An End-to-end Transformer-based Model for Interactive Grounded Language Understanding

Claudiu D. Hromei[1,*], Daniele Margiotta[1], Danilo Croce[1] and Roberto Basili[1]

[1]*Università degli Studi di Roma Tor Vergata, Italy*

Abstract

This paper delves into Interactive Grounded Language Understanding (IGLU) problems within the context of Human-Robot Interaction (HRI), where a robot interprets user commands about the environment. In this scenario, the robot's objective is to determine if a given command can be executed within the environment. If ambiguity or incomplete information is detected, the robot generates pertinent clarification questions. Drawing inspiration from the GrUT framework and employing a BART-based model that combines the user's utterance with the description of the environment, this study evaluates the applicability of the GrUT approach in an end-to-end Grounded QG task. The assessment of question quality is conducted through both automated metrics and human evaluation. While the results highlight the proficiency of the BART-based method in question generation, challenges arise due to dataset limitations from the IGLU competition at NeurIPS 2022. Nevertheless, this research provides valuable insights into BART's generative capabilities in the realm of HRI.

Keywords

Interactive Grounded Language Understanding, Human-Robot Interaction, Transformer-based models

1. Introduction

In recent years, Large Language Models (LLMs) have garnered substantial attention due to their remarkable performance across a wide range of NLP tasks. In addition to achieving state-of-the-art results in individual tasks, LLMs such as T5 [2], mT5 [3], IT5 [4], and FlanT5 [5] have demonstrated exceptional capabilities in solving various tasks individually and collectively through multi-task training paradigms [6]. Notably, the Decoder family, starting from GPT [7] until ChatGPT, revolutionized the field of Natural Language Processing (NLP) with their capability of solving different tasks through linguistic interaction.

These models are used not only for chatting or solving linguistic tasks but, in the Human-Robot Interaction (HRI) field, certain applications to grounded command interpretation were proposed. In recent work, GrUT [8] is proposed as an architecture for the interpretation of commands given by humans to a robot. BART [9], an Encoder-Decoder model based on Transformers, is used as the core Machine Learning model to produce the interpretation based on the command and to link real entities from the environment with the linguistic interpretation, by exploiting

NL4AI 2023: Seventh Workshop on Natural Language for Artificial Intelligence, November 6-7th, 2023, Rome, Italy [1]

*Corresponding author.

hromei@ing.uniroma2.it (C. D. Hromei); margiotta@revealsrl.it (D. Margiotta); croce@info.uniroma2.it (D. Croce); basili@info.uniroma2.it (R. Basili)

0009-0000-8204-5023 (C. D. Hromei); 0000-0001-9111-1950 (D. Croce); 0000-0002-1213-0828 (R. Basili)

CEUR Workshop Proceedings (CEUR-WS.org)

a description of the surrounding world in natural language. This process is accomplished in an end-to-end fashion: GrUT takes as input a textual description of the environment along with the user's command and then generates the linguistic interpretation of the commands, grounding mentioned entities within the context of the environment. This approach assumes that the command is complete and fully executable, something that is not always true. In fact, an interesting problem is assessing the completeness of the information and asking clarifying questions when information is not sufficient to execute the request. As an example, one such command could be "*Place the book on the chair*" in a home scenario where there are multiple chairs. An intuitive response is asking for more information about which *chair* the speaker is referring to. To fulfill this task, not only the request must be interpreted, but also the command must be assessed in the context of the environment. If feasible, it should activate the robot's planning process. However, if the feasibility is uncertain, the robotic agent is expected to generate a question to gather more information or provide assistance in carrying out the request. This Question Generation (QG) process inspired the recent *Interactive Grounded Language Understanding in a Collaborative Environment* (IGLU) challenge [10], hosted by NeurIPS 2022.

The IGLU challenge presents a unique interaction framework comprising two key agents: a (human) Architect responsible for issuing commands to a (robotic) Builder, which operates within a static environment. The Builder's central objective lies in assessing the feasibility of executing the provided actions, necessitating further inquiry if required, and generating clarifying questions from a predefined repository of potential queries. The virtual environment closely emulates the characteristics of a Minecraft world, boasting an assortment of blocks, each characterized by specific attributes such as color and position.

Typically, participants in this challenge initiate the process with an initial classification step aimed at determining the practicality of executing a given command within the given environment. Subsequently, they engage in a retrieval process to select the most suitable clarifying question for information gathering or enhancing command understanding [10]. It is noteworthy that none of the approaches employed in the challenge are fully end-to-end systems as they do not directly produce affirmative answers or seek assistance when presented with a question as an input prompt, but they all imply a two-step process.

