Sampling Diverse Candidates in Pragmatic Reasoning

A person is standing on a surfboard, gliding over a wave.
Someone rides a surfboard, skillfully navigating the ocean.
A surfer is balancing on their board, cutting through the water.
An individual is surfing, maintaining balance on their surfboard.
The person on the surfboard rides a wave with practiced ease.

Layout: typeset by the author using LaTeX.
Cover illustration: Original image from Flickr.

Sampling Diverse Candidates in Pragmatic Reasoning

David Werkhoven
13909495

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 900
1098 XH Amsterdam

Supervisor
K. Naszádi

Language Technology Lab (LTL)
Faculty of Science
University of Amsterdam
Science Park 900
1098 XH Amsterdam

Semester 2, 2023-2024

Abstract

Effective communication between humans often depends on selecting the most appropriate utterance based on context, ensuring the listener interprets and acts upon the intended message. This selection process involves generating multiple candidate utterances and then selecting the optimal utterance from these candidates. This paper examines the impact of different decoding algorithms to generate such candidates and if diversity in these candidates plays a crucial role or not. To examine this, a speaker must describe a target object in a way that distinguishes it from two distractors, and the listener must identify the correct target to achieve a successful outcome. Specifically, this paper compares three decoding methods, Multinomial Sampling, Beam Decoding, and Diverse Beam Decoding to examine their effectiveness in this context. The findings highlight the importance of diversity within candidate utterances, through comparative analysis between the decoding methods. Diverse Beam Decoding comes forth as the most promising, demonstrating improved potential in fluency and accuracy compared to Multinomial Sampling and the traditional Beam Decoding method.

Contents

Chapter 1

Introduction

Language is the foundation of human communication. It allows humans to exchange their ideas, intentions and emotions, making us unique compared to other species. However, despite extensive research, the mechanisms of language-based communication are not yet fully understood. How do speakers convey meaning through words? How do listeners interpret intended messages despite the potential ambiguities in language? These questions remain unanswered, yet they are crucial for a complete understanding of linguistics.

"How exactly does an individual attempt to express their thoughts about a scenario to a listener?" This question raises curiosity, because it explores the complex process of translating personal experiences and perceptions into understandable communication, highlighting the complexities of human communication through language. Speakers in conversation carefully select their words. Their aim is to clearly illustrate scenarios for listeners, enabling them to visualize and comprehend the situation (Frank et al. 2012).

When selecting words, speakers often consider various options and then choose the ones they deem most fitting for the listener. This involves an internal process where various phrases, are weighed against factors such as the listener's background, knowledge, and potential interpretations. By choosing the most appropriate words, speakers attempt to communicate their message accurately and effectively, improving mutual understanding. This process can be seen as sampling, where speakers sample different linguistic options before settling on the best choice.

This paper will focus on the importance of sampling in conversations that require pragmatic reasoning. Specifically, it will explore the question, "Which sequence decoding technique generates the best candidates for pragmatic reasoning in an image referential game setting?". The pragmatic conversations will be simulated with the use of a referential image game.

The game in this paper consists of three images in total and involves a conversation between two types of agents, a Speaker and a Listener. The Speaker receives the three images and is informed of the target image that needs to be described. The Speaker's task is to generate a caption that describes the target image in a way that it can be distinguished from the other two images in the game. During the game, the Listener is presented with the three images along with the Speaker's caption. If the Listener correctly identifies the target image based on the caption, the game is won; otherwise, it is lost.

It is hypothesized that diversity in the candidate samples, generated by the decoding techniques, will impact the effectiveness of pragmatic reasoning in this context. This hypothesis is made due to the importance of having various candidates that are not similar to choose from when trying to describe the target. If all candidates are alike, the importance of choosing the optimal one decreases.

Chapter 2

Theoretical Background and Methodology

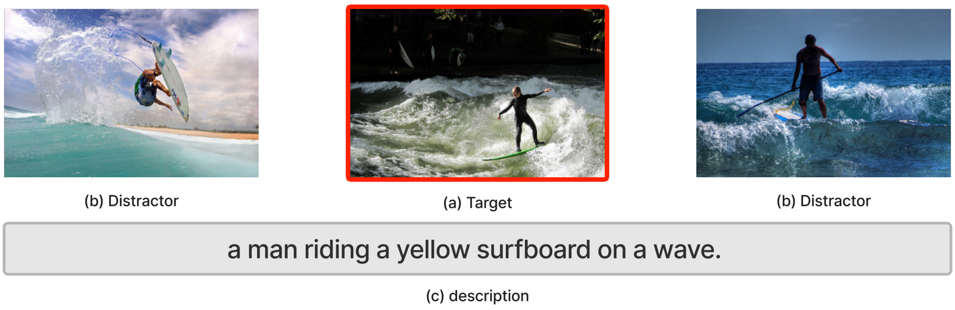
This chapter provides all the necessary background information. It begins by discussing the method used to evaluate pragmatic reasoning within the context of a reference game. After, it explains the different decoding methods.

2.1 Referential Messaging Games

This paper researches pragmatic reasoning in a referential game context. The referential games in this paper only consider real world images. The game is played between a listener L and a speaker S.

1. Reference candidates r_1 r_2 and r_3 are revealed to both players.
2. S is secretly assigned a random target $t \in \{1, 2, 3\}$.
3. S produces a description that distinguishes it from the other reference candidates $d = S(t, r_1, r_2, r_3)$, which is shown to L.
4. L chooses $c = L(d, r_1, r_2, r_3)$.
5. The speaker and listener win if $c = t$.

.



