Analysing the Communicative Effectiveness of Large Language Models under the Rational Speech Act Framework

Mingyue Jian

Master of Science
Cognitive Science
School of Informatics
University of Edinburgh
2024

Abstract

Are large language models (LLMs) capable of behaving like pragmatic speakers? While most current studies focus on evaluating LLMs' ability to understand the implicature of non-literal content, there has been limited investigation into their ability to generate such content. We present an evaluation for the communicative effectiveness of LLMs using the Rational Speech Act framework, which is a probabilistic method for understanding pragmatic reasoning in the context of human communication. Our research uses this framework to evaluate the communicative effectiveness of large language models by comparing the probabilities assigned by the LLM (a variant of Llama3-8B-Instruct) to certain utterances against the probabilities assigned by an RSA model. We find that the LLM's output is positively correlated to that from an RSA model ($PCC = 0.203$, $SRCC = 0.342$).

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Mingyue Jian*)

Acknowledgements

This project marks the end of my academic journey as a taught student. When I switched from a German literature degree four years ago, I never imagined I'd make it this far in the tech world. Alongside the skills I've gained from my degrees, I've also received incredible love and support from so many people along the way.

First and foremost, I want to express my sincere gratitude to my supervisor, Dr. Siddharth Narayanaswamy, for coming up with this cool project and for his patient, insightful guidance and support throughout the process.

I also want to extend my deepest thanks to my coursemates and friends for being with me throughout the year. This journey wouldn't have been the same without you, and I'll always cherish the days — whether they were happy or challenging — in Appleton Tower, the library, and on the meadows with you. A special thank you to my besties, Joe and Sophie, for all the encouragement, love, and prayers.

Finally, I'd like to thank my parents, my older brother, and my cousins for their endless love, support, and belief in me. Even though we haven't seen much of each other over the past four years, especially during the pandemic, your unwavering trust has kept me going.

Table of Contents

Chapter 1

Introduction

1.1 Background: Pragmatic reasoning in human communication

Humans are typically pragmatic agents in communication. Imagine you're in a room with three pieces of furniture: a small red desk, a small yellow desk, and a large red chair. If your roommate refers to one of these items as "the red one", you would first eliminate the yellow desk as a possibility, since it's not red. Then, you'd turn your attention to the two remaining red objects. Given that both are red, you might reconsider, thinking that if your roommate intended to refer to the chair, they would have simply said "the chair", as it is a more distinctive feature among the three objects in the room. Ultimately, you would likely interpret your roommate's reference as pointing to the small red desk. This process of back-and-forth thinking to infer another person's intention is known as pragmatic reasoning in our daily lives.

Humans use an intensional strategy in their "theory of mind", a concept in cognitive science that involves understanding and predicting the beliefs, desires, and intentions of others by treating them as rational agents and considering their place in the world and their purpose [15, 40].

In communication, we tend to use the intensional strategy to enhance efficiency. One of the most influential frameworks for modelling pragmatic reasoning is the Rational Speech Act (RSA) framework [21, 14]. It formalises how speakers and listeners use context, shared knowledge, and probabilistic reasoning to communicate effectively. This framework operates language understanding as a recursive process, where both speakers and listeners in a conversation behave rationally to reason each other's intention.

Speakers choose their words to optimally convey their message, anticipating that listeners will interpret these words in the most likely context. Meanwhile, listeners use the context and their linguistic knowledge to infer the speakers' intended meaning.

In the earlier example, when your roommate refers to "the red one", you first exclude the yellow desk because it does not match the description—acting as a "literal listener" within the RSA framework. This decision is guided by a "meaning function", which evaluates whether the literal interpretation aligns with the speaker's words. Meanwhile, the pragmatic speaker, your roommate, strategically selects their words, anticipating how you, as the listener, will interpret the message. As a pragmatic listener, you then refine your interpretation, considering both the explicit message and the surrounding context, ultimately inferring your roommate's intended meaning. This dynamic interplay between speaker and listener, with each considering the other's perspective, is central to effective communication as modelled by the RSA framework.

1.2 Motivation: Are LLMs pragmatic speakers?

One of the enduring challenges in Natural Language Processing is creating models that can accurately reflect human pragmatic behaviour. With the emergence of large language models (LLMs) [5, 1, 11, 29, 31, 49, 50], a key question arises: can these models exhibit pragmatic reasoning similar to humans? Determining the extent to which LLMs can engage in this type of reasoning is vital for establishing their reliability, and it also has the potential to shed light on human pragmatic abilities [27]. While much research has focused on evaluating LLMs' pragmatic abilities as listeners, specifically how well they understand non-literal input [27, 34, 42, 47, 39], there has been comparatively less investigation into their capabilities as speakers. The question remains: can LLMs effectively use context to generate output that conveys more than just the literal meaning? Exploring this aspect is essential for gaining deeper insight into the internal mechanisms of LLMs during generation, ultimately enhancing their reliability and trustworthiness.

1.3 Research question

With the constructed RSA framework for modelling human pragmatic reasoning in communication, we investigate whether LLMs employ similar reasoning when generating text and, if so, to what extent they use the same framework for pragmatic communication.

1.4 Contributions

To achieve this, we employ a reference game task, asking the experimented models to refer to a target object. We compare the output of a vanilla LLM with that of an RSA-based model, which serves as the benchmark for pragmatic generation, to assess whether their output sequences are correlated. Our results indicate that our vanilla LLM model (a variant of Llama3-8B-Instruct [56, 50]) shows a positive correlation with the two RSA models that utilise different meaning functions in the context of the reference game task. The contributions of the research includes:

- We propose a pipeline for the pragmatic generation evaluation for LLMs using the RSA framework at inference. The pipeline is model-agnostic, which could be applied to different models for comprehensive analysis.

- We present two methods to construct the utterance space of the RSA framework. The utterance space are sequences that the model could generate, given its intention. The two methods are: sampling from top-k sequences by a beam search algorithm, and constructing pragmatic and literal utterances using logical rules. The development of the utterance space is essential to the RSA framework, since both agents in communication would reason over all the possible utterances the other agent could have said, given their intention.

- Additionally, we present two methods of constructing the meaning function within the RSA framework. The first is a prompt-based one, which leverages the in-context learning ability of the LLM, and returns the meaning function decision from the LLM. The other is a rule-based meaning function, which is a specialised meaning function to our reference game task, which has proved to have better performance in the meaning function task compared to the prompt-based one in our evaluation.

- Our research provides insights into the pragmatic generation capabilities of the LLaMA3 variant model and examines how its correlation with the RSA model varies depending on different types of utterances. Additionally, we analyse how the alignment is affected when the RSA model is applied with different meaning functions.

Chapter 2

Related Work

2.1 Large Language Models (LLMs)

2.1.1 What are they

A language model is a probabilistic framework that generates a probability distribution over potential tokens based on the sequence of preceding tokens [4]. For instance, given a sequence of k tokens $(x_1, \ldots, x_k)$, the model computes the probability for each possible next token as follows:

$$p(x_{k+1}|x_1, \ldots, x_k) = \sigma(f(x_1, \ldots, x_k; \Theta)), \tag{2.1}$$

where f is a function parameterised by learned weights Θ, and $\sigma(:)$ is the softmax function, which transforms the output of f into a probability distribution.

Large language models (LLMs) mainly refer to transformer-based [52] language models that are pre-trained on massive text data and characterised by a substantial parameter count. This extensive training endows LLMs with exceptional learning capabilities, enabling them to exhibit superior performance across various downstream tasks [38].

2.1.2 In-context learning ability of LLMs

A notable feature of LLMs is their in-context learning (ICL) capability, as identified in GPT-3 [5]. This ability enables the model to generalise and perform novel, previously unseen tasks when provided with one or more examples within the prompt during inference. In contrast to the traditional method of fine-tuning language models through supervised learning, ICL offers a training-free learning paradigm, significantly reducing the computational resources and time required for completing specific tasks [16].

Despite the flexibility and efficiency of ICL, its performance is highly sensitive and susceptible to biases stemming from prompt design [53]. Factors influencing this sensitivity include the selection of examples, the structure of the prompt template, and even the sequence in which examples are presented. The underlying mechanism of how LLMs utilise ICL remains largely opaque. Consequently, identifying an optimal prompt necessitates considerable human effort to experiment with various prompt designs to maximise LLM performance. This iterative process of crafting and refining prompts to achieve desired outputs is known as prompt engineering.

2.1.3 Prompt engineering

Prompt engineering has emerged as a prominent research focus in recent years [32, 33, 30, 43, 9], especially following the remarkable ICL capabilities demonstrated by several LLMs [5, 1, 11, 29, 31, 49, 50].

Few-shot prompting is one of the more lightweight approaches for prompt engineering [5]. It focuses on the textual design of the prompt without necessitating additional resources. Few-shot prompting involves presenting the model with a limited number of input-output examples to induce an understanding of a given task. This technique not only helps the model generalise the mapping between input and output but is also effective for generating results in a specific format.

2.1.4 Pragmatic reasoning ability of LLMs

Mahowald et al. delineate the linguistic and cognitive capabilities of LLMs [35]. They define **formal linguistic competence** as the model's expertise in understanding and applying language rules, encompassing skills such as syntax, lexical semantics, phonology, and morphology. In contrast, they define **functional linguistic competence** as an advanced stage, emphasising the ability of LLMs to appropriately select and use language in real-world contexts. This includes skills such as formal reasoning, which involves logical and mathematical reasoning; world knowledge, encompassing common sense, concepts, and factual information processing; and social reasoning, which involves pragmatics and theory of mind.

Their findings indicate that with the expansion of training data and model capacity, LLMs have largely mastered formal linguistic competence in English. However, nonaugmented LLMs still fall short in functional linguistic competence [45, 44], specifically social reasoning [8].

2.1.5 Research in the field about LLM's pragmatic reasoning ability

The field of explainable AI (XAI) seeks to determine whether the increasing scale of LLMs endows them with similar functional linguistic capabilities. Here, we focus on the pragmatic reasoning abilities of LLMs.

Most researchers have evaluated LLMs' pragmatic understanding in terms of their ability to interpret non-literal linguistic inputs. These evaluations predominantly focus on LLMs' interpretive abilities (or their performance as listeners), assessing whether they can comprehend the underlying implications of the input [27, 34, 42, 47, 39]. Most of this research concludes the limitations of LLMs' pragmatic abilities by comparing the output from the LLM to human output within pragmatic scenarios [48, 27, 19].