In this context, the current paper takes inspiration from GrUT proposed in [8] for describing the environment through natural language and explores the task of end-to-end Grounded QG by adopting a similar BART-based model and a similar approach of *Textification*. The objective is to reason about a command given in input and generate a response that could be "*I can execute it.*" when no more information is needed. In this case, the model is requested to recognize that the input is complete and unambiguous. If this is not true it should generate a question, such as "*Which chair are you referring to?*", or, in the Minecraft-like world, "*Which block are you referring to?*".

The primary objectives of this work are: *i)* to evaluate the applicability of the GrUT [8] approach in an end-to-end fashion to the Grounded QG task, by adopting a natural description of the environment and directly generating the eventual question; *ii)* to assess the quality of the questions generated by the system, through both automatic measures and human judgment.

In the rest, Section 2 provides an analysis of the literature, Section 2.1 describes more deeply the IGLU task and scenario, Section 3 presents our architecture, Section 4 discusses the evaluation with an error analysis, while Section 5 derives some conclusions.

2. Related Work

The Transformer architecture [11] can be divided into two main components, each giving rise to distinct model families. The encoder, represented by BERT [12], RoBERTa [13], and DeBERTa [14], is responsible for encoding input sequences and generating meaningful representations (embeddings) using the self-attention mechanism. On the other hand, the decoder, exemplified by models like GPT [7], GPT-3 [15], and LLaMA [16], generates output sequences in an auto-regressive manner based on the input and previously generated output tokens. Additionally, there exists another family of models, the Encoder-Decoder, such as T5 [2] and BART [9], which combine the strengths of both the encoder and decoder components. These models maintain the integration of the two aforementioned blocks and are typically used in tasks like machine translation, summarization, and question-answering, where complex input understanding and transduction are required.

BART [9] is pre-trained to *denoise* the corrupted text that is given and to reconstruct its original form. The corruption during pre-training concerns masking different spans of the text, rotating the document using some pivot sentence and recognizing if any span of text is added artificially or removed. These objective functions that BART is trained to optimize allow the architecture not only to understand the text and the semantics but also to reason about the order of the sentences and to detect any missing tokens. An interesting application of such architecture is presented in GrUT [8], where the authors train BART on a collection of commands given to a robot in an automation house. The model takes in input the text of the command, such as "*Place the book on the black chair*", along with a linguistic description of the surrounding environment in order to predict a grounded interpretation of the command. Grounding, in this case, means that the model interprets the commands, i.e. generates their logical form, according to Frame Semantics [17] theory by linking each linguistic element with its unique identifier. As an example, the grounded interpretation of the above command is: Placing(Theme(b1), Goal(c1)), where b1 refers to the *book* and c1 is the *black chair*. Without the description of the world, BART could produce an interpretation only at the linguistic level, i.e. an interpretation that holds for any *book* and any *chair*.

On the other hand, the task of generating clarifying questions (QG), in an open or closed domain, is well-known in the literature, as part of the bigger area of meaningful human-robotic interaction, starting from the seminal work of Winograd [18]. Many architectures have been proposed in recent years to solve such tasks, most of the time that involve human-generated templates, including cloze type [19], rule-based [20, 21], or semi-automatic questions [22, 23, 24]. The first applications of Transformer-based models are presented in [25, 26], in order to generate questions given the text of a paragraph as input. In [25] the BERT model is trained on the inverted Stanford Question Answering Dataset (SQuAD) [27], which is a reading comprehension dataset consisting of 100,000+ questions posed by crowd workers on a set of Wikipedia articles. The model is fed with a paragraph of text concatenated with an answer and is requested to generate a question in relation to the text and the answer. On the other hand, [26] applies a GPT-2 model to the same inverted SQuAD dataset, but without any answer as input, letting the model free to generate questions, based only on the contextual text. Finally, another interesting, and more recent, application of Transformers-based architectures to the QG task is the post-

training in [28] of the original KoBART[1] model, which is a Korean version of the BART model. The post-training concerns similar denoising functions over input texts, taken from a new dataset KorQuADQG, which is entirely composed of questions, by adopting a Question Infiling technique.

All the architectures discussed so far apply a model in order to generate questions about a contextual text. Still, none of them try to interact with the user in order to gather more information. Nevertheless, in this paper, a straightforward and simple application of the BART [9] architecture is adopted for the IGLU competition, which we further explain in the next section.