(b) Distractor

(a) Target

(b) Distractor

(c) description

Figure 2.1: The speaker provides a description (c) when given a target image (a) alongside two distractor images (b). The provided description highlights a yellow surfboard, which is present in (a) but not in (b).

Figure 2.1 illustrates a typical scenario of a reference game. To obtain successful collaboration between S and L, it is important that S effectively communicates and accomplishes the communicative goal of the game. The utterance generated by S must not only be accurate but also pragmatic, demonstrating a detailed understanding of L's behavior and expectations (Andreas et al. 2016). Several key components play an important role in generating such effective utterances in the context of a reference game.

2.1.1 Building Blocks for Multimodal Representation

The models for both the listener and speaker are created from a basic kit of tools designed for implementing multimodal representations of images and text. They make use of the following tools.

1. a image encoder REF_e

2. a description encoder DES_e

3. a choice ranker R

4. a image describer D

The image encoder REF_e encodes an image so the models can process it. The description encoder DES_e encodes a description, also known as an utterance, of an image to include it in the analysis. The choice ranker R takes an encoded description and a set of image encodings, assigning a score to each (description, image) pair. Finally, the referent describer D takes an image encoding and generates a description for the image.

Choice Ranker

The choice ranker takes an utterance and a set of image representations, assigning a score to each pair formed by an utterance and an image. These scores are then converted into a probability distribution over the images (Andreas et al. 2016). Essentially, the choice ranker enables the creation of a probabilistic model that determines which image from a set of image representations is most likely to correspond with a given utterance (Andreas et al. 2016).

2.1.2 Literal Listener

The literal listener $Listener_l$ functions by mapping utterances to images. It takes an utterance and a set of image representations, then selects the image considered to most likely match the utterance. This selection process is provided by encoding the utterance with the DES_e and the images with the REF_e. Then through the use of the choice ranker R, it creates the distribution that shows what image matches the utterance most (Andreas et al. 2016).

$$\begin{aligned} e_d &= DES_e(d) \\ e_{r_1} &= REF_e(r_1) \\ e_{r_2} &= REF_e(r_2) \\ e_{r_3} &= REF_e(r_3) \\ PListener_l(i|d, r_1, r_2, r_3) &= R(e_{r_i}|e_{r_{-i}}, e_d) \end{aligned}$$

Figure 2.2: The formula of model $Listener_l$. The listener makes use of the choice ranker to provide a distribution that allows the listener to pick the most fitting referent for an utterance.

The literal listener model $Listener_l$ is typically trained by maximizing the probability that the model correctly identifies an image given an utterance, distinguishing it from other images that closely resemble the target image (Andreas et al. 2016).

2.1.3 Literal Speaker

The literal speaker $Speaker_l$ is provided with an image, which serves as the basis for generating an utterance in English-language that describes the image accurately

(A. Liu et al. 2023). It produces multiple utterances describing the target image and provides a distribution indicating which utterance is most likely to describe the target effectively without any pragmatic context. This process is done through the use of the referent encoder REF_e and the image describer D (Andreas et al. 2016).

$$\begin{aligned} e_{r_1} &= REF_e(r) \\ PSpeaker_l(d|r) &= D(d|e) \end{aligned} \tag{2.1}$$

Figure 2.3: The formula of model $Speaker_l$, the speaker makes use of the image describer and the image encoder to produce a description for a given image.

The literal speaker is typically implemented using an LSTM-based utterance generation model, alongside an image recognition model, often based on architectures like ResNet (A. Liu et al. 2023), (Degen 2023), (Andreas et al. 2016), (White 2020).

2.1.4 Pragmatic Speaker

The pragmatic speaker $Speaker_p$ contains both $Speaker_l$ and $Listener_l$ models. It uses them to obtain an utterance distribution from the literal speaker and scores it with the literal listener to choose the most contextually-appropriate utterance (Andreas et al. 2016).

The core of pragmatic reasoning lies in the pragmatic speaker's ability to anticipate what the listener would consider the best sample. By making use of the literal listener's evaluations, the pragmatic speaker aims to identify and choose the utterance that would be most effective in communicating the target image to the listener (Degen 2023). This process of choosing the optimal utterance based on the literal listener's evaluations and literal speaker utterance distribution is also known as re-ranking (White 2020).

Sampling

Generating every possible utterance that a literal listener can produce based on a reference image is feasible when dealing with a small, fixed vocabulary. However, this task becomes infeasible with a large vocabulary, especially in reference games using real-world images (White 2020). For example, describing such images can quickly lead to vocabularies of 50,000 words or more. If we set a maximum utterance length of 10 words, the number of possible combinations would be on the order of 50000^{10}. This large number makes finding the optimal utterance using a pragmatic speaker computationally infeasible in such scenarios. To address this issue, sampling techniques are introduced. These methods use selected decoding strategies to generate a set of utterances using the literal speaker. An important aspect is the proposal distribution used for sampling sentences that are likely to be well understood by the literal listener (Andreas et al. 2016), displaying the importance of an effective decoding method. The process of the pragmatic speaker therefore becomes the following:

1. Draw samples $d_1, \ldots, d_n$ from $P_{\text{Speaker}_l}(\cdot|r_i)$.

2. Score samples: $p_k = P_{\text{Speaker}_l}(d|r)^{\lambda} \cdot P_{\text{Listener}_l}(i|d_k, r_1, r_2, r_3)^{\lambda-1}$.

3. Select d_k such that $k = \arg\max p_k$.

While step two would suffice without the use of *Speaker*$_l$ it typically is included in research due to its role in addressing imperfections in the listener model (Andreas et al. 2016), (White 2020), (A. Liu et al. 2023). Without *Speaker*$_l$, there is a potential issue where the speaker model might generate sentences that obtain the correct response from *Listener*$_l$, but do not resemble natural human language (Goodfellow et al. 2014). To prevent this, the pragmatic speaker considers two criteria. First, 'how likely is it that a listener would interpret this sentence correctly?' and second, 'how likely is it that a speaker would naturally produce it?' (Andreas et al. 2016). So by making use of the *Speaker*$_l$ in step two this preferred outcome is produced.