However, there is limited research on evaluating LLMs' generative abilities (or their performance as speakers), particularly concerning the pragmatics of their generated outputs.

2.2 Measurement of pragmatics

Pragmatics encompasses various dimensions, including physical, linguistic, social, and epistemic aspects [24]. Our research concentrates on linguistic pragmatics, an essential component of communicative competence [6]. Diverse methodologies exist for assessing linguistic pragmatics across different disciplines.

2.2.1 Linguistic approach

In linguistics, pragmatics refers to the notion that utterances convey meaning beyond their literal semantics [23, 26]. Grice proposed a framework for measuring pragmatic reasoning through various conversational maxims: truthfulness, relevance, informativeness, and perspicuity [21]. These maxims operate under a cooperative principle between the speaker and the listener to lead to implicatures. Although this framework is challenging to formalise due to the complexity of natural language and the absence of definitive interpretations, making it difficult to obtain quantitative results, it serves as a foundational element for future proposed frameworks.

One attempt to get pragmatic implicatures using linguistic theories is a grammatical model [10]. The core idea of the grammatical model is that pragmatic implicatures are derived from the grammar of a sentence by using logical principles to determine which alternative utterances can be excluded or included based on a speaker's assertion and the set of non-contradictory alternatives.

This model is constrained by its restrictive method of formulating alternative utterances, primarily through the substitution of scalar items (such as quantifiers, numerals, and modals) in embedded positions [46]. Additionally, the discrete selection process may result in scenarios where multiple referents remain after the exclusion process, rendering the listener unable to discern the intended meaning among the remaining alternatives.

2.2.2 Statistical approach

Researchers have been using statistical models to study pragmatic reasoning in communication [57]. These models considering both the speaker and listener as cooperative agents, and derive implicatures through iterative statistical inferences about each other's intentions. On a very high level, these models iteratively refine a heterogeneous relation between utterances U and meanings M, such that the relations begin being purely literal, and is refined using pragmatic reasoning. Here, we introduce the Iterated Best Response (IBR) model [18], which uses a proper relation:

$$U \times M \to Bool. \tag{2.2}$$

As well as the Rational Speech Act (RSA) framework [21], which approximates a relation:

$$U \times M \to [0,1]. \tag{2.3}$$

The listener and the speaker can then use the relation to choose an appropriate meaning for each utterance, and vice verse.

2.2.2.1 IBR

The IBR model treats each interaction as a level, where the speaker expresses their intention, and the listener comprehends it. The model quantifies pragmatic implicature by iteratively calculating the probability of the other agent's intention based on the output from the previous level of that agent.

Level-0 agents are unstrategic and consider only the literal truthfulness of the utterance. In contrast, a level-$(t+1)$ agent assumes the other agent behaves as a level-t player and responds rationally. The following equations define the behaviour of level-$(t+1)$ agents [57]:

$$L_{t+1}(m \mid u) \propto \begin{cases} 1 & \text{if } m = \arg\max_{m \in M} S_t(u \mid m) P(m) \\ 0 & \text{o.w.} \end{cases}$$

$$S_{t+1}(u \mid m) \propto \begin{cases} 1 & \text{if } u = \arg\max_{u \in U} L_t(m \mid u) \\ 0 & \text{o.w.,} \end{cases}$$

where at $t = 0$, the speaker S_0 expresses things literally, and the listener L_0 assigns equal probability to all possible utterances $u \in U$ that are literally true for the meaning m from a set of possible meanings M. This step acts as a filtering mechanism to exclude irrelevant sentences. In subsequent levels, both agents assign equal probabilities to all best responses and zero otherwise. However, a limitation of the IBR model is that it treats each possible sentence as equally likely, which can be unrealistic in practical communication.

2.2.2.2 RSA

The most distinctive difference between the RSA framework and the IBR model is that, aside from S_0 (referred to as the literal speaker in RSA), the distributions of the agents are not necessarily uniform, as outlined in Equation 2.2.2.

In the RSA framework, there are two types of communication agents: literal agents (equivalent to level-0 agents in IBR) and pragmatic agents (equivalent to level-t agents in IBR). The framework iteratively updates the relation (on U, M) of the current level agent based on that of the other agent from the previous level.

The framework starts with a literal listener L_0:

$$P_{L_0}(m|u) \propto \delta_{m \in [[u]]} \cdot P(m), \tag{2.4}$$

where δ is the meaning function that returns 1 if u contains the meaning m and 0 otherwise. The prior function $P(m)$ indicates the prior probability of m being true.

Next, the pragmatic speaker S_1 reasons about L_0 and select the best u by:

$$P_{S_1}(u|m) \propto (P_{L_0}(m|u))^{\alpha}, \tag{2.5}$$

where α is a parameter that scale the rational level of S_1. A higher α will sharpen the probability distribution and vice verse.

Finally, the pragmatic listener L_1 selects their interpretation m by reasoning about S_1:

$$P_{L_1}(m|u) \propto P_{S_1}(u|m) \cdot P(m). \tag{2.6}$$

Zhou el. al show that RSA-based models align more to human predictions than the IBR model and the grammatical model in a reference game setting [57]. Nevertheless, both models serves as foundational building blocks for the more advanced RSA framework.

2.3 RSA-based models in the field

The RSA framework has been widely employed by researchers for various pragmatic reasoning tasks [14]. The core of these tasks is the generation and interpretation of referring expressions. Notably, Monroe et al. adapted an RSA model using an LSTM architecture for a colour reference game. Their findings indicate that the trained RSA-based neural model significantly enhances the accuracy of interpreting human-produced colour descriptions compared to a basic RNN listener. This improvement underscores the potential of the RSA framework for tasks necessitating grounded language understanding [37]. Similarly, researchers have successfully trained RSA models in image reference games [12, 13, 3] with pragmatically annotated training data, further underscoring the model's applicability in conceptual understanding tasks. These studies collectively highlight the RSA framework's robustness in enhancing interpretative interactions, and thus, we wonder whether LLMs, with their power on natural language generation and understanding tasks, are also using this framework internally.

2.4 Applying RSA to LLMs

To date, only one study has examined the pragmatic reasoning ability of LLMs at inference using the RSA framework. Carenini et al. [7] demonstrate that the reasoning mechanisms of the GPT-2 XL model can be accurately predicted within the RSA framework for a metaphor understanding task structured as "X is Y". In this study, the meaning space M is confined to a pre-defined feature set F, and the utterance space U

is limited to a pre-defined set of nouns. The experiment involves prompting the LLM to generate the adjective or feature f implied by the input metaphor and comparing the resulting distribution to that produced by the RSA model. Their findings indicate that the LLM's behaviour aligns with predictions made by the RSA pragmatic listener model in this task.

This study primarily evaluates the LLM as a pragmatic listener rather than a pragmatic speaker. Furthermore, the task is constrained to a specific metaphor format, which may not generalise to arbitrary natural language contexts.

2.5 Summary of related work and research gap

In summary, there is a growing trend in the XAI literature to evaluate the pragmatic reasoning ability of LLMs. The probabilistic RSA framework is considered the most powerful tool for pragmatics evaluation. Several studies have constructed and trained RSA-based language models for pragmatic tasks, finding that these models behave more similarly to human predictions compared to vanilla neural models. One study evaluated the pragmatic interpretation abilities of a LLM using the RSA framework in a metaphor understanding task, finding that the LLM's interpretation closely aligned with that of the RSA model. However, there is a notable gap in evaluating LLMs using the RSA framework, particularly regarding their performance as pragmatic speakers.

To address this, we propose a pipeline that integrates a reference game to evaluate the distribution generated by the RSA model in comparison with that of a standard LLM during inference.

Chapter 3

Methodology

In this chapter, we detail our proposed pipeline for assessing the communicative effectiveness of LLMs within the RSA framework, as shown in Figure 3.1. The research question will be quantified by comparing the output distributions of the LLM and an RSA model on a specific task. We begin by introducing the task specification for the evaluation — a text-based reference game, and the corresponding dataset. Subsequently, we construct a probability table that records the scores generated by each model for each instance of the reference game, where each utterance u describes an object o.

The methodology is organised into three stages: first, constructing the utterance U and meaning O spaces within the context of the reference game; second, scoring the probability table between U and O using both the vanilla LLM and the RSA model; and finally, evaluating the output distribution across various metrics.

3.1 Task specification

We use a reference game as the task for our pipeline [41, 28], where a set of objects O is presented, including a target object $o_t \in O$. Since our reference games include exactly one target object, the set of potential meanings is precisely the set of objects. The speaker's role is to choose from a set of possible utterances U the utterance u_t that most effectively conveys o_t to the listener. The listener, upon receiving the set of objects O and the selected utterance u_t, must identify the target object o_t.

This task is ideal for evaluating pragmatic reasoning within a goal-oriented communicative context and provides a straightforward framework for eliciting pragmatic and discourse-level phenomena. Since we work with pre-trained LLMs that operate on textual inputs, we use text-based reference games.