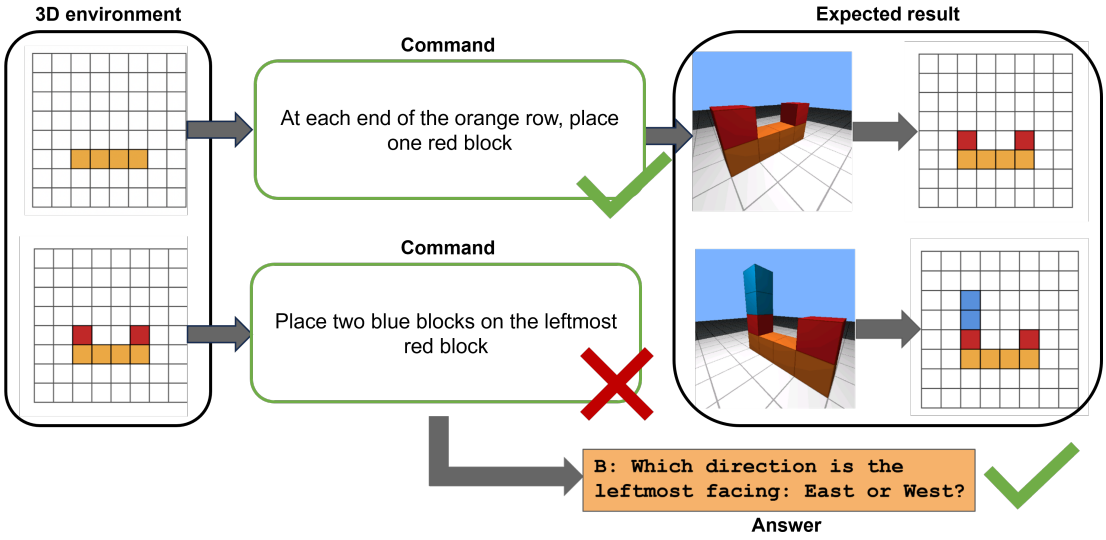

Figure 1: Taken from IGLU challenge description. *Top*: The architect's command was clear and no questions were needed, thus the Builder can execute it. *Bottom*: The word '*leftmost*' in the Command is ambiguous, so the Builder asks a clarifying question.

2.1. The IGLU competition

The IGLU [10] challenge was organized to facilitate research in the area of Human-Robot Interaction for collaboration through natural language. The aim is to build interactive agents that learn to solve a task while being provided with grounded natural language instructions in a collaborative environment. The Robotic Agent should leverage not only the given instruction but the information about the environment as well, in order to collaborate with the Human Agent. Interactive agents are those who can follow instructions in natural language and ask for clarification when needed. The IGLU setup involves collaboration between human and AI agents with physical bodies, who must use language to achieve a common objective in a voxel-based environment. In this setup, the "Architect" is the Human Agent, who is presented with a three-dimensional arrangement of colored blocks and must communicate instructions to

[1] https://github.com/SKT-AI/KoBART

the "Builder" agent, which can manipulate the blocks and interact with the environment. If the instructions are unclear, the Builder can ask for clarification from the Architect. IGLU is related to two primary areas of AI research: Natural Language Understanding and Generation (NLU/G) and Reinforcement Learning (RL).

This paper presents an architecture proposed to solve the tasks in the NLU/G area, whose objective is to identify when and what clarifying questions to ask, with a more in-depth focus on the Grounded Question Generation task. In IGLU, the Builder receives instructions from the Architect (such as "*Place two blue blocks on the leftmost red block.*" as in the bottom flow of Figure 1), and it must determine if the given information is adequate and complete to carry out the task or if more details are required. To obtain additional information, the Builder may ask questions like "*Which direction is the leftmost facing: east or west?*" to resolve the ambiguity and ensure that the task is completed accurately. The two tasks are addressed as: *i*) a classification problem (to ask or not to ask) and *ii*) a ranking problem (what to ask) to select the best question among the list of all target questions, provided during the challenge, by sorting it. It's important to note that this natural language processing and generation (NLP/G) task is separate from learning how to interact with the 3D environment, which will not be addressed here.

Remarkably, the best-scoring system in the competition[2] relies on multiple BERT models, each receiving as input both the user command and a structured representation of the virtual environment. These models independently make binary predictions regarding the presence or absence of certain information, using the context of the world and the user's command. Then, the BM25 model is applied to ranking a closed set of predefined questions in order to produce the most relevant. Although this architecture is similar to our system, the description of the world we provide to the model is simpler. Furthermore, our primary objective is to evaluate the feasibility of a simple end-to-end system capable of not just generating pertinent questions but also avoiding the complexity of an overly complex process involving question classification and ranking within a predetermined set of questions.