2.1.5 Rational Speech Act Framework

The introduction of multimodal approaches to solving reference games in a speaker-listener setting was first introduced by David Lewis (Lewis 1969). David Lewis's framework provided a foundation for understanding pragmatic reasoning in communication through the use of signaling games. Expanding David Lewis's work, the Rational Speech Act (RSA) framework was introduced by (Goodman et al. 2016). This research described RSA as a model that implements a social cognition approach to interpreting utterances (Goodman et al. 2016).

The introduced RSA model in (Goodman et al. 2016) discusses pragmatic reasoning within communication by modeling interactions between a speaker and a listener similar to the approach in (Lewis 1969). This combined approach offers a computational solution to pragmatically solving signaling games, such as reference games. Additional studies, such as (White 2020), extended the RSA framework to include variants like Sampled RSA and Full RSA. Where Full RSA uses the entire vocabulary to generate the optimal utterance and the Sampled RSA variant optimizes computational efficiency by sampling utterances from the literal speaker similar to how the pragmatic speaker utilizes the re-ranking in this paper.

RSA is considered an important framework, because it provides a computational solution to resolving reference games, therefore advancing our understanding of how pragmatic reasoning works in communicative contexts.

2.2 Decoding Methods

There are various decoding methods used to obtain samples from the literal speaker. In this paper three specific decoding methods will be discussed to obtain such samples.

2.2.1 Multinomial Sampling

Multinomial sampling is a technique used to generate utterances for reference game images. This algorithm produces interesting results due to the randomness involved in selecting words from a multinomial distribution. Specifically, it randomly selects the next token based on the probability distribution over the entire vocabulary provided by the model. Every token with a non-zero probability has a chance of being selected, which reduces the risk of repetition. However, this same randomness can sometimes lead to non-fluent or grammatically incorrect utterances.

The algorithm generates utterance samples for a reference game target image in the following way:

1. The number of utterances to be generated is specified for the multinomial sampling task, along with the image representation of the reference game target and the maximum utterance size.
2. The multinomial sampling task uses a multinomial distribution to sample a token for each utterance.
3. The task continues to iterate, selecting a new token for each sample utterance at every iteration until the maximum utterance length is reached.
4. The task returns the sampled utterances.

Multinomial sampling is computationally feasible, because it samples at each iteration, making it practical even with large vocabularies. This efficiency allows it to be applied in a reference game setting based on real-world images.

```
 1 # Encode the image representation
 2
 3 # Do this until the maximum sequence length is reached
 4 for i=1, ... I do:
 5     for j=1 ... J do: # For every sample utterance
 6         # Obtain the logits that represent the speakers
 7         # belief for each possible word at this iteration
 8         # by making use of the encoded image and speaker model
 9
10         # Use of multinomial distribution
11         # to draw a word sample
12
13         # Add the word to sample utterance
14
15 # Return set of B utterances
```

Figure 2.4: Pseudocode representing the multinomial sampling method. Where I is equal to the max sequence length of the utterances and J is equal to the amount of samples that need to be generated. This method utilizes the speaker model on line 6 and a multinomial distribution on line 10 to generate a set amount of utterances.

2.2.2 Pruning

Pruning is an important technique that is applied within the beam decoding algorithm, when generating utterance samples it reduces the number of candidate beams at each step of the decoding process. This reduction is important to improve runtime efficiency when generating samples. The goal of pruning is to focus computational resources on the most promising sequences, typically those with higher probabilities or scores according to a scoring function. Pruning helps to maintain a manageable search space within decoding algorithms while still aiming to capture high-quality samples. It ensures that the decoding algorithm remains computationally feasible even when dealing with large vocabularies.

2.2.3 Beam Decoding

Beam decoding is a search algorithm that can be used to draw samples for utterances of images. It is typically employed to generate sequences, and since the utterances describing images in a reference game are sequences of words, it is well-suited for this purpose (Vijayakumar 2016). Beam decoding is feasible within a reference game setting with real world images to generate samples as it implements pruning.

Beam decoding, stores the top-B highly scoring candidates at each iteration of the algorithm. At each iteration, beam decoding considers all possible single token extensions of the candidate beams given by the vocabulary and then selects the B most likely beam extensions for the next iteration. This selection process is where pruning is implemented (Vijayakumar 2016). Another method that can be applied in parallel with the top-B pruning, is to restrict consideration to only the top-K most likely single token extensions, this reduces the amount of extensions to increase computational efficiency. Since beam decoding maintains multiple hypotheses at each time step and eventually selects the one with the highest overall probability for the entire utterance, it can identify high-probability sequences that begin with low-probability initial tokens, which would have been overlooked by methods such as greedy search. However, while beam decoding allows for the exploration of multiple sequences in parallel, it often favors a single highly valued beam, resulting in outputs that differ only slightly from one another.

