name, and custom_input. The custom_input field is derived from user_question and serves as a simplified version by discarding all redundant information. Your task is to translate both user_question and custom_input into language while keeping operation_name as 'operation_name'. Note that after the translation, the translated custom_input must remain a part of the translated user_question.

Figure 14: The prompt used to translate user question and custom input into target languages for the Compass dataset.

Dataset	Set	ZH	DE	TE	RU
MultiCoXQL	***Before Correction***				
	Train	83.66%	84.18%	55.28%	82.74%
	Test	83.46%	85.36%	54.92%	83.21%
	After Correction				
	Train	84.25%	85.83%	53.12%	83.87%
	Test	82.66%	85.29%	54.58%	83.33%
Compass	***Before Correction***				
	Train	85.75%	81.07%	37.56%	82.73%
	Test	84.51%	87.50%	70.14%	86.74%
	After Correction				
	Train	85.97%	88.37%	37.86%	85.07%
	Test	84.51%	89.12%	73.59%	88.02%

Table 9: Semantic similarity between the original input in English and the translated texts in Chinese (ZH), German (DE), Telugu (TE) and Russian (RU) from the training and test sets of MultiCoXQL and Compass measured by a multilingual sentence transformer.

K Error Analysis

K.1 Compass: Parsing

Figure 15 shows examples, where the LLMs generate labels (`importance`) in the target languages (*Chinese* and *Russian*) instead of in English.

K.2 Compass: Custom Input Extraction

Figure 16 includes 4 example pairs in English, German, Chinese, and Telugu, each consisting of a user question, the corresponding ground-truth custom input, the predicted custom input, and the approach used (Naïve, TANL, GOLLIE, GPT-NER). Figure 16 highlights several recurring mistakes in LLM-generated custom input extraction. Sometimes the models insert or replace terms that weren't in the user's original query, capture only a fragment of the desired input, or accidentally carry over artifacts from the extraction process into their output. There are also cases where, despite having ample examples to guide them, the LLMs simply fail to perform the extraction task. While Figure 16 reveals that GOLLIE struggles with custom input extraction, it does not imply that GOLLIE is the worst-performing method; the error patterns described above are evident across nearly all of the evaluated approaches.