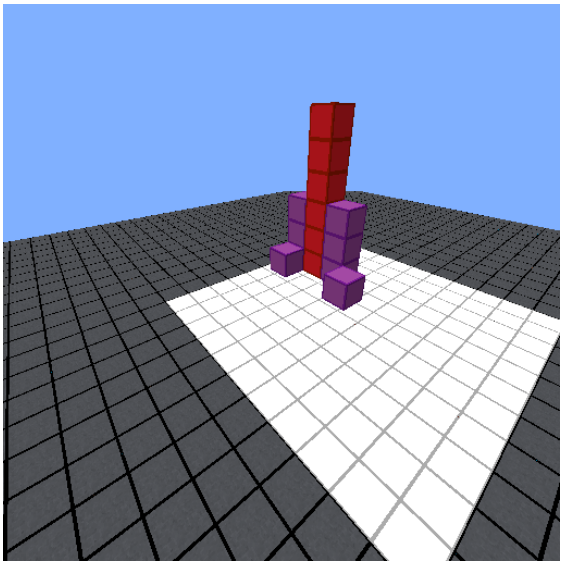

Figure 2: An example of visual rendering of the environment, where the Instruction given by the Human is "*Break the green blocks*" and the expected answer is "*There are no green blocks, which blocks should I break?*".

[2]As resulting from the current page: https://www.aicrowd.com/challenges/neurips-2022-iglu-challenge/problems/neurips-2022-iglu-challenge-nlp-task/leaderboards

3. Generating clarifying questions through an End-to-End Approach

Our approach to generating the relevant questions is by leveraging natural language descriptions of maps, inspired by GrUT [8]. GrUT is an approach that emphasizes grounding natural language understanding in textual descriptions via Transformers. We adopt a similar approach by combining textual descriptions of maps with the natural language commands. To construct a textual description of a map, we begin by providing information about the blocks, their positions, and their colors. For example, in the case of the map shown in Figure 2, the description would be structured as follows:

"*There are no blue blocks, no yellow blocks, no green blocks, no orange blocks, eight purple blocks, four of which are on the ground, six red blocks, one of which is on the ground.*" (1)

This structured description serves as the foundation for our approach and the context for the BART model. Moreover, the details about the color of the blocks on the ground could be leveraged in ambiguous situations. Subsequently, we combine this textual Map Description (MD) with the natural language command, such as "*Break the green blocks.*". These two components together form the input for our system based on BART. The model is capable of understanding the textual map description and the command and generating a corresponding question when more information is needed. In the case of our example, the output question could be: "*There are no green blocks, which blocks should I break?*". One crucial aspect to highlight is that BART is context-aware through the MD, which means it can generate questions about missing information or ambiguities in the given command. Consequently, the MD functions as both a descriptor of the Minecraft-like environment and a surrogate for visual information.

In the context of the IGLU challenge, the pivotal component at play is the BART model, which assumes a central role in determining both when to pose questions (referred to as "to ask or not to ask") and what specific question to ask (known as "what to ask"). Within this challenge, the task involves the selection of the most appropriate question from a predefined set of questions based on the given command. To facilitate this, the answer generated by the BART model undergoes a subsequent evaluation by our ranker component. This ranker component engages in a similarity assessment by comparing the generated answer against every question within the predefined question set, ultimately identifying and selecting the most relevant question.

The overall workflow is illustrated in Figure 3. The command given to the Builder is concatenated with a description of the world and then fed in input to the BART model. The answer generated by the BART model serves as a classification marker, indicating whether to ask questions to the Architect. When the BART model generates the response "*I can execute it.*" it signifies the completion of the process. However, if the generated response differs from this, it signals a recognition that additional information is required. Since the output question must be selected from a predetermined set, we employ a ranking approach based on Cosine Similarity (referred to as Cos Sim in Fig. 3). This method calculates the similarity between the embedding of the candidate question and the embeddings of all the questions within the predefined set ($qbank$ in Fig. 3). For this purpose, we utilize a Sentence-BERT [29] model