Sequence Length Normalization

An additional method that can be applied in beam decoding is sequence length normalization. This method ensures that all candidate utterances are treated equally. In beam decoding, finished utterances can appear within the top-B beams selected for the next iteration. Since these beams cannot be extended further, they are stored for re-evaluation at the end of the utterance generation process. Bias towards choosing shorter sequences occurs, because the overall probability of a sequence decreases as its length increases. Sequence length normalization is applied to prevent this bias when re-evaluating all found candidates.

$$\text{P'}(U) = \frac{\log P(U)}{N^{\lambda}}$$

Figure 2.5: The formula for sequence length normalization, where U is the probability of a finished utterance and N represents the length of that utterance. λ is the penalty weight that scales the log-probability of the sequence by a factor related to its length. When $\lambda = 1.0$, it is equivalent to averaging the log-probabilities.

```
 1 # Encode the image representation
 2
 3 # Do this until the maximum sequence length is reached
 4 for i=1, ... I do:
 5     # Obtain the logits that represent the speakers
 6     # belief for each possible word at this iteration
 7     # by making use of the encoded image and speaker model
 8
 9     # sample utterance
10     for j=1 ... J do: # For every sample utterance
11         # Extend the utterance with all words
12         # Optionally apply top-K to only extend with
13         # The top-K words
14
15     # Perform top-B pruning on all extended utterances
16     # Store finished utterances within the top-B utterances
17
18 # Perform sequence length normalization
19 # Perform top-B pruning on all found utterances
20 # Return set of B utterances
```

Figure 2.6: Pseudocode representing the beam decoding method. Where I is equal to the max sequence length of the utterances and J is equal to the amount of samples that need to be generated. This method utilizes the speaker model on line 5 and extends all beams on line 11, afterwards beam pruning is applied on line 15.

2.2.4 Diverse Beam Decoding

Diverse beam decoding is a variation of beam decoding that stimulates more diversity in the outputted sequences. In standard beam decoding, there is a habit for beams to be very similar, which can be computationally wasteful (Vijayakumar 2016). To prevent this, diverse beam decoding introduces several innovations.

Instead of treating all beams equally, diverse beam decoding organizes them into groups. Within each group, the algorithm operates independently. The diversity among candidate sequences is measured using a dissimilarity term, calculated per group instead of across all beams. This group adjustment avoids the need to compare every beam with every other, which leads to computational efficiency (Vijayakumar 2016).

The algorithm operates left-to-right through time and top to bottom through groups. This means that the algorithm iterates like traditional beam search, but does it per group. This involves independently performing beam search for each group. After each group iteration, the algorithm adjusts the probabilities of the beams using a diversity penalty γ. When doing so, it holds the previous groups fixed.

These adjustments discourage the generation of identical or overly similar beams. By focusing on diversity, diverse beam decoding promotes a wider range of potential sequences, thereby improving the effectiveness of the decoding process (Vijayakumar 2016).

```
1  # Encode the image representation
2
3  # Do this until the maximum sequence length is reached
4  for i=1, ... I do:
5      # Obtain the logits that represent the speakers
6      # belief for each possible word at this iteration
7      # by making use of the encoded image and speaker model
8
9      for g=1 .. G do: # For every group
10
11         # For every sample utterance in the group
12         for j=1 ... J do:
13
14             # Extend the utterance with all words
15             # Optionally apply top-K to only extend with
16             # The top-K words
17
18         # Perform top-B pruning on all extended utterances
19         # Store finished utterances within the top-B
20         # utterances of this group
21
22         # Adjust the probabilities of the chosen beams
23         # To lower their chances in the upcoming groups
24
25 # Perform sequence length normalization
26 # Perform top-B pruning on all, found utterances
27 # Return set of B utterances
```

Figure 2.7: Psuedocode representing the diverse beam decoding method. Where I is equal to the max sequence length of the utterances and J is equal to the amount of samples that need to be generated. G is the amount of groups. On line 9, it is visible that an additional loop is created to iterate through each group, showing one of the primary differences between beam decoding and diverse beam decoding. Another important difference occurs on line 22, where adjustments are made to the probabilities of selected beams, resulting in varied top-K words in following group iterations.

Chapter 3

Related Work

This chapter discusses research that is relevant to this study. It features findings from other referential game research implemented within a reference game setting and reviews the work that introduced the diverse beam search decoding algorithm.

3.1 Solving Referential Games

In the research of referential games, various studies have examined into understanding how language develops through interactive communication. Studies have explored the differences between Full RSA and Sampled RSA models, evaluating whether Sampled RSA can achieve comparable accuracy. Additionally, previous research has used reference games involving navigation tasks or simple images rather than real world images. Furthermore, studies have applied reinforcement learning to the pragmatic speaker within a reference game setting, investigating its impacts on communication strategies.

For instance, (White 2020) applied different speaker models to two seperate referential game scenarios. One involved a dataset named "shapeworld", featuring a small variety of differently colored shapes, while the other focused on colors that were similar to each other. Their findings highlighted that computing the Full RSA model for the color-based referential game was infeasible due to the large vocabulary required, this shows the need for sampling in more difficult referential game settings. They also noted that in more challenging reference games like those involving the colors dataset, speaker utterances needed to be effective, concise, and conventional. On the other hand, the simpler reference games did not need strict conventional language use, neither did they require lengthy utterances, as the targets were easier to describe with fewer words.

Another study by (Andreas et al. 2016) implemented a referential game setting using images where identically drawn characters were presented performing various tasks. Part of the study was to evaluate the number of samples required for generating effective utterances that accurately described the target. This evaluation is important since a too large number of samples could cause the model to be too slow to use in practice. When drawing samples from the base speaker model, the findings mention that utilizing 100 samples provided a reliable baseline for producing suitable descriptions. The result that drawing 100 samples improves the accuracy score, highlights the importance of diversity, as selecting from the top ten samples proves less effective compared to choosing from a larger pool of 100.

Both studies show the importance of sample quality in winning reference games. (White 2020) highlights that sample fluency and conciseness impacts accuracy, while (Andreas et al. 2016) indicates that increasing the amount of samples results in higher accuracy. Therefore, using more diverse samples could potentially increase fluency and accuracy, without using a large number of samples.