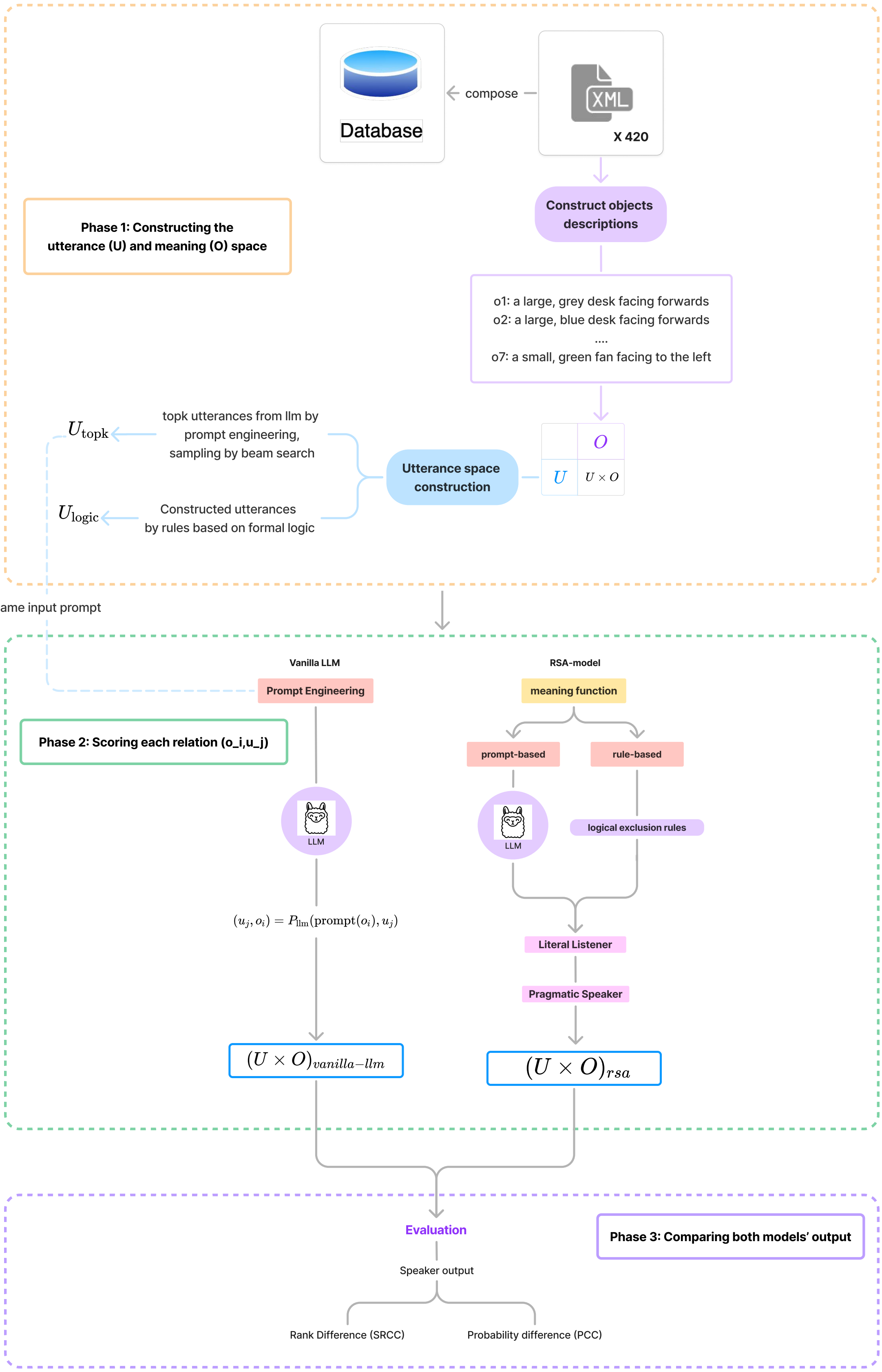

Database
compose
XML
X 420
Construct objects descriptions
Phase 1: Constructing the utterance (U) and meaning (O) space
o1: a large, grey desk facing forwards
o2: a large, blue desk facing forwards
....
o7: a small, green fan facing to the left
O
U
$U \times O$
Utterance space construction
U_{topk}
topk utterances from llm by prompt engineering, sampling by beam search
U_{logic}
Constructed utterances by rules based on formal logic
Same input prompt
Vanilla LLM
Prompt Engineering
Phase 2: Scoring each relation (o_i,u_j)
LLM
$(u_j, o_i) = P_{\text{llm}}(\text{prompt}(o_i), u_j)$
$(U \times O)_{vanilla-llm}$
RSA-model
meaning function
prompt-based
rule-based
LLM
logical exclusion rules
Literal Listener
Pragmatic Speaker
$(U \times O)_{rsa}$
Evaluation
Phase 3: Comparing both models' output
Speaker output
Rank Difference (SRCC)
Probability difference (PCC)

Figure 3.1: Project methodology pipeline

3.2 Dataset for the referring expression task

For the task specification, we employ the TUNA dataset [51], which is centred on a referring expression task grounded in images. The TUNA dataset organises each reference game world around seven distinct objects within the same domain, with predefined attributes and fixed possible features for each attribute, as outlined in Table 3.1. These fixed attributes and features enable the construction of a meaning and utterance space within a constrained set, thereby reducing the inherent complexity of natural language. For our experiment, we use data from the furniture domain, which includes 420 reference game worlds. Within each game world, we conduct a reference game for each object as the referent, resulting in a total of 2,940 reference game instances.

Attribute	**Possible features**
Type	chair, sofa, desk, fan
Colour	blue, red, green, grey
Size	large, small
Orientation	left, right, front, back

Table 3.1: Preset attributes and corresponding possible features for the ‘furniture’ domain in the TUNA dataset.

3.3 Construction of the meaning space

The TUNA dataset presents each object in an XML file [20], from which we parse the attributes of each object to construct the literal meaning utterances. For each reference game, we generate a noun phrase (NP) description for each object based on the following structure:

$$\texttt{a <SIZE>, <COLOUR> <TYPE> facing <ORIENTATION>} \tag{3.1}$$

Here, the values of each object’s attributes are mapped into the corresponding positions within this template. Thus, the meaning space of each reference game would be a set of 7 object descriptions (Examples given at the purple border box in Figure 3.1).

3.4 Construction of the utterance space

In the RSA framework, speakers and listeners interpret the meaning of utterances by taking into account other possible utterances that could have been used. These alternatives define the meaning space used in the probability table. In a reference game, the utterance space includes all potential utterances within the restricted world that could describe any object in the meaning space. An optimal utterance space would encompass both literal and pragmatic expressions, enabling a thorough assessment of communicative effectiveness. However, even in a restricted setting, the construction of such an utterance space, accounting for the wide variety of sentence structures and connotations found in natural language, can be difficult.

Previous research has predominantly concentrated on producing sentences that are pragmatic, rather than exploring the full spectrum of meaning generation [54]. In pragmatic referring expression generation, this typically involves sampling from a learned model during inference [54, 3, 37]. However, this method inherently produces only pragmatic sequences, as LLM is trained on pragmatic data.

We present two approaches for constructing the utterance space U. The first approach involves sampling the top-k utterances from LLM using the beam search algorithm, which produces pragmatic sequences with greater flexibility in phrasing. The second approach constructs both pragmatic and literal utterances based on logical rules, inspired by grammatical model logic. This method is particularly effective in a text-based reference game setting, where a literal utterance includes all relevant features, while a pragmatic sequence may involve omitting some features, thereby simulating more natural communication strategies.

3.4.1 Sampling top-k utterances from LLM

Several sampling methods are commonly used in the literature, with top-k [17], top-p [25], and temperature sampling [2] being the most prevalent. In top-p sampling, the sequence is generated by sampling from the smallest set of tokens whose cumulative probability exceeds a certain threshold p. Temperature sampling adjusts the smoothness of the probability distribution, allowing control over the diversity of the output. In contrast, top-k sampling involves selecting the next token from the k most likely tokens in the distribution.

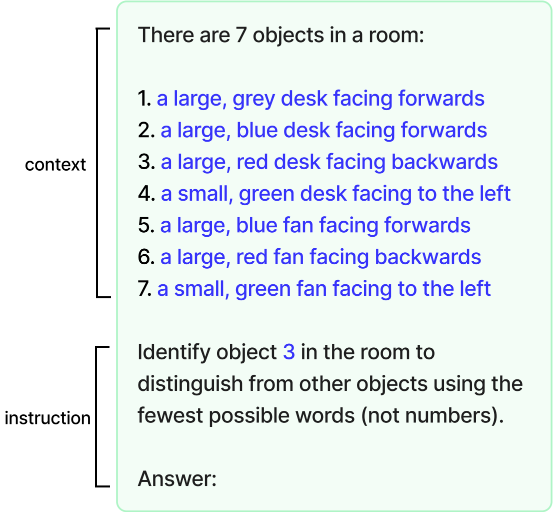
We choose top-k sampling because it allows us to evaluate sequences that are more likely to be generated by the language model, aligning with the default generation settings.

We sample top-k utterances from the LLM by a beam search presented as Algorithm 1. As observed by [54], humans are unlikely to consider the entire space of possible utterances each time they speak. Instead, we consider a smaller set of utterances, those which are both relevant to the topic at hand and aligned with typical patterns of speech. Additionally, this approach aligns with the practical constraints of text generation models, where evaluating the full search space is computationally infeasible due to the vast number of potential utterances [22, 36]. Beam search helps to mitigate this by narrowing down the possibilities to the most promising candidates, allowing the model to generate high-quality outputs without needing to explore every possible sequence.

3.4.1.1 Algorithm input

Before we go into the detail of the algorithm, we explain more about the construction of input to the algorithm.

Using a prompt-based inference method to sample from the LLM, we design the prompt by incorporating both the world context (the meaning space, encompassing the descriptions of all seven objects) and a clear task description to identify the target referent object o_t. To ensure that the task evaluates the inherent pragmatic reasoning ability of the LLM, the instruction is kept minimal and focused on the task's core objective, avoiding explicit guidance on pragmatic principles. An example of this prompt is shown in Figure 3.2.


context
instruction

There are 7 objects in a room:

1. a large, grey desk facing forwards
2. a large, blue desk facing forwards
3. a large, red desk facing backwards
4. a small, green desk facing to the left
5. a large, blue fan facing forwards
6. a large, red fan facing backwards
7. a small, green fan facing to the left

Identify object 3 in the room to distinguish from other objects using the fewest possible words (not numbers).

Answer:

Figure 3.2: Example of the prompt used for generating top-k sequences with the LLM. The blue text indicates variable elements specific to each reference game instance.

We set a beam width w of 60, determined through trial and error. We find that, with the designed prompt template, the LLM begins to generate redundant sequences when $w > 50$ approximately. Therefore, increasing the beam width beyond this point does not enhance the generation of pragmatic sequences and only results in unnecessary computational overhead.

The model M generates tokens with log probabilities, based only on the previous prompt, and outputs w possible tokens with their log probabilities for the next step.

3.4.1.2 Top-k Sampling process

In Algorithm 1, the sampling process begins by initialising a beam with an empty sequence and iteratively expands it by generating w alternative tokens for each sequence, updating their log probabilities accordingly. The beam is updated at each step to retain only the top w sequences, with the process continuing until all sequences in the beam are marked as complete with a boolean indicator, indicated by the presence of a stop token. This approach ensures the identification of the most probable utterances while controlling computational complexity.

During our exploration, we observe that without a starting word prompt, the LLM generates some sequences with consistent pragmatics but varied phrasing, such as “A red desk”, “red desk”, “a red desk”, “the red desk”, or “THE RED DESK” etc. Pragmatically, these four instances share two semantic meanings - “a red desk” and “the red desk” represent two different implications, with the latter suggesting the existence of only one red desk in the world. To maintain consistency and avoid unnecessary variability, such as different capitalisations that do not alter pragmatics, the generation process is initiated with both “a” and “the” separately. This also ensures that the answer is a noun phrase referring expression format. This approach ensures the capture of top-k sequences that are both relevant and pragmatically coherent within the context.