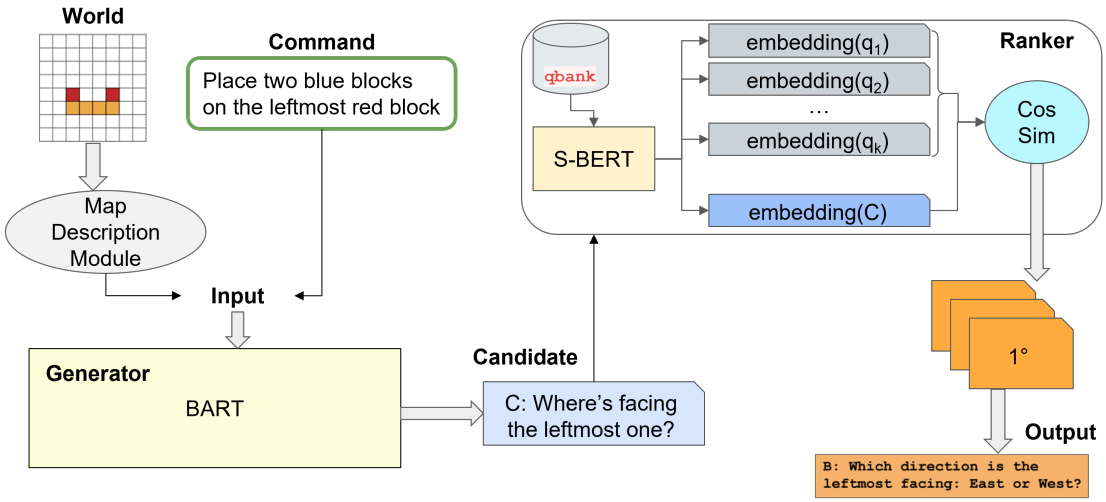

Figure 3: The workflow of our Architecture based on BART for generating and selecting the most suitable answer to the input command based on the current state of the world.

(denoted as S-BERT in Fig. 3)[3] for generating embeddings of sentences, as it was demonstrated in [29] to drastically reduce the time of computation of similar sentences. Finally, a ranking is conducted in non-ascending order based on the Cos Sim score, with the top-ranked question being selected as the most suitable one.

In this study, our primary emphasis lies on the Generator component. Our objective is to investigate the Transformer-based model's capacity to determine whether a question should be posed in situations where the command's execution feasibility is uncertain. Additionally, we aim to assess the quality of questions generated when the command cannot be executed, as reported in the next section.

4. Experimental Evaluation

In this section, we will assess BART's proficiency in generating contextually grounded questions, which serves as the main evidence of its ability to comprehend instructions and discern missing information that can be reformulated into a question. This evaluation process yields valuable insights into the model's cognitive understanding and its competence in generating questions grounded in context. We will address the following key aspects:

- **Quality of Generated Answers**: We will scrutinize the quality of the answers produced by BART.
- **In-Depth Error Analysis**: We will perform a comprehensive error analysis to gain insights into the limitations of the BART model. This examination will delve into instances where the model encounters challenges and provides a deeper understanding of its performance boundaries.

[3]The model here used is a variant of DistilBERT from https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v2

- **End-to-End Question-Answer Generation**: We will explore whether an end-to-end system can successfully generate valid answers. It's important to note that while two sentences, denoted as A and A', may exhibit variations in their BLUE scores, they can still be pertinent to our task in terms of semantic coherence. Thus, we will conduct a thorough manual analysis of the generated questions.

The model is based on BART-base, implemented using the Huggingface framework[4]. The model is trained by providing in input the concatenation of the environment description with the user utterance, while in the output the expected response from the robot, that is the artificial string "*I can execute it.*" in the case no other information is required, while the actual question when additional information is needed. The model is fine-tuned according to the parameters summarized in Table 1.

Table 1
Summarization of the parameters of the BART model.

Parameter Name	Value
Optimizer	AdamW
Early_stopping_delta	$1 \cdot 10^{-3}$
Early_stopping_metric	eval_loss
Batch_size	16
Early_stopping_patience	2
Scheduler	linear_with_warmup
Warmup Ratio	0.1
Max_length	128
Learning rate	$3 \cdot 10^{-5}$
Epochs	50 (max)
Model Size	base

4.1. Evaluating the overall process.

First, we evaluated our model according to the set adopted in the IGLU challenge [10]. Initially, the system's performance is assessed in terms of its capability to accurately determine whether a question is necessary or not. This binary classification task is specifically evaluated using the average F1 measure of the two classes. Notice that, when no question is needed, the model is expected to produce the artificial string "*I can execute it*". In the second step, the system's task is to rank all possible questions in the repository so that the correct one appears first. To measure its performance in this task, the Mean Reciprocal Rank is used, defined as:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{2}$$

Where Q is the set of queries (in our case is the set of test commands) and $rank_i$ is the position of the correct answer (question).