3.2 Decoding Diverse Solutions from Neural Sequence Models

In the research by (Vijayakumar 2016), diverse beam search is proposed as a solution to solve the issue of generating similar utterances encountered with the traditional beam search algorithm. The study evaluates the performance of beam search and diverse beam search by applying both methods to caption various images. It highlights that the complexity of the input images affects the diversity of generated captions. Complex images allow for varied descriptions, while simpler images do not.

The findings mention that traditional beam search often produces similar utterances, resulting in inherent ambiguity in tasks like image captioning. Diverse beam search, on the other hand, outperforms it by generating more diverse utterances while maintaining computational efficiency. As a result, the research concludes that diverse beam search exceeds traditional beam search in various domains, making it a reliable alternative.

Since beam search can introduce inherent ambiguity in image captioning, it is likely to do the same in a reference game setting. The objective in reference games is to describe the target in a way that distinguishes it from the distractors. Therefore, diverse beam search is expected to outperform beam search in the results of this study.

Chapter 4

Experimental Setup

This paper focuses on two main outcomes, the speaker's ability to accurately describe the target in a manner that distinguishes it from the distractors, and the diversity of the samples. These aspects are combined to assess the impact of diversity during the sampling phase since it is hypothesized that diversity improves the quality of candidate utterances. To evaluate these outcomes, the following setup is established.

4.1 Reference Game Dataset

The images for the referential games are obtained from the Microsoft COCO: Common Objects in Context dataset, the dataset is introduced in (Lin et al. 2014). The reference game images used in evaluation are based on the development set to ensure that none of the images were used in training the speaker and listener models. This selection prevents any overlap and guarantees an unbiased evaluation.

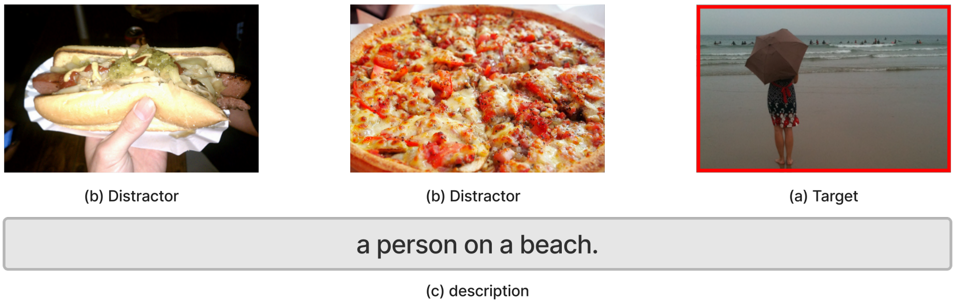
To ensure reproducible results, the reference games were drawn using a fixed seed value of 517 for the uniform random selection process. This seed value allows for exact replication of the reference games used in the evaluation.

4.1.1 Distractor Difficulty

The evaluation involves two games with different levels of distractor difficulty, one easy and the other hard. In harder games, the difficulty increases from the greater similarity among the target and distractor images. The decision to evaluate hard and easy games is guided by previous research, such as (Vijayakumar 2016), (A. Liu et al. 2023) and (White 2020), which suggests that distractor images similar to the target referent require more complex captions to accurately describe them. Therefore, different results are expected, as it is likely that varying levels of diversity and fluency in the samples are needed to solve such games.

Easy Difficulty

Easy difficulty games are created by initially selecting the target images for each game, this selection is made randomly and uniformly. Thereafter, for each target image, a set number of distractors N is drawn uniformly at random. During the evaluation of this paper, N is set to two.



(b) Distractor

(b) Distractor

(a) Target

(c) description

Figure 4.1: An example of an easy reference game where the distractors (b) does not closely resemble the target (a), winning involves describing the image (c) without needing much detail.

Hard Difficulty

Hard difficulty games are generated using the same method as described in (A. Liu et al. 2023). Similar to drawing easy difficulty games, each target image is randomly and uniformly selected. Following this, for each target image, N distractors are chosen. However, rather than selecting them uniformly at random, a visual and textual similarity approach is applied.

Visual-based Cosine Similarity

The most visually similar images are computed based on the cosine similarity of all image embeddings from a pretrained ResNet model. For each image in the evaluation set, a similarity ranking of the next 1000 most visually similar images is saved.

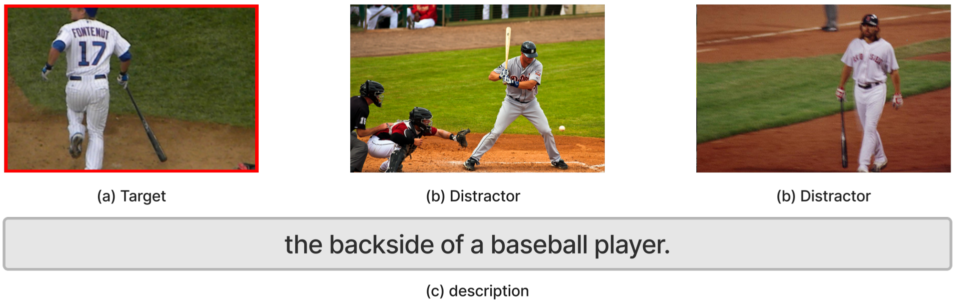
Text-based Cosine Similarity

The textual similarity is based on the image captions included in the dataset introduced in (Lin et al. 2014). The textual similarity is computed by applying cosine similarity between vector representations of the image captions. The vector representations are computed from embeddings generated using either the pretrained RoBERTa model (Y. Liu et al. 2019) or the pretrained CLIP mean pooled model available in the sentence-transformers library (Reimers et al. 2019). Again a similarity ranking of the next 1000 most similar images is saved.