We generate the top 60 sentences starting with “a”, and only 20 with “the” due to time constraints. After generating this many sequences, we then deduplicate sentences that have the same semantic meaning but differ only in, for instance, ending punctuation.

The algorithm ultimately outputs w sequences along with their log probabilities.

3.4.2 Logic-based utterances construction from the world

We also construct sequences for the utterance space based on logical rules. The core idea is that in a reference game setting, the pragmatics of referring to a particular object

Algorithm 1: Beam search

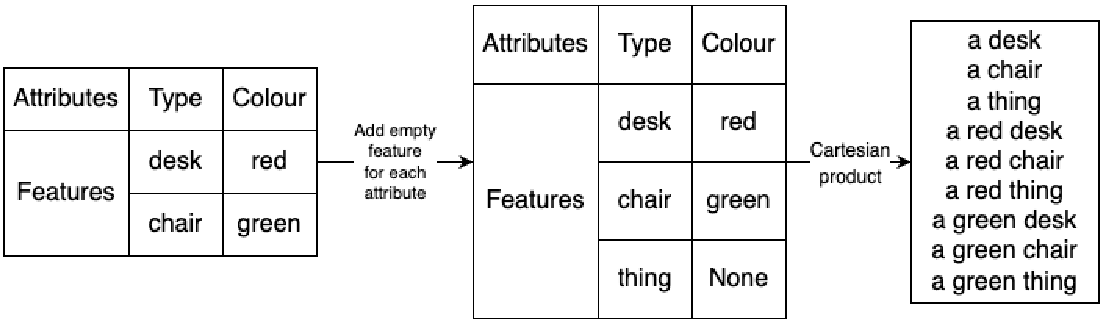
```
Input: prompt $P$, language model $M$, beam width $w$
Output: dictionary of generated utterances and their log probabilities
$B \leftarrow \{\varepsilon : (\texttt{logprob} = 0, \texttt{stopped} = \text{False})\}$;    /* Initialise a beam */
while $\exists u \in B. \neg B[u].\texttt{stopped}$ do
    $B' \leftarrow \emptyset$;
    for $u \in B$ do
        if $B[u].\texttt{stopped}$ then
            $B'[u] \leftarrow B[u]$;
        else
            /* The model produces $w$ alternative tokens $t$, with
               matching logprobs $p$.                              */
            for $(t, p) \in M(P + u, w)$ do
                $B'[u + t] \leftarrow ($
                    $\texttt{logprob} = B[u].\texttt{logprob} + p,$
                    $\texttt{stopped} = \texttt{"\n"} \in t,$
                $);$
            end
        end
    end
    $B \leftarrow B'$;
    if $|B| > w$ then $B \leftarrow \texttt{dict}(\ \texttt{sort}_{\texttt{logprob}}(\texttt{list}(B))[:w]\ )$ ;
end
return $\{u : B[u].\texttt{logprob} \mid u \in B\}$
```

involve using one or more of its features. When an utterance includes all the features of the object, it is considered a literal utterance. In contrast, a pragmatic utterance refers to expressions that omit some of the object's features. The formalisation is as follows:

$$F_* = \bigtimes_{A \in \text{Attributes}} (F_A \cup \{\varepsilon\}), \tag{3.2}$$

$$U_{\text{logic}}(O) = \{u(\mathbf{f}) | \mathbf{f} \in F_* | \exists\, o \in O . \mathbf{f} \subseteq o\}, \tag{3.3}$$

where F_* represents all possible combinations of features in the reference game setting, and O is a set of objects in a particular game. $U_{\text{logic}}(O)$ is a set of possible utterances that describe objects in O. $u()$ is a function we define to transform the feature set $\mathbf{f}$ to an utterance. In particular, since the generated sequence should follow a noun phrase format, ε for the 'Type' feature would be rephrased as 'thing'. Figure 3.3 displays an example of the logical construction process.



Attributes	Type	Colour
Features	desk	red
	chair	green

Attributes	Type	Colour
Features	desk	red
	chair	green
	thing	None

a desk
a chair
a thing
a red desk
a red chair
a red thing
a green desk
a green chair
a green thing

Figure 3.3: Example of logical construction process, given the attribute sets in the world.

3.5 Scoring the probability table

In the scoring phase, the aim is to examine the behaviour of the LLM as it generates a specific preset sequence, given the context and target object. This process is similar to how a pragmatic speaker produces an utterance to convey a particular meaning. The logit probabilities, which are the raw output values from the LLM before they are transformed into a probability distribution, directly indicate the model's preferences. Therefore, these logit probabilities are used as scores to assess the LLM's behaviour.

3.5.1 The vanilla LLM

For the vanilla LLM, the scores will be the log probabilities of the sequences within each reference game setting:

$$p(u|O,o_t) = p(\boldsymbol{u}|\boldsymbol{c}(O,o_t)) = \prod_{i=1}^{n} p(u_i|\boldsymbol{c}(O,o_t)\boldsymbol{u}_{1:i-1}), \tag{3.4}$$

where $\boldsymbol{c}()$ is the prompt template, as displayed in Figure 3.2, which produces a textual prompt for a particular object set O and designated referent o_t.

The scores for the top-k sequences will be generated along with the sequence using a beam search (Algorithm 1). For the logic-constructed sequences, we compute the probability retroactively for each utterance.

Since several of the constructed utterances share common prefixes with each other, we can reduce the computation time required for this calculation by using memoisation with a recursive form of the probability formula:

$$p(\boldsymbol{u}_{1:i}|\boldsymbol{c}(O,o_t)) = p(\boldsymbol{u}_{1:i-1}|\boldsymbol{c}(O,o_t)) \cdot p(u_i|\boldsymbol{c}(O,o_t)\boldsymbol{u}_{1:i-1}). \tag{3.5}$$

For example, consider the utterances "a large red desk" and "a large fan". When we compute the first sequence, we store the log probability of "a", "a large", "a large red" and "a large red desk". When we encounter the second utterance with the common prefix "a large", we need only use the LLM to compute the probability of the final token, since we can retrieve the rest from memory.

3.5.2 The RSA model

The RSA model's scoring process begins with the literal listener L_0:

$$P_{L_0}(o|u) \propto M(u,o) \cdot P(o), \tag{3.6}$$

where, in the context of our reference game, each object is equally likely to be selected, resulting in a uniform prior $P(o)$. The literal listener's interpretation relies entirely on the meaning function $M()$.

The function $M(u,o)$ returns a value in $[0,1]$, indicating whether the utterance u literally describes the object o, which is a task of natural language understanding. To achieve this, we construct two types of meaning functions: a prompt-based meaning function that leverages an LLM to make judgements, and a rule-based meaning function that determines the boolean value according to predefined logical rules.

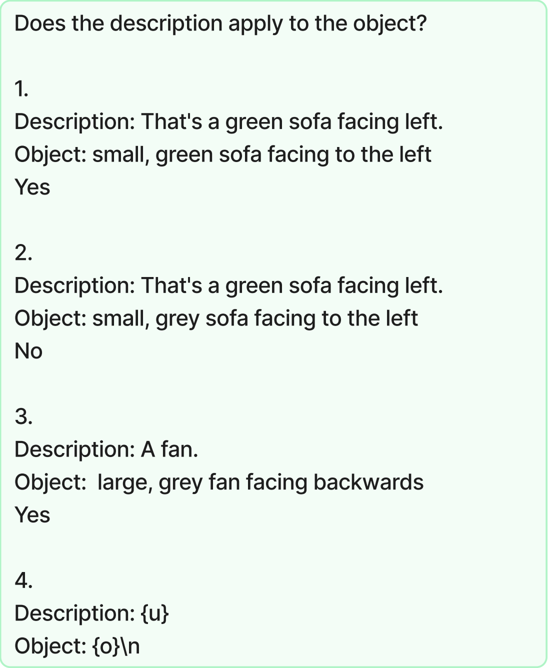
3.5.2.1 Meaning function - prompt-based

Large-scale language models excel in natural language understanding tasks and can be effectively leveraged during inference [55]. We employ prompt engineering to obtain a numeric score, using few-shot prompting techniques to guide the LLM by providing input-output examples that establish a fixed output template. Figure 3.4 illustrates the prompt used for the meaning function, which was refined through trial and error.

The prompt-based meaning function is defined as:

$$M_{\mathrm{p}}(u,o) = \frac{P(\mathrm{Yes} \,|\, \mathrm{LLM}(o,u))}{P(\mathrm{Yes} \cup \mathrm{No} \,|\, \mathrm{LLM}(o,u))}, \tag{3.7}$$

which is the probability of the model answering "Yes".



```
Does the description apply to the object?

1.
Description: That's a green sofa facing left.
Object: small, green sofa facing to the left
Yes

2.
Description: That's a green sofa facing left.
Object: small, grey sofa facing to the left
No

3.
Description: A fan.
Object:  large, grey fan facing backwards
Yes

4.
Description: {u}
Object: {o}\n
```

Figure 3.4: 3-shot prompt for the prompt-based meaning function using the LLM. The variables o and u represent the object and utterance, respectively, and are customised for each instance. Here, each new blank line corresponds to the newline character '\n' in actual implementation.

3.5.2.2 Meaning function - rule-based

The core idea of the rule-based meaning function is based on feature exclusion: an utterance u that includes a feature contradicting those of the object o does not describe o. For example, if o is "a large, grey chair facing forwards", then the utterance u as "a green thing" does not describe o because the colour feature in u contradicts the colour feature of o.

Before applying the rule, we first eliminate nonsensical generated utterances, such as "a", "a large", etc., which do not meaningfully describe any object. We also create a set of synonym mappings (see Appendix A.1), enhancing accuracy when scoring top-k sequences that exhibit greater variation in phrasing.

We define the rule-based meaning function as follows:

$$o = \{f_1, f_2, \ldots f_n\} \subsetneq F, \tag{3.8}$$

$$\overline{o} := F \setminus o, \tag{3.9}$$

$$M_{\mathrm{r}}(u,o) := \nexists w.(\exists f. f \in \overline{o} \wedge D(w,f)) \wedge (w \in u), \tag{3.10}$$

where $f_1, f_2, ...$ are the specific features of the object o, F is a full set of predefined features of the furniture domain in the TUNA dataset (Table 3.1), and D is a relation containing (w, f) iff word w describes feature f. This formula gives a Boolean result, which is mapped to $\{0, 1\}$ with 1 being True.