[4] https://huggingface.co/facebook/bart-base

The training dataset[5] utilized is provided by the IGLU competition and contains $6,828$ instructions, each with an associated environment description consisting of a list of blocks identified by their color and (x, y, z) coordinates. Unfortunately, the test set is not statistically representative since it is not made available anymore. In this paper, we relied on the provided dataset, by dividing it into training, validation, and testing sets with a ratio of 80/10/10.

In the initial task, the system achieves an F1 score of 74.36% on a test dataset comprising 683 examples. Notably, in 90 instances where a question was expected, the system accurately generates questions in 45 of those cases (with 45 false negatives where it fails to propose them). Simultaneously, it introduces 29 false positives, which are instances where the system asks questions even when it shouldn't.

A direct comparison with the conference participants' systems is not possible. The best-performing systems achieved the F1 scores of 76.6%, 76.1%, and 75.4% for the top three systems in the competition on the official test set. It's worth noting that our result on our local test, even though is not aligned with the online test set, is comparable with those of the best-performing systems and is particularly interesting considering the end-to-end nature of the proposed model.

In the second subtask, the Ranker whose architecture is summarized in Figure 3 achieves a 0.2311 MRR on our local test set, which means that on average our method ranked the relevant question fifth. The result is really impressive considering that the entire collection is made of 835 questions.

4.2. Evaluating the Question Generation Process

To assess the quality of the end-to-end Transformer-based system, independently of any bias introduced by the need to select the correct response from a repository, we evaluated the quality of the generated text. Specifically, when the system was required to produce a sentence, we measured the distance in terms of the Bleu Score between the correct sentence and the generated one. We obtained BLEU1 = 0.255, BLEU2 = 0.147, BLEU3 = 0.083, and BLEU4 = 0.061. It is evident that as the number of expected n-grams increases, the result decreases, reaching a low BLEU4 score of 0.061.

However, this quantitative measure, originally designed for evaluating tasks like Machine Translation, can be overly restrictive. For instance, in response to a command such as "*Destroy all the red blocks*", a system that answers, "*The map contains no red blocks*" may share no common terms with a response like "*I don't see any elements of the requested color*", reaching a BLEU score of 0.

As a result, we conducted a qualitative assessment using the test set comprising 683 examples. From this set, we selected a subset of 47 examples where the system generated a request. We manually examined whether the generated sentence, although different from the expected one, contained a question that was useful in resolving the ambiguity or limitations introduced by the user's request. In such cases, the generated sentence was considered correct; otherwise, it was deemed incorrect. This allowed us to calculate a Relaxed-Accuracy, computed as the percentage of examples considered correct.

[5]You can downloaded it from https://github.com/microsoft/iglu-datasets with MIT license

Furthermore, for the error analysis conducted here, we identified eight categories of "missing" information in the command that the GS annotated question is addressing. These categories include aspects such as the Number or the Color of the blocks to be placed or removed, the Direction in which a line of blocks must be placed, or Block Missing when the command refers to a specific color block that does not exist in the environment. Table 2 provides a description of all the identified categories, along with an example question for each.

Table 2
The categories of "missing" information in the command identified in this work. Each category is described by a question example. A "Relaxed" Accuracy is computed for each category on the test set.

Category	Description with example	Relaxed-Acc
Block	"*Which specific block do you mean ?*"	38.46%
Vertical-Horizontal	"*How are they arranged? Vertical or horizontal?*"	50%
Number	"*How many blocks? Or how long?*"	57.14%
Square	"*Where should I place the blocks?*"	77.14%
Color	"*Which color should the block be?*"	50%
Direction	"*In which direction? What is the orientation?*"	22.23%
Block Missing	"*There is no red block*"	58.34%
Complete	"*I can execute it.*"	97.81%
Overall	-	92.54%

As you can see from Table 2, the reported Relaxed-Accuracy is quite low in most categories, with Direction achieving 22.23%. This is mainly due to two emerging phenomena: *i)* the majority of the commands are complete and our BART system correctly generates the sentence "*I can execute it.*" meaning no more questions are needed (the Complete category); *ii)* BART is not able to generate the exact question annotated by the Gold Standard. Although we achieved some low results on specific categories, the Overall Relaxed-Accuracy reaches 92.54%: an interesting result considering the nature of the task and the limited dataset. As an example, for the command "*In the center place three orange blocks horizontally*" BART generates the question "*Where in the center do I place the orange blocks?*" when the GS annotation is "*Where in the center should the three orange blocks be placed?*", which are basically two equivalent questions. This phenomenon counts for half of the errors of BART. The other half of the errors occur when the GS annotates a command as Complete but BART still generates a question. For example, for the command "*In the center place three green blocks horizontally*" BART generates the question "*Which direction should the horizontal row span?*", as the environment is empty and contains no other blocks. The model successfully recognized that it could not infer the direction in which the blocks should be placed and asked a question. This "error" regarding the empty map occurs many times. On the other hand, there are instances where the GS annotates the command with a question and BART generates "*I can execute it.*" meaning that no other information is needed. In fact, for the command "*Place a green block at the southeast corner. then place a green block on every side of it.*" the GS produces the question "*Where in the Southeast corner?*" while BART successfully recognizes that there is no need for more questions as the "*Southeast corner*" is unambiguous: it means the very far position where South and East meet. Another example is the command "*Build a column of two green blocks on top of the yellow block. Break the yellow block and replace it with a green block.*" coupled by the GS with the question "*Where do I place*