Visual and Textual Cosine Similarity

The combined similarity ranking implements both visual-based and textual-based cosine similarities. This process involves combining their respective similarity scores.

The hard reference games were generated by uniformly and randomly choosing N distractor images from the set of 1000 images based on visual and textual cosine similarity. During evaluation N is set to two.



(a) Target

(b) Distractor

(b) Distractor

the backside of a baseball player.

(c) description

Figure 4.2: An example of an hard reference game where the distractors (b) closely resemble the target (a), winning involves providing a detailed description (c) that effectively distinguishes the target from the similar distractors.

4.2 Pragmatic Speaker Model

This section describes the implementation of the various components used to model the pragmatic speaker used during evaluation.

4.2.1 Referent Encoder

Referent images are encoded using a pretrained ResNet model, introduced in (He et al. 2016). This model processes each image to produce an embedding vector that represents its features.

4.2.2 Description Encoder

Descriptions are encoded using the pretrained RoBERTa base tokenizer, introduced in (Y. Liu et al. 2019). During evaluation the tokenizer has a vocabulary size of 50 265. It tokenizes each utterance, assigning a unique token ID to each token, therefore encoding the utterance.

4.2.3 Literal Listener Model

The literal listener receives a set of candidate images along with an utterance. It uses the referent encoder to encode each image and the description encoder to encode the provided utterance. Afterwards, it computes the dot product of each image's encoding with the utterance encoding, followed by applying the softmax function to these dot products. This process determines the probability of how closely related each image is to the utterance.

4.2.4 Literal Speaker Model

The literal speaker makes use of the referent encoder and an LSTM-based utterance generation model. After generating a referent encoding using the referent encoder, the speaker autoregressively generates an utterance using the LSTM network. The process of utterance generation implements one of the three earlier mentioned decoding methods. A vocabulary size of 50 265 is used.

4.2.5 Pragmatic Re-ranking

The pragmatic speaker uses the components discussed in this section to choose the optimal utterance that best describes the target referent. It makes use of the sampled utterances from the literal speaker and combines their probability scores with those from the literal listener to select the utterance with the highest probability. During evaluation, equal weight is given to both aspects.

4.3 Decoding Method Parameters

This section mentions the parameters used for the three implemented decoding methods including the evaluated sample sizes and the reasoning behind their selection.

4.3.1 Multinomial Sampling

Multinomial sampling is evaluated using three different sample sizes 5, 10, and 25. The utterance length was fixed at 10, and a seed value of 517 was used for the distribution. These specific sample sizes were used to make sure comparison with diverse beam decoding was possible. Due to its long runtime, sample sizes larger than 25 were not feasible for this study.

4.3.2 Beam Decoding

Beam decoding is evaluated using three different beam sizes 5, 10, and 25. The utterance length was fixed at 10 and the top-K selection of words to extend the beams with was equal to the beam size in all scenarios. These specific sample sizes were used to make sure comparison with diverse beam decoding was possible. Due to its long runtime, sample sizes larger than 25 were not feasible for this study.

4.3.3 Diverse Beam Decoding

Diverse beam decoding is evaluated using three different beam sizes 5, 10, and 25. Each with different group sizes, varying between 2, 5, 10 and 25. Additionally, different diverse penalty weight values 0.2, 0.8 and 0.01 were used. The diverse penalty weight values are based on findings in the (Vijayakumar 2016) research, the research mentions values between 0.2 and 0.8 work best, the value 0.01 was used to simulate an extreme amount of diversity. The utterance length was fixed at 10 and the top-K selection of words to extend the beams with was equal to the beam size in all scenarios. These specific sample sizes were used to ensure the algorithm could be computed efficiently. Sample sizes larger than 25 took too long to compute and were therefore not feasible for this study.

4.4 Evaluation Metrics

This section describes the metrics used to evaluate the pragmatic speaker's ability to reason pragmatically and solve reference games. It will discuss the use of these metrics and explain how they were computed.

4.4.1 Accuracy

The accuracy is evaluated using the average accuracy, which measures how often the listener correctly selects the target, for N games. The evaluation is done by dividing the number of times the target is correctly chosen by the total number of games played.

Let A be the accuracy value, C be the amount of correct games and N the amount of games played.

$$A = \frac{C}{N}$$

4.4.2 Entropy

The entropy is evaluated using the average entropy, which measures how certain the speaker is about its choice. If the entropy is low, the speaker is more confident in its choice. On the other hand, higher entropy indicates greater uncertainty. If diversity results in lower entropy, it means the model becomes more confident due to the use of diverse sentences.

4.4.3 Distinct n-grams

During evaluation the diversity is measured using the amount of distinct unigrams, bigrams and trigrams. For each game, the count of distinct n-grams is computed from all sampled utterances. Before calculation, utterances are filtered to include only words. This excludes special tokens such as EOS or BOS tokens. The evaluation included two different n-gram measurements

The absolute number of n-grams represents the amount of distinct n-grams found within the samples of a game. Let A be the absolute amount of n-grams, N denote the amount of samples and G the amount of distinct n-grams in a single sample. Then the absolute amount of n-grams is.

$$A = \sum_{i=1}^{N} (G_i)$$

The normalized number of n-grams is calculated by dividing the absolute amount of distinct n-grams through the total number of words in all the combined samples of a reference game. It represents a comparable value to the other decoding methods because it is normalized, meaning that the length of the utterance does not affect the comparison. Let A be the absolute amount of n-grams from a reference game and N be the total amount of words in the reference game, then the normalized amount of n-grams n is.

$$n = \frac{A}{N}$$

Chapter 5

Results

This chapter presents the evaluation results, which cover two methods, performance and diversity. Both methods were applied to easy and hard reference games. The performance results show which of the three evaluated decoding methods is optimal for solving the reference games. The diversity results display which decoding method produces the most varied samples. By combining these methods, an observation can be made about whether diversity in the samples impacts the ability to correctly solve a reference game. Throughout the results, beam decoding will be referenced as BD and diverse beam decoding is referred to as DBD.