3.5.2.3 Meaning function - evaluation

We evaluate these two meaning functions by comparing their results on a set of test cases against human-labelled data. This evaluation is essential for fine-tuning parameters such as the number of examples given in the prompt, as well as for assessing the relative performance of the two meaning functions. We selected four constructed worlds for this purpose: "topk1" and "topk2", each comprising 2,072 (o, u) pairs generated from top-k sequences, and "logic1" and "logic2", containing 504 and 609 pairs respectively, with utterances derived from rule-based logical constructions.

For the prompt-based meaning function, we test with 3-shot prompts and 6-shot prompts (presented in Appendix B.1), and calculate the threshold T that would give the best performance for each n-shot prompt setting. The threshold allows us to compare the prompt-based meaning function to our ground-truth annotations, by considering values of at least T as 1 and other values as 0.

Table 3.2 displays the performance of the prompt-based meaning function across different metrics using n-shot ($n = \{3, 6\}$) with optimised thresholds for best performance, as well as the performance of the rule-based meaning function. Overall, the rule-based meaning function consistently identifies the literal relationships of all tested pairs, while the prompt-based method occasionally falls short. Notably, the prompt-based meaning function performs better with our constructed 3-shot prompt compared to the 6-shot prompt.

We consider the rule-based meaning function particularly well-suited for our reference game setting due to the restricted attribute set for each object. Although the generated top-k sequences may exhibit more diverse phrasings — such as when u_{topk} is "a tiny green table" and o is "a small green desk", where "tiny" and "table" are not present in the world vocabulary, the variations still fall within the attribute space of the furniture domain ($A_{\text{furniture}}$ = {'Type', 'Colour', 'Orientation', 'Size'}). Consequently, the rule-based meaning function can effectively capture these variations through synonym mapping.

We anticipate that a more carefully crafted prompt or a more advanced language model could improve the performance of the prompt-based meaning function, although this would necessitate additional human and time resources. Nonetheless, this type of meaning function may be more suitable in a more flexible task setting, where the relationships between o and u extend beyond literal templates.

	3-shot ($T = 0.5$)			6-shot ($T = 0.8$)			rule-based		
	Acc	P	R	Acc	P	R	Acc	P	R
topk1	0.99	1.00	0.92	0.95	0.83	0.87	1.00	1.00	1.00
topk2	0.98	0.96	0.89	0.97	0.98	0.83	1.00	1.00	1.00
logic1	0.96	1.00	0.81	0.93	0.91	0.78	1.00	1.00	1.00
logic2	0.95	1.00	0.71	0.94	0.91	0.78	1.00	1.00	1.00

Table 3.2: Performance of the prompt-based and rule-based meaning function across different metrics (Accuracy, Precision and Recall), the prompt-based meaning functions are using n-shot ($n = \{3,6\}$) with optimised thresholds T for best performance.

3.5.2.4 Pragmatic speaker

The pragmatic speaker S_1 of the RSA model is:

$$P_{S_1}(u|o) \propto e^{\alpha(\ln P_{L_0}(o|u) - \ln|u|)} = \left(\frac{P_{L_0}(o|u)}{|u|}\right)^{\alpha}, \tag{3.11}$$

where α is a scaling parameter that adjusts the level of rationality attributed to P_{S_1}, and $|u|$ is the length of the utterance which imposes a cost on longer productions. In our evaluation, we set $\alpha = 1$, as this allows us to directly assess the inherent rationality of the language model without artificially amplifying or dampening its behaviour.

3.6 Technical details

Although there are many popular pre-trained LLMs that excel in various downstream tasks, not many are open-source and provide access to logit probabilities during usage. For this project, we use the MiniCPM-Llama3-V 2.5 model [56], which is built on Llama3-8B-Instruct [50] available on HuggingFace[1] with a full license. Specifically, we employ the LLM with the `python-llama-cpp` library [2], which provides bindings for the source C++ implementation of the LLaMA model.

[1] https://huggingface.co/openbmb/MiniCPM-Llama3-V-2_5-gguf/tree/main

[2] Source code and Documentation available at: https://github.com/abetlen/llama-cpp-python

Chapter 4

Evaluation and Results

4.1 Data overview

Table 4.1 presents an example of a probability table showing the LLM and RSA models scores for possible utterances describing an object. Due to time constraints, we focus exclusively on the furniture domain within the TUNA dataset. This ‘furniture’ dataset consists of 420 XML files, each representing a reference game world containing 7 distinct furniture objects. In total, this results in 2,940 reference games corresponding to the same number of utterance sets, with each set describing one object in one world. We treat each utterance in each game as a separate utterance instance[1], totalling 427,200 instances. Among these, 298,200 instances are generated by top-k sampling, while 129,000 instances are produced using logic-based rules.

Regarding the models we experiment on, we have the probability outputs from the vanilla LLM, as well as two sets of outputs from the two constructed RSA-based models. One RSA model uses a prompt-based meaning function, while the other employs a rule-based meaning function. By examining the correlation between the vanilla LLM output and these two RSA models, we can not only determine whether the LLM behaves like a pragmatic speaker, but also investigate how the different approaches to constructing the meaning function influence this correlation.

[1] That is, utterances produced for two different games are considered as different instances, even if the utterances themselves are identical.

$o =$ **a large, grey desk facing forwards**

possible utterances	**Score_LLM**	**Score_RSA**	**sequence type**
a large grey desk facing forwards is in the room	6.143601e-13	0.008319	topk
a grey desk facing forward	7.591893e-13	0.015358	topk
a large desk	6.755013e-12	0.011092	topk
a grey desk facing forwards	7.591893e-13	0.015358	topk
⋮	⋮	⋮	⋮
a grey thing facing forwards	1.880327e-16	0.014261	logic
a grey thing facing to the left	1.932432e-21	0.000000	logic
a grey thing	1.561245e-07	0.033276	logic

Table 4.1: Example of a probability table showing the LLM and RSA model scores for possible utterances describing an object, categorised by sequence type (topk and logic). The RSA score here is caculated with a rule-based meaning function.

4.2 Evaluation intuition and selected metrics

There are two types of evaluations we aim to conduct. Firstly, we focus on each reference game separately. Specifically, for each object and all the utterances describing it, we want to determine whether the two models we compare exhibit similar scoring preferences for that set of utterances. To achieve this, we compare the two sets of scores using both the Pearson Correlation Coefficient (PCC) and Spearman's Rank Correlation Coefficient (SRCC). PCC measures the linear relationship between the two sets of data, providing insight into whether the scores are correlated and whether they change in the same direction. SRCC, on the other hand, assesses whether the models rank the scores in a similar order, regardless of the exact values. This allows us to evaluate the consistency in how the models prioritise or rank the different utterances, even if their scoring magnitudes differ.

We also aim to analyse the overall correlation across all reference games by examining the scoring for each utterance instance related to any object. This analysis will help us identify trends in scoring differences and determine whether the scores exhibit a linear relationship or are randomly distributed. To evaluate this, we will plot each data point in a common space (427,200 in total) to assess the strength of the overall correlation between the two models, using PCC and SRCC for quantification.

The first evaluation will provide insight into the correlation of the models' scoring within each reference game, allowing us to determine whether individual game instances impact scoring. The second evaluation will focus on the overall correlation of the models' scoring across all data points, independent of the game context.

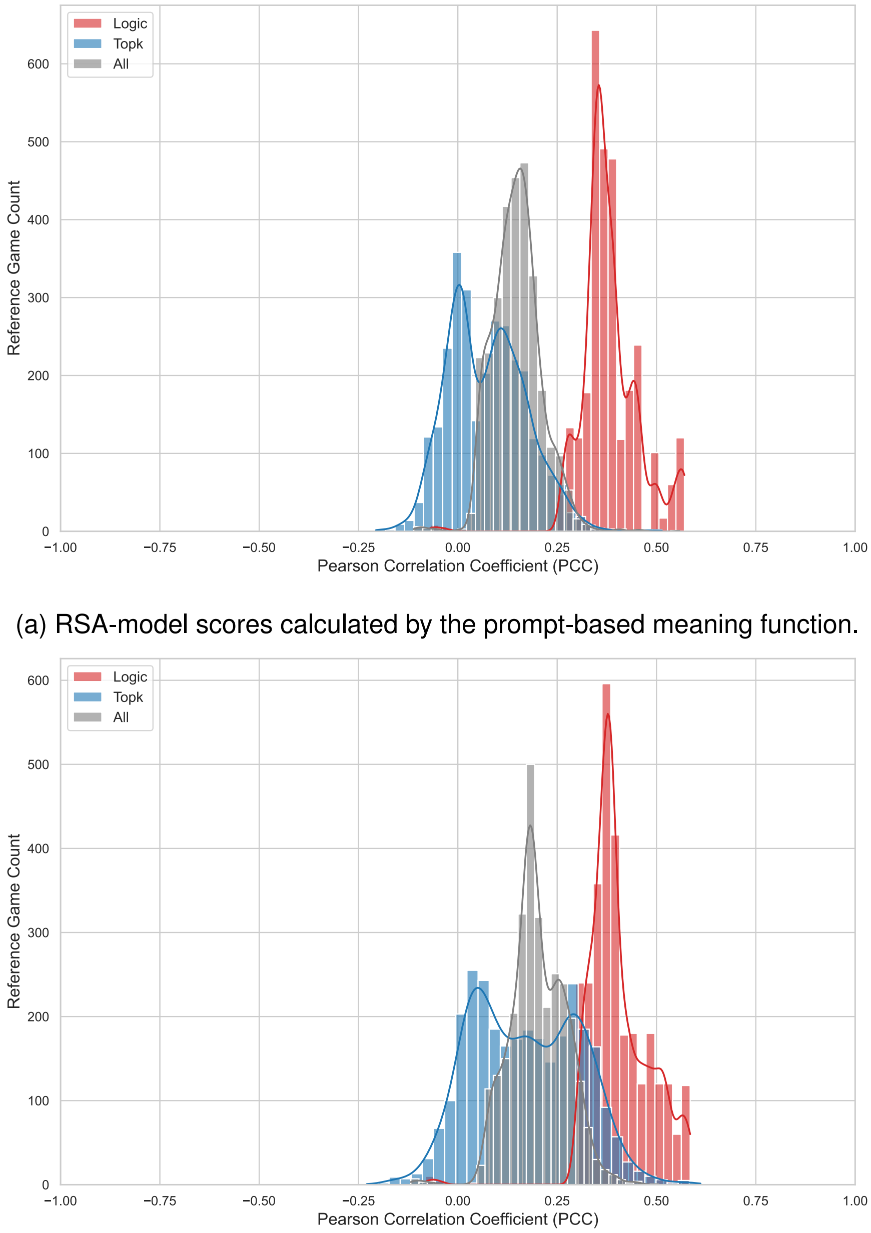
4.3 Evaluation between each reference game

4.3.1 Performance on PCC

Figure 4.1 illustrates the distribution of Pearson Correlation Coefficient (PCC) scores for each reference game, categorised by the type of utterance (logic-constructed or top-k generated) and the overall performance, irrespective of utterance type. The scores are calculated between the vanilla-LLM and each of the RSA-models using prompt-based and rule-based meaning functions separately.