the green block?". In this case, the command is clear: there is only one yellow block in the environment and the Builder is requested to stack a column of two green blocks on top of it, then destroy the yellow block, and finally replace it with a green one, meaning to put a green block in the same place of the previous yellow one.

4.3. Human Machine Comparison in the Generation

This section will delve into a comparative analysis, where the model will be assessed against the GS. This evaluation involves human judgment, where individuals will decide which response they prefer and which one exhibits superior English syntax and semantics. This assessment will provide a holistic view of the model's linguistic prowess and its effectiveness in generating contextually accurate questions.

In our evaluation process, we carefully assessed the test dataset by assigning scores based on two crucial aspects: *Utility* and *Fluency*. These two metrics were instrumental in gauging the performance of our BART model. *Utility*, the first parameter, pertains to the model's ability to generate a coherent question that aligns seamlessly with the missing information in the given command. In essence, it measures the model's proficiency in asking the right question to elicit the required information. *Fluency*, the second dimension, delves into the language fluency of the generated questions in English. This aspect focuses on the model's capability to craft sentences that flow naturally and grammatically, regardless of the correctness of the response provided. Table 3 describes the scores for the two aspects here evaluated with a brief description for each score.

Table 3
Scores for the Utility and Fluency metrics from 1 to 5, where both need to be maximized.

Utility	Fluency
1 - Completely inconsistent	*1 - Not English*
2 - Incorrect question	*2 - Random English words*
3 - Awareness of the task, but the question is not relevant	*3 - Understandable but critical errors*
4 - Asks at least one missing information (color, blocks, etc.)	*4 - Light grammatical errors*
5 - Perfect	*5 - Perfect*

To ensure objectivity and impartiality in our evaluation process, we enlisted the assistance of an evaluator who was not part of the development team. In order to avoid any bias, it was crucial that this external evaluator remained unaware of whether the sentences they were assessing had originated from the GS or BART. This unbiased assessment was conducted on a test dataset comprising 230 sentences, in which the question of the GS was different than the question generated by BART. These examples were equally divided between BART and the GS and coupled with the visual representation of the environment (as in Figure 2). It helped us derive meaningful insights into the semantic and syntactic capabilities of the model, shedding light on its respective strengths and areas for improvement.

The Gold Standard annotation attains a Utility score of 3.10, implying that the IGLU competition's annotated data is generally not so accurate, occasionally lacking comprehensive information and posing some misleading questions. Conversely, our BART model achieves a superior score of 3.95, reflecting its ability to generate more relevant questions, that address

important missing information in the command, though it is not without occasional inaccuracies. In terms of Fluency scores, both models perform very well, with no significant disparities observed: 4.84 for the Gold Standard annotation and 4.97 for the BART model.

5. Conclusions

In this paper, a method based on BART has been introduced, which serves the purpose of determining whether received commands are complete or necessitate the generation of a clarification question. Instead of a misleading BLEU score, the method's evaluation encompassed both a Relaxed-Accuracy and the *Utility* and *Fluency* scores for assessing the syntax and semantics of the generated questions. Intriguingly, the results indicate that indeed is possible to apply such an end-to-end architecture to the Interactive Grounded Language Understanding task.

However, this commendable achievement comes with a caveat. The dataset employed in this study, provided for the IGLU competition at NeurIPS 2022, lacked meticulous construction, and consistency, and frequently led astray with misleading information. Consequently, training robust models without succumbing to overfitting challenges proved to be a daunting task. Nevertheless, despite the dataset's limitations, this work provides valuable insights into BART's generative capabilities. Moreover, the indirect comparison of our model on the local test set showed an alignment in performance with the best-scoring systems of the competition.