5.1 Performance

5.1.1 Easy Reference Games

In this section, the easy reference games are evaluated based on accuracy, entropy, and runtime. The tables all represent the same decoding methods performed with different numbers of samples. Table 5.1 shows the results for 5 samples per game, where multinomial sampling is the most accurate and confident method. Table 5.2 shows the results for 10 samples per game, with DBD (10 groups and a diversity penalty of 0.01) achieving the highest accuracy score, although multinomial sampling remains the most confident approach due to its lower entropy score. In Table 5.3, the results for 25 samples per game are displayed, and once again, multinomial sampling is the most accurate and confident method, similar to Table 5.1. In all three cases, BD is outperformed by both multinomial sampling and DBD. Multinomial sampling and DBD both show great potential and have almost equal performance, with DBD performing best when using the most diverse diversity penalty. However, multinomial sampling has the least runtime, making it the most computationally efficient. Given its similar performance to DBD, which has a significantly longer runtime, multinomial sampling appears to be the more feasible option.

Nonetheless, all methods perform well, achieving higher than 80% accuracy in all cases, indicating that all three decoding methods are capable of effectively playing and solving easy reference games.

Performance with 5 samples

Decoding method	**Accuracy**	**Entropy**	**Runtime (S)**
Multinomial Sampling	0.9	0.4321	3.310
BD (B: 5)	0.85	0.7858	9.860
DBD (G: 5, γ: 0.2)	0.87	0.8565	35.310
DBD (G: 5, γ: 0.8)	0.84	0.6688	34.180
DBD (G: 5, γ: 0.01)	0.89	0.6647	29.230

Table 5.1: Performance metrics with 5 samples. For 100 easy difficulty games with an utterance length of 10. In this scenario, multinomial sampling achieves the highest accuracy score, it is also the most confident choice based on the lowest entropy score. DBD with a diversity penalty of 0.01 comes close to similar results but takes alot longer to compute.

Performance with 10 samples

Decoding method	Accuracy	Entropy	Runtime (S)
Multinomial Sampling	0.89	0.4296	3.430
BD (B: 10)	0.89	0.9939	27.180
DBD (G: 2, γ: 0.2)	0.85	0.9776	44.117
DBD (G: 5, γ: 0.2)	0.87	1.1257	96.184
DBD (G: 10, γ: 0.2)	0.86	1.1132	176.020
DBD (G: 2, γ: 0.8)	0.85	0.8193	45.602
DBD (G: 5, γ: 0.8)	0.86	0.7829	95.842
DBD (G: 10, γ: 0.8)	0.87	0.7592	180.170
DBD (G: 2, γ: 0.01)	0.87	0.9123	44.030
DBD (G: 5, γ: 0.01)	0.85	0.9070	94.309
DBD (G: 10, γ: 0.01)	0.91	0.8500	183.030

Table 5.2: Performance metrics with 10 samples. For 100 easy difficulty games with an utterance length of 10. In this scenario, DBD outperforms multinomial sampling when applied with a diversity penalty of 0.01 and an individual group for every sample utterance. However, multinomial sampling still is faster to compute and a more confident choice based on the entropy.

Performance with 25 samples

Decoding method	Accuracy	Entropy	Runtime (S)
Multinomial Sampling	0.95	0.4152	4.560
BD (B: 25)	0.87	1.1251	94.220
DBD (G: 2, γ: 0.2)	0.83	1.1269	190.017
DBD (G: 5, γ: 0.2)	0.85	1.1863	398.587
DBD (G: 25, γ: 0.2)	0.87	1.1891	1708.790
DBD (G: 2, γ: 0.8)	0.88	0.9680	169.327
DBD (G: 5, γ: 0.8)	0.82	0.9253	367.800
DBD (G: 25, γ: 0.8)	0.87	0.9143	1810.460
DBD (G: 2, γ: 0.01)	0.84	1.0637	163.675
DBD (G: 5, γ: 0.01)	0.86	1.0420	358.173
DBD (G: 25, γ: 0.01)	0.91	1.0672	1795.140

Table 5.3: Performance metrics with 25 samples. For 100 easy difficulty games with an utterance length of 10. In this scenario, multinomial sampling outperforms the other methods with at least a 4% higher accuracy while also being the fastest to compute. DBD is the second best method when implemented with an individual group for each sample utterance. However, it takes significantly longer to compute compared to multinomial sampling.

5.1.2 Hard Reference Games

In this section, the hard reference games are evaluated based on accuracy, entropy, and runtime. The tables all represent the same decoding methods performed with different numbers of samples. It is clear that accuracy values during hard games are considerably lower than during easy games, this shows the need for better pragmatic reasoning to solve harder difficulty games. These lower accuracy scores also suggest that there is still room for improvement in the pragmatic speaker. An interesting change compared to the easy reference games is that in all three cases 5.4, 5.5, and 5.6 DBD achieves higher accuracy than multinomial sampling, which was not the case for the easy games. However, multinomial sampling still consistently shows the lowest entropy values, indicating higher confidence in all cases. Another difference from the easy reference games is that a diversity penalty of 0.01 was not always optimal for DBD, a diversity penalty of 0.2 achieved better accuracy in 5.4.