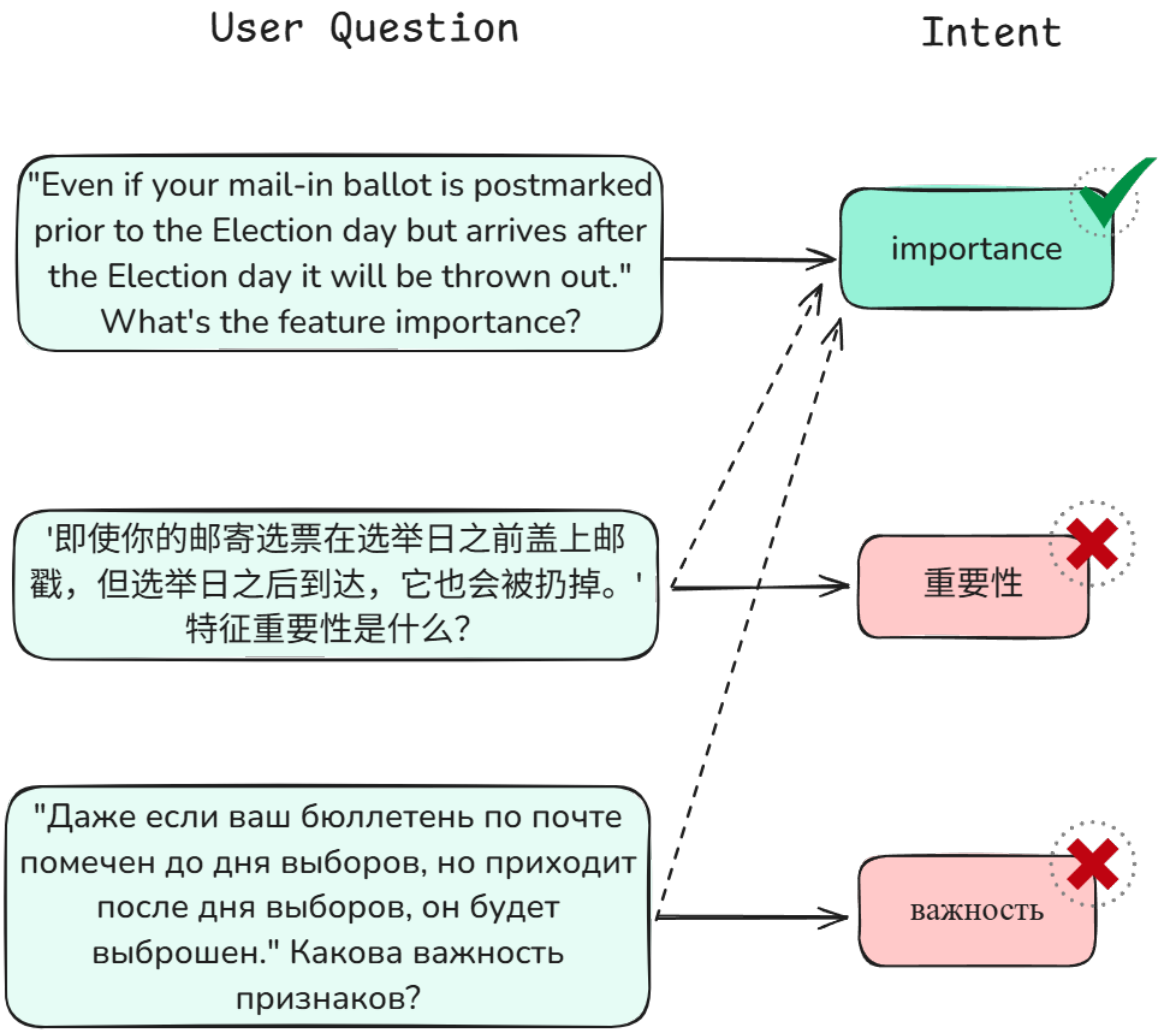

Figure 15: The same example is shown in *English*, *Chinese*, and *Russian*, along with their corresponding predicted intents. **Dashed** arrows indicate the ground-truth label, while **solid** arrows represent the predicted label.

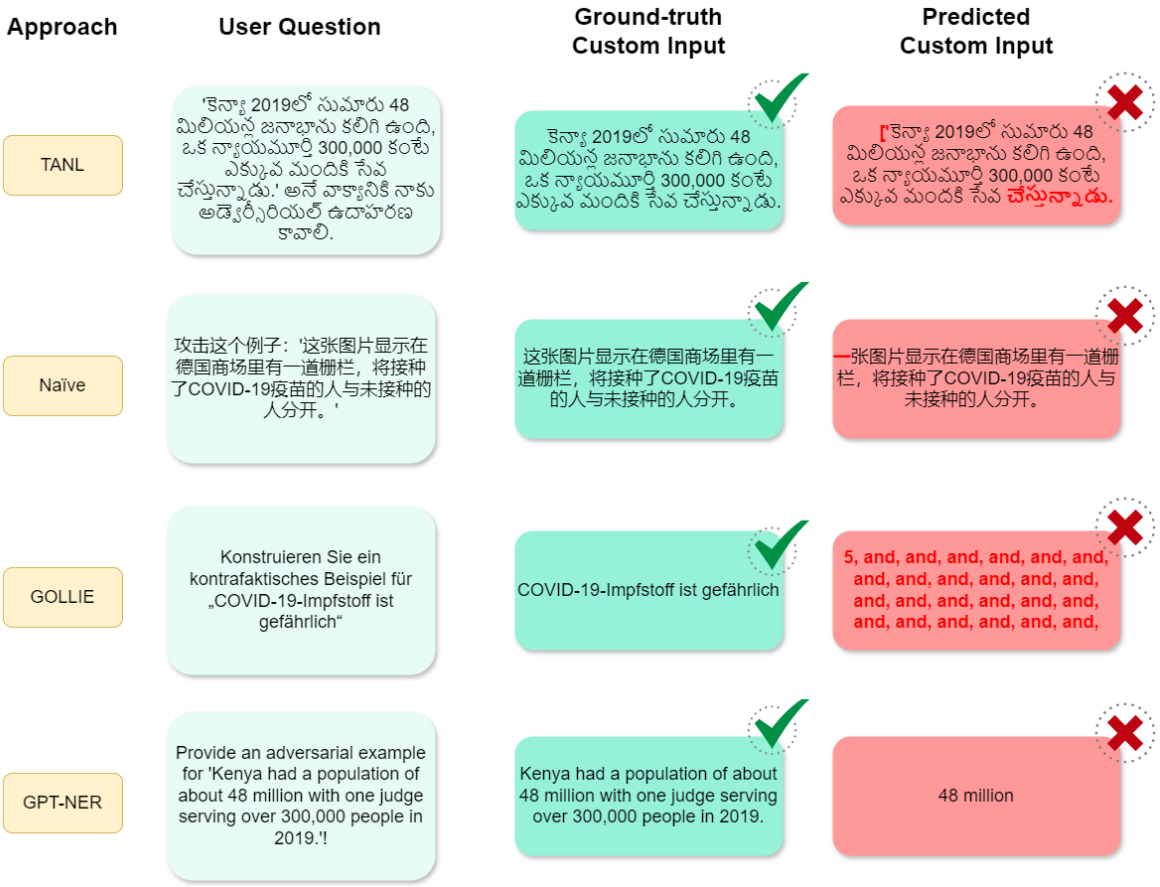

Figure 16: Four example pairs in English, German, Chinese, and Telugu, each consisting of a user question, the corresponding ground-truth custom input, the predicted custom input, and the approach used. Redundant words or those not appearing in the user question are highlighted in red.