Looking forward, there are exciting prospects for extending this research, after having consolidated the size for scaling of our human evaluation from Section 4.3. One promising avenue is the exploration of Large Language Models (LLMs) such as LLaMA, as in `ExtremITA` [6]. Additionally, the integration of models that incorporate both visual and textual signals holds great potential, paving the way for more sophisticated and context-aware Grounded Question Generation techniques in the future.

Acknowledgments

We would like to thank the "Istituto di Analisi dei Sistemi ed Informatica - Antonio Ruberti" (IASI) for supporting the experimentations. Claudiu Daniel Hromei is a Ph.D. student enrolled in the National Ph.D. in Artificial Intelligence, XXXVII cycle, course on *Health and life sciences*, organized by the Università Campus Bio-Medico di Roma. We acknowledge financial support from the PNRR MUR project PE0000013-FAIR.

References

[1] E. Bassignana, D. Brunato, M. Polignano, A. Ramponi, Preface to the seventh workshop on natural language for artificial intelligence (nl4ai), in: Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023), 2023.

[2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67.

[3] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the NAACL-HLT 2021, Online, June 6-11, 2021, ACL, 2021, pp. 483–498.

[4] G. Sarti, M. Nissim, IT5: large-scale text-to-text pretraining for italian language understanding and generation, CoRR abs/2203.03759 (2022).

[5] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Y. Zhao, Y. Huang, A. M. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling instruction-finetuned language models, CoRR abs/2210.11416 (2022).

[6] C. D. Hromei, D. Croce, V. Basile, R. Basili, ExtremITA at EVALITA 2023: Multi-Task Sustainable Scaling to Large Language Models at its Extreme, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[7] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).

[8] C. D. Hromei, D. Croce, R. Basili, Grounding end-to-end architectures for semantic role labeling in human robot interaction, in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), Udine, November 30th, 2022, volume 3287, CEUR-WS.org, 2022.

[9] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, CoRR abs/1910.13461 (2019).

[10] J. Kiseleva, A. Skrynnik, A. Zholus, S. Mohanty, N. Arabzadeh, M.-A. Côté, M. Aliannejadi, M. Teruel, Z. Li, M. Burtsev, M. ter Hoeve, Z. Volovikova, A. Panov, Y. Sun, K. Srinet, A. Szlam, A. Awadallah, Iglu 2022: Interactive grounded language understanding in a collaborative environment at neurips 2022, 2022. arXiv:2205.13771.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017).

[12] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.),

Proceedings of the NAACL 2019, 2019, pp. 4171–4186.
[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR (2019).
[14] P. He, X. Liu, J. Gao, W. Chen, Deberta: decoding-enhanced bert with disentangled attention, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021.
[15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020).
[16] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.
[17] C. J. Fillmore, Frames and the semantics of understanding, Quaderni di Semantica 6 (1985) 222–254.
[18] T. Winograd, Procedures as a representation for data in a computer program for understanding natural language, Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC, 1971.
[19] K. M. . K. Hermann, T. . Grefenstette, E. . Espeholt, L. . Kay, W. . Suleyman, M. ;, P. Blunsom, Teaching machines to read and comprehend, in: Advances in neural information processing systems, 1693-1701, 2015.
[20] R. Mitkov, L. A. Ha, Computer-aided generation of multiple-choice tests, in: Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing, 17–22, 2003.
[21] V. . W. Rus, B. . Piwek, P. . Lintean, M. . Stoyanchev, S. ;, C. Moldovan, The first question generation shared task evaluation challenge, 2010.
[22] G. Alvaro, J. Alvaro, A linked data movie quiz: the answers are out there, 2010.
[23] G. A. . C. Rey, I. . Alexopoulos, P. . Damljanovic, D. . Damova, M. . Li, N. ;, V. Devedzic, Semi-automatic generation of quizzes and learning artifacts from linked data, 2012.
[24] D. Liu, C. Lin, Sherlock: a semi-automatic quiz generation system using linked data, in: International Semantic Web Conference (Posters & Demos), 9–12. Citeseer, 2014.
[25] K. Kriangchaivech, A. Wangperawong, Question generation by transformers, CoRR abs/1909.05017 (2019).
[26] L. E. Lopez, D. K. Cruz, J. C. B. Cruz, C. Cheng, Transformer-based end-to-end question generation, CoRR abs/2005.01107 (2020).
[27] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ questions for machine comprehension of text, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.
[28] G.-M. Park, S.-E. Hong, S.-B. Park, Post-training with interrogative sentences for enhancing BART-based Korean question generator, in: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of ACL and the 12th International Joint Conference on NLP, 2022.
[29] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, CoRR abs/1908.10084 (2019).