Both subplots reveal similar patterns in the distribution of PCC scores. The PCC scores for top-k generated sequences display a broader spread, predominantly ranging between -0.25 and 0.25. In contrast, the PCC scores for logic-constructed sequences are mainly concentrated between 0.25 and 0.5, indicating a stronger correlation between the scores from the two models when evaluating logic-constructed sequences. This suggests that the two models are more aligned in their evaluation of logic-constructed utterances compared to top-k generated sequences. Overall, the positive correlation observed in most game sets across both subplots suggests that the LLM scores sequences in a manner that closely resembles a pragmatic model.

The primary difference between the two subplots is observed in the distribution of PCC scores for top-k generated sequences. In Figure 4.1b, where the RSA-model scores are calculated using the rule-based meaning function, the distribution is broader and more varied compared to Figure 4.1a, where the prompt-based meaning function is used. This broader distribution in the rule-based model indicates greater variability in the correlation between the RSA model and LLM when evaluating top-k generated sequences, suggesting that the rule-based meaning function may introduce more variation in the alignment of these models compared to the prompt-based function.



(a) RSA-model scores calculated by the prompt-based meaning function.

(b) RSA-model scores calculated by the rule-based meaning function.

Figure 4.1: Plots of Pearson Correlation Coefficient score count between the RSA (with different meaning functions) and LLM scores of utterances (divided by the source of the utterance) in each reference game.

4.3.2 Performance on SRCC

Figure 4.2 shows the distribution of Spearman's Rank Correlation Coefficient (SRCC) scores for each reference game, categorised by the type of utterance (whether logic-constructed or top-k generated) and the overall performance, regardless of utterance type. Similar to the PCC plots, the two SRCC distribution subplots display a comparable pattern, where the SRCC scores for logic-constructed sequences are more concentrated compared to those of the top-k generated sequences.

Notably, when SRCC scores were calculated across the entire sequence space for each reference game set, the distribution of correlation scores divided into two distinct groups: one primarily showing negative correlation, mostly between -0.25 and 0, and the other between 0.25 and 0.75, indicating positive correlation. This separation may be attributable to the different sources of utterance construction, where the characteristics of logic-constructed and top-k generated sequences impact their correlation with the RSA model differently.

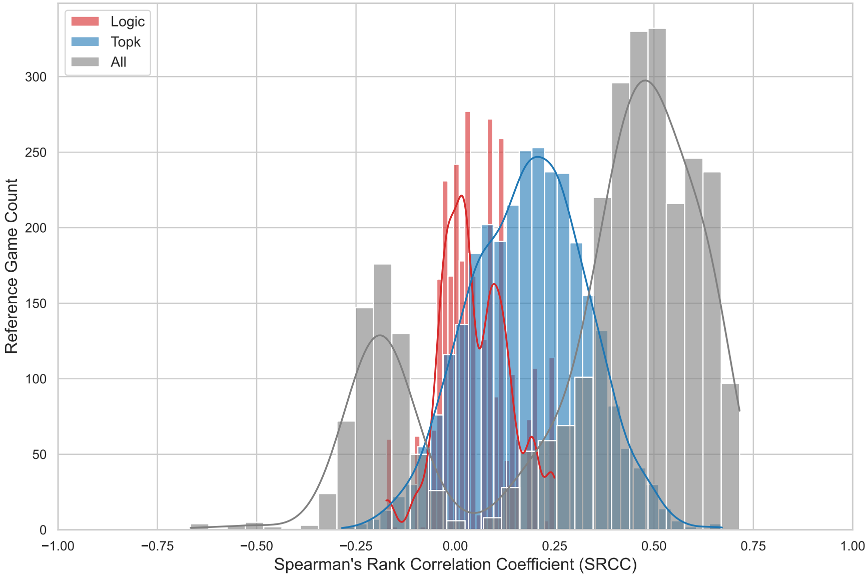
For the LLM model, the top-k scoring will be higher than that of the logic-constructed utterance. We also expect the top-k utterances to, at least, describe the object accurately, meaning that these utterances will generally receive a higher score from the RSA model than the average logic-constructed utterance (since, by design, the majority of those will not describe the target object). When this is the case, the combined ranking shows a strong correlation between LLM and RSA scores. However, often the top-k utterances will contain frequent hallucinations and therefore not describe the object accurately. When this occurs, the combined ranking contains a very weak correlation, since several of the logic-constructed sequences will be scored more highly by the RSA model than many of the top-k sequences.

Interestingly, the SRCC scores for the logic-constructed utterances cluster around 0.25 when the RSA model employs a rule-based meaning function. This suggests that the variability in the reference game settings has a reduced impact on the rank correlation between the two models' scoring of logic-constructed sequences when the RSA model uses a rule-based meaning function.

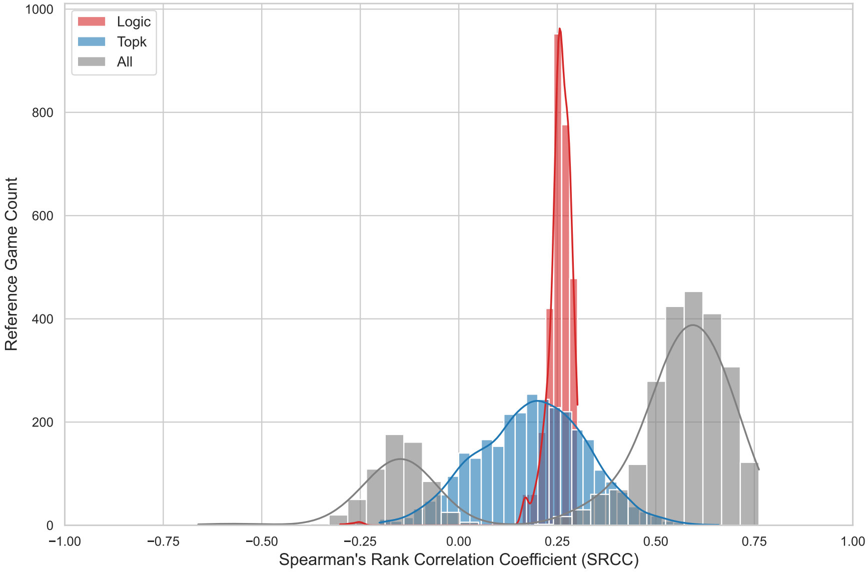
4.3.3 Results overview on evaluation between each reference game

Table 4.2 presents the mean scores and standard deviations of PCC and SRCC across all reference games, when the utterance space is composed of different utterance types, and the RSA model is calculated with two different meaning functions.

Overall, the 6 sets of model evaluation all show a positive correlation of the vanilla-LLM and the RSA model. In terms of the utterance type, when we only investigate the correlation when the models are scoring the logic-constructed sequences, the correlation of the models scoring are more strongly correlated than that of the top-k constructed



(a) Prompt-based meaning function



(b) Rule-based meaning function

Figure 4.2: Plots of Spearman's Rank Correlation Coefficient count between the RSA (with different meaning functions) and LLM scores of utterances (divided by the source of the utterance) in each reference game.

sequences. In terms of different types of RSA models, the table suggests that the vanilla-LLM has a better correlation to the RSA model that is calculated with a rule-based meaning function.

In summary, this table suggests that the rule-based approach for the RSA model's meaning function more accurately captures the LLM's tendency to produce pragmatic speech. Additionally, the vanilla-LLM and the RSA model are more correlated when scoring the logic-constructed utterances.

		PCC		SRCC	
Utt. Type	**RSA MF**	**Mean**	**σ**	**Mean**	**σ**
Logic	Prompt-based	0.382	0.074	0.050	0.086
	Rule-based	**0.405**	0.077	**0.255**	0.045
Top-k	Prompt-based	0.073	0.097	0.184	0.146
	Rule-based	**0.170**	0.134	**0.188**	0.137
All	Prompt-based	0.148	0.061	0.329	0.303
	Rule-based	**0.202**	0.071	**0.415**	0.321

Table 4.2: Comparison of the mean and standard deviation (σ) for PCC and SRCC metrics across different reference games, highlighting the correlation between the LLM model and two RSA models using different meaning functions (MFs), and with different utterance types.

4.4 Evaluation across all reference games

Figure 4.3 illustrates the correlation between the scoring of the vanilla-LLM and the RSA models using different meaning functions and different utterance types across all reference games. The scatter plots reveal that there isn't a clear linear relationship between the scores from the vanilla-LLM and the RSA models under the reference game setting.

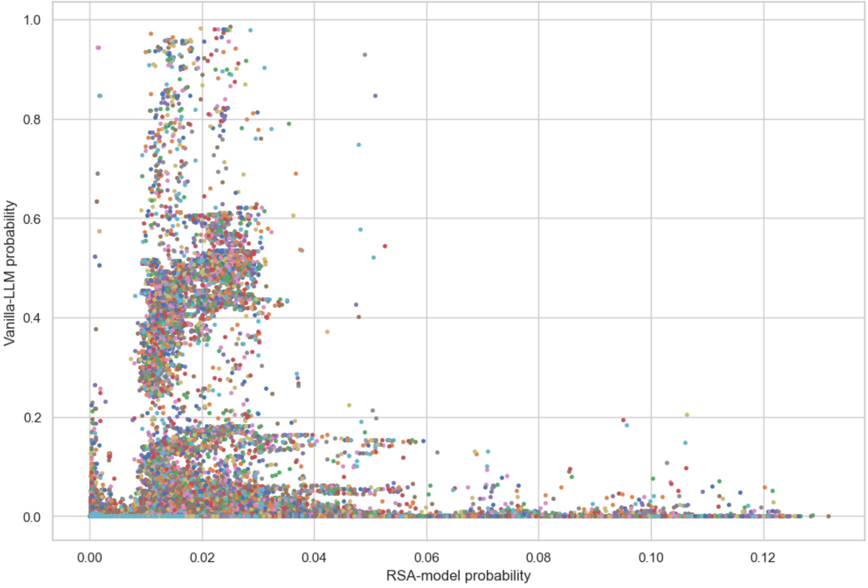
However, the distribution of points suggests that the LLM's scoring aligns better with the RSA model that is calculated with a rule-based meaning function (Figure 4.3b) than with a prompt-based function (Figure 4.3a), despite the overall lack of strong linear correlation. Interestingly, the scoring of the RSA-model with a prompt-based meaning function cluster between 0.00 to 0.04. We investigate whether using a probability-based meaning function makes the final scoring of the model too smooth, so we experiment by changing the probability to a boolean value using a threshold of 0.5, as investigated in Section 3.5.2.3. We find that even with a more extreme prompt-based meaning function scoring for the RSA model, the pattern of the plot does not change, as shown in Appendix C.1.

Notably, in the plot corresponding to the rule-based meaning function (Figure 4.3b), the data points are more tightly clustered, especially in regions where the RSA model assigns higher probabilities. This pattern might suggest that the nuances of the LLM's scoring is better captured by a RSA model with a rule-based meaning function, reflecting a more structured relationship, albeit still non-linear. In contrast, the prompt-based function (Figure 4.3a) shows a more dispersed distribution of points, indicating weaker and less consistent alignment with the LLM's output. These observations align with the quantitative results presented earlier, where the rule-based meaning function exhibited higher correlation metrics, suggesting its superior capacity to mirror the LLM's scoring tendencies, particularly in pragmatic contexts.

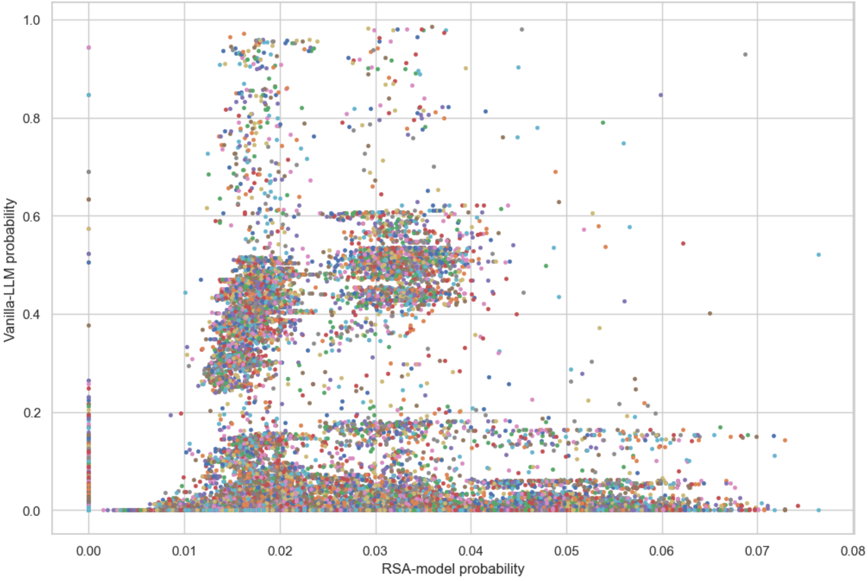
Table 4.3 shows the PCC and SRCC score across all utterances, scored by the vanilla LLM and each of the RSA models, regardless of the reference game context. The table shows that the correlation of the vanilla-LLM with the RSA models are positive. The vanilla-LLM model is more correlated to the RSA model with a rule-based meaning function for both metrics.

	PCC		**SRCC**	
RSA MF	**Score**	*p*	**Score**	*p*
Prompt-based	0.148	0.000	0.251	0.000
Rule-based	**0.203**	0.000	**0.342**	0.000

Table 4.3: Comparison of mean and standard deviation (σ) for PCC and SRCC metrics across different reference games for each of the two meaning functions.



(a) Prompt-based meaning function



(b) Rule-based meaning function

Figure 4.3: Visualisations of probability correlation of each utterance for every reference game on the vanilla-LLM and the two RSA models using different meaning functions.

Chapter 5

Discussion

The evaluation conducted in the previous chapter reveals several insights into the pragmatic generative capabilities of the vanilla-LLM within the context of our reference game setting. The analysis demonstrates that the vanilla-LLM generally behaves as a pragmatic speaker, as evidenced by the positive correlations observed between its scoring and that of the two constructed RSA models, particularly when a rule-based meaning function is employed for the RSA model.

5.1 Findings

A notable finding is that the vanilla-LLM aligns more closely with the RSA model that uses a rule-based meaning function than with the one that uses a prompt-based meaning function. In our reference game setting, the rule-based meaning function outperforms the prompt-based approach in factual judgement tasks. This stronger alignment indicates that the vanilla-LLM is particularly effective at making factual judgements in natural language sequences, where structured reasoning is crucial.

Moreover, the correlation between the LLM and the RSA models is more pronounced when evaluating logic-constructed utterances compared to top-k generated utterances. The higher predictability and structured nature of logic-constructed sequences likely contribute to this stronger correlation. In contrast, top-k generated sequences, which are prone to hallucinations, exhibit more variability and unpredictability, leading to weaker correlations. This variability underscores the challenges LLMs face in maintaining coherence and accuracy in less structured, more generative tasks.

While the overall findings support the LLM's pragmatic abilities within the restricted reference game setting, the generalisability of these results to more natural, everyday language use remains uncertain. The structured nature of the reference games and the specific construction of the utterance and meaning spaces may not fully capture the complexities of real-world communication, where the utterance space is vast and less constrained. However, this research provides a general framework for evaluating LLMs' pragmatic abilities, offering a foundation for extending these assessments to more natural and varied language use.

5.2 Limitations and Future Work

There are several limitations to the current study that warrant further investigation. Firstly, the RSA framework requires an exhaustive list of meanings and utterances, which is intractable in most real-world scenarios, particularly in complex and long-sequence tasks such as essay writing or dynamic conversational exchanges. This limitation points to the need for more scalable approaches to constructing meaning and utterance spaces in pragmatic reasoning tasks.

Apart from that, the text-based referring expression task is not an ideal task for evaluating the pragmatic reasoning ability of language models. We, and the RSA modelling community in general [12, 13, 3], use it as a benchmark as it allows us to fulfil two key requirements for using the RSA model: that the meaning function be computable, or at least reasonably approximable, and that the utterance and meaning spaces be small enough that applying that function to every utterance-meaning pair does not consume an impractical amount of resources.

However, in many reference games, and other referring expression tasks, a reasonable solution may be found by surface-level meaning and unambiguous descriptions, without needing to use scalar implicatures or other pragmatic reasoning methods. Thus, using the reference game task does not always apply the nuanced pragmatic reasoning ability of the LLM.

Future work should explore different datasets that reflect a broader range of communication settings and natural language use cases. Additionally, testing on other LLMs, particularly those with more advanced pragmatic reasoning capabilities like GPT models that are trained on a bigger dataset, would provide a more comprehensive understanding of how different LLMs handle pragmatic tasks. We could also explore how the alignment of the LLM compares to that of the RSA model when the RSA model is iterated multiple times, rather than just a single back-and-forth interaction as currently implemented.

In summary, while the findings provide valuable insights into the LLM's pragmatic abilities within a controlled setting, expanding the scope of evaluation to more diverse datasets and models is essential for advancing our understanding of LLMs in natural language communication.

Chapter 6

Conclusions

We present an evaluation of the communicative effectiveness of a large language model (a variant of Llama3-8B-Instruct) within the framework of the Rational Speech Act (RSA) model. Our approach quantifies this effectiveness by comparing the probabilities assigned by the LLM to certain utterances against those assigned by an RSA model.

To achieve this, we first establish a reference game task as the communication environment. We then develop a pipeline based on RSA principles to guide the comparison. The pipeline begins with constructing the utterance and meaning spaces, integrating world information. For constructing the utterance space, we propose two methods: sampling top-k sequences via beam search from the LLM and generating sequences based on logical rules derived from the world context. The next step involves scoring each utterance-meaning pair. For the LLM, scores are obtained by evaluating the probability of the LLM generating the utterance when prompted with the target referent. For the RSA model, we develop two variants based on different meaning functions: one that leverages the LLM's generative capabilities to assess whether an utterance conveys a particular meaning, and another that uses a rule-based meaning function specifically tailored to the reference game task. Finally, we compare the scores across reference games using Pearson and Spearman correlation coefficients.

Our results show a positive correlation between the LLM's scoring and that of the two RSA models, with a stronger alignment to the RSA model using a rule-based meaning function. We also examine how different types of utterances impact this correlation, finding that the models are more closely aligned when scoring utterances constructed by logical rules. While we acknowledge that these results may not be fully generalisable due to the limitations of the reference game task and the RSA framework, this project nonetheless contributes valuable insights. It offers a practical template for evaluating LLMs’ pragmatic abilities, which can be further refined and expanded in future research. This framework could serve as a foundation for more comprehensive assessments of LLMs’ communicative effectiveness, ultimately contributing to the understanding of the internal “reasoning” of LLMs, and the development of more sophisticated and human-like AI systems.

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[3] Jacob Andreas and Dan Klein. Reasoning about pragmatics with neural listeners and speakers. *arXiv preprint arXiv:1604.00562*, 2016.

[4] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Michael Canale and Merrill Swain. Theoretical bases of com-municative approaches to second language teaching and testing. *Applied linguistics*, 1(1):1–47, 1980.

[7] Gaia Carenini, Louis Bodot, Luca Bischetti, Walter Schaeken, and Valentina Bambini. Large language models behave (almost) as rational speech actors: Insights from metaphor understanding. In *NeurIPS 2023 workshop: Information-Theoretic Principles in Cognitive Systems*, 2023.

[8] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[9] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

[10] Gennaro Chierchia, Danny Fox, and Benjamin Spector. Scalar implicature as a grammatical phenomenon. In *Handbücher zur Sprach-und Kommunikationswissenschaft/Handbooks of Linguistics and Communication Science Semantics Volume 3*. de Gruyter, 2012.

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[12] Reuben Cohn-Gordon, Noah D Goodman, and Christopher Potts. An incremental iterated response model of pragmatics. *arXiv preprint arXiv:1810.00367*, 2018.

[13] Rodolfo Corona Rodriguez, Stephan Alaniz, and Zeynep Akata. Modeling conceptual understanding in image reference games. *Advances in Neural Information Processing Systems*, 32, 2019.

[14] Judith Degen. The rational speech act framework. *Annual Review of Linguistics*, 9(1):519–540, 2023.

[15] Daniel C. Dennett. True Believers:The Intentional Strategy and Why It Works. In *Mind Design II: Philosophy, Psychology, and Artificial Intelligence*. The MIT Press, 03 1997.

[16] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[17] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

[18] Michael Franke and Gerhard Jäger. Pragmatic back-and-forth reasoning. In *Pragmatics, semantics and the case of scalar implicatures*, pages 170–200. Springer, 2014.

[19] Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. Understanding social reasoning in language models with language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[20] Albert Gatt, Ielka van der Sluis, and Kees van Deemter. Xml format guidelines for the tuna corpus. 2008.

[21] Noah D Goodman and Michael C Frank. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829, 2016.

[22] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[23] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975.

[24] Ali Hashemi and Samran Daneshfar. An overview of pragmatism and pragmatism assessment. *Theory and Practice in Language Studies*, 10(5):584–591, 2020.

[25] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[26] Laurence Robert Horn. *On the semantic properties of logical operators in English.* University of California, Los Angeles, 1972.

[27] Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. A fine-grained comparison of pragmatic language understanding in humans and language models. arxiv. *arXiv preprint arXiv:2212.06801*, 2022.

[28] Robert M Krauss and Sidney Weinheimer. Changes in reference phrases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science*, 1:113–114, 1964.

[29] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[30] Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. In-context learning with many demonstration examples. *arXiv preprint arXiv:2302.04931*, 2023.

[31] Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*, 1, 2021.

[32] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[33] Renze Lou, Kai Zhang, and Wenpeng Yin. Is prompt all you need? no. a comprehensive and broader view of instruction learning. *arXiv preprint arXiv:2303.10475*, 2023.

[34] Annie Louis, Dan Roth, and Filip Radlinski. " i'd rather just go to bed": Understanding indirect answers. *arXiv preprint arXiv:2010.03450*, 2020.

[35] Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 2024.

[36] Clara Meister, Tim Vieira, and Ryan Cotterell. If beam search is the answer, what was the question? *arXiv preprint arXiv:2010.02650*, 2020.

[37] Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338, 2017.

[38] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[39] Aida Nematzadeh, Kaylee Burns, Erin Grant, Alison Gopnik, and Thomas L Griffiths. Evaluating theory of mind in question answering. *arXiv preprint arXiv:1808.09352*, 2018.

[40] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978.

[41] Seymour Rosenberg and Bertram D Cohen. Speakers' and listeners' processes in a word-communication task. *Science*, 145(3637):1201–1203, 1964.

[42] Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. The goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[43] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[44] Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. Neural theory-of-mind? on the limits of social intelligence in large lms. *arXiv preprint arXiv:2210.13312*, 2022.

[45] Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg, Maarten Sap, and Vered Shwartz. Clever hans or neural theory of mind? stress testing social reasoning in large language models. *arXiv preprint arXiv:2305.14763*, 2023.

[46] Benjamin Spector, Danny Fox, and Gennaro Chierchia. Hurford's constraint and the theory of scalar implicatures. *Manuscript, MIT and Harvard*, 2008.

[47] Settaluri Lakshmi Sravanthi, Meet Doshi, Tankala Pavan Kalyan, Rudra Murthy, Pushpak Bhattacharyya, and Raj Dabre. Pub: A pragmatics understanding benchmark for assessing llms' pragmatics capabilities. *arXiv preprint arXiv:2401.07078*, 2024.

[48] Settaluri Lakshmi Sravanthi, Meet Doshi, Pavan Kalyan Tankala, Rudra Murthy, and Pushpak Bhattacharyya. Do llms understand pragmatics? an extensive benchmark for evaluating pragmatic understanding of llms.

[49] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[51] Kees van Deemter, Ielka van der Sluis, and Albert Gatt. Building a semantically transparent corpus for the generation of referring expressions. In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 130–132, 2006.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[53] Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, page 3, 2023.

[54] Julia White, Jesse Mu, and Noah D Goodman. Learning to refer informatively by amortizing pragmatic reasoning. *arXiv preprint arXiv:2006.00418*, 2020.

[55] Ni Xuanfan and Li Piji. A systematic evaluation of large language models for natural language generation tasks. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 2: Frontier Forum)*, pages 40–56, 2023.

[56] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[57] Irene Zhou, Jennifer Hu, Roger Levy, and Noga Zaslavsky. Teasing apart models of pragmatics using optimal reference game design. In *Proceedings of the annual meeting of the cognitive science society*, volume 44, 2022.

Appendix A

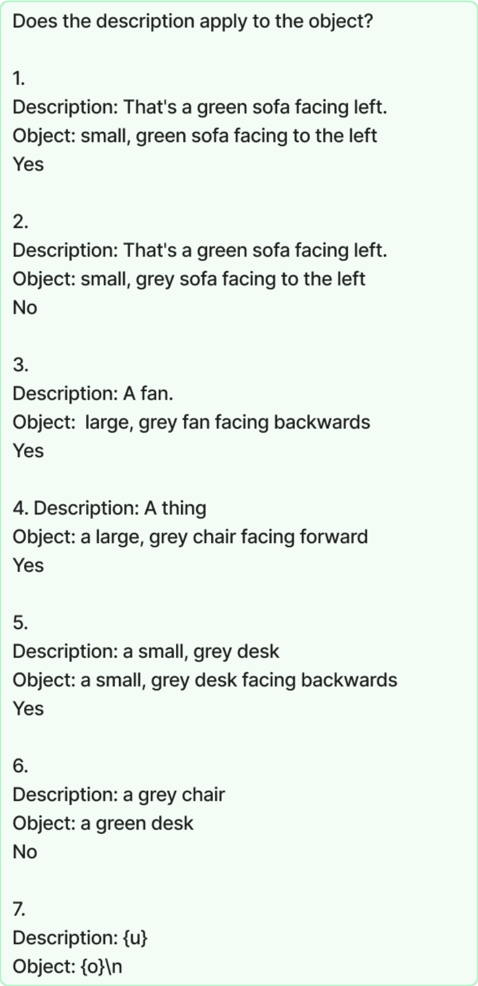
Synonym mapping for the features in the dataset

Features	**Corresponding synonyms**
desk	table
front	forward, forwards, facing upwards
back	backward, backwards, opposite direction, facing the wall, away
large	big
small	tiny, little
grey	gray

Table A.1: Corresponding synonyms for features of the selected furniture domain of the TUNA dataset, the synonyms are picked by manual checking the generated top-k sequences.

Appendix B

6-shot prompt for prompt-based meaning function



```
Does the description apply to the object?

1.
Description: That's a green sofa facing left.
Object: small, green sofa facing to the left
Yes

2.
Description: That's a green sofa facing left.
Object: small, grey sofa facing to the left
No

3.
Description: A fan.
Object:  large, grey fan facing backwards
Yes

4. Description: A thing
Object: a large, grey chair facing forward
Yes

5.
Description: a small, grey desk
Object: a small, grey desk facing backwards
Yes

6.
Description: a grey chair
Object: a green desk
No

7.
Description: {u}
Object: {o}\n
```

Figure B.1: 6-shot implementation for the prompt-based meaning function.

Appendix C

Additional Graphs

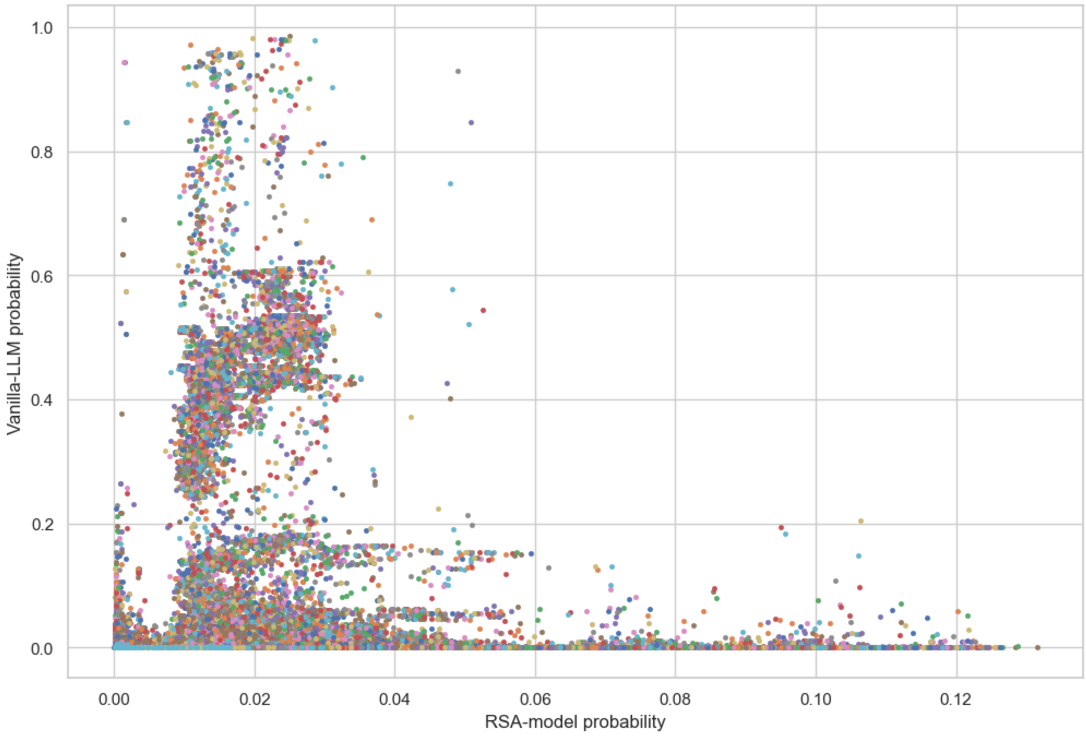


Figure C.1: Visualisations of probability correlation of each utterance for every reference game on the vanilla-LLM and the two RSA models using a prompt-based meaning functions. The meaning function scoring is further mapped to a boolean value using a threshold of 0.5.