The runtime of multinomial sampling still remains significantly lower than the other methods, making it the most computationally efficient for hard games as well. Traditional BD is still outperformed by the other decoding methods.

Performance with 5 samples

Decoding method	**Accuracy**	**Entropy**	**Runtime (S)**
Multinomial Sampling	0.49	0.4749	3.172
BD (B: 5)	0.45	0.8391	10.809
DBD (G: 5, γ: 0.2)	0.50	0.9773	35.792
DBD (G: 5, γ: 0.8)	0.45	0.6740	32.826
DBD (G: 5, γ: 0.01)	0.46	0.8578	32.927

Table 5.4: Performance metrics with 5 samples. For 100 hard difficulty games with an utterance length of 10. In this scenario, DBD outperforms multinomial sampling, which was not the case in 5.1. Another interesting observation is that a diversity penalty of 0.2 outperforms a diversity penalty of 0.01 which also was not the case in 5.1. However multinomial sampling still is the fastest to compute being at least 10 times quicker than any DBD method.

Performance with 10 samples

Decoding method	Accuracy	Entropy	Runtime (S)
Multinomial Sampling	0.51	0.4934	3.823
BD (B: 10)	0.46	0.9441	27.304
DBD (G: 2, γ: 0.2)	0.46	1.1481	42.776
DBD (G: 5, γ: 0.2)	0.51	1.0478	94.363
DBD (G: 10, γ: 0.2)	0.48	1.2354	184.092
DBD (G: 2, γ: 0.8)	0.44	0.8711	45.527
DBD (G: 5, γ: 0.8)	0.47	0.8127	90.533
DBD (G: 10, γ: 0.8)	0.48	0.8121	182.154
DBD (G: 2, γ: 0.01)	0.42	1.0848	40.563
DBD (G: 5, γ: 0.01)	0.52	1.0883	92.760
DBD (G: 10, γ: 0.01)	0.49	1.0109	170.902

Table 5.5: Performance metrics with 10 samples. For 100 hard difficulty games with an utterance length of 10. In this scenario, DBD with a diversity penalty of 0.01 and 2 samples per group is the most accurate method. This differs from the most accurate method in 5.2 which had the same diversity penalty but more groups. Multinomial sampling still achieves close to equal results and is the most confident choice based on the entropy values.

Performance with 25 samples

Decoding method	Accuracy	Entropy	Runtime (S)
Multinomial Sampling	0.51	0.5447	4.027
BD (B: 25)	0.50	1.1266	97.932
DBD (G: 2, γ: 0.2)	0.50	1.2351	175.340
DBD (G: 5, γ: 0.2)	0.51	1.3643	358.602
DBD (G: 25, γ: 0.2)	0.50	1.4106	1747.980
DBD (G: 2, γ: 0.8)	0.46	1.0093	161.418
DBD (G: 5, γ: 0.8)	0.45	0.9781	357.234
DBD (G: 25, γ: 0.8)	0.42	0.9161	1663.919
DBD (G: 2, γ: 0.01)	0.48	1.2066	170.002
DBD (G: 5, γ: 0.01)	0.43	1.1641	343.541
DBD (G: 25, γ: 0.01)	0.52	1.3280	1603.468

Table 5.6: Performance metrics with 25 samples. For 100 hard difficulty games with an utterance length of 10. In this scenario, multinomial sampling performs worse than the DBD method in accuracy which differs from the results show in in 5.3. However, multinomial sampling still is the most confident choice and is far more efficient to compute. Which makes it a good alternative of the DBD method.

5.2 Diversity

5.2.1 Easy Reference Games

The following results evaluate the diversity in the easy reference games, comparing different distinct values of n-grams that indicate the diversity in vocabulary used for sampled utterances. Throughout all results in Tables 5.7, 5.8 and 5.9 DBD with a diversity penalty of 0.01 consistently produces the most diverse sample utterances. Multinomial Sampling produces an equal or greater amount of diverse utterances compared to the traditional BD method. When comparing diversity with accuracy, as shown in Tables 5.1, 5.2 and 5.3, it becomes clear that the diversity in sampled utterances impacts accuracy. An example of this is that methods that outshine BD in terms of accuracy also show greater diversity than BD.

Distinct n-grams with 5 samples

Decoding method	**Normalized**			**Absolute**		
	n=1	**n=2**	**n=3**	**n=1**	**n=2**	**n=3**
Multinomial Sampling	0.4382	0.5838	0.5810	1942	2588	2576
BD (B: 5)	0.3981	0.4726	0.4571	2017	2397	2319
DBD (G: 5, γ: 0.2)	0.3221	0.3800	0.3644	1668	1968	1888
DBD (G: 5, γ: 0.8)	0.2335	0.2568	0.2439	1222	1343	1275
DBD (G: 5, γ: 0.01)	0.4935	0.6192	0.5880	2527	3173	3015

Table 5.7: Distinct n-grams with 5 samples. For 100 easy difficulty games with an utterance length of 10. In this scenario, it is clear to see that a diversity penalty of 0.01 produces the most diverse samples, with multinomial sampling being the second most diverse method. BD produces more diverse samples than DBD when using a diversity penalty that is not 0.01. This suggests that for obtaining diverse samples, values lower than 0.2 should be used, as traditional BD will otherwise generate more diverse samples.

Distinct n-grams with 10 samples

Decoding method	**Normalized**			**Absolute**		
	n=1	**n=2**	**n=3**	**n=1**	**n=2**	**n=3**
Multinomial Sampling	0.3089	0.4561	0.4784	2737	4042	4240
BD (B: 10)	0.3194	0.4116	0.4115	3183	4102	4101
DBD (G: 2, γ: 0.2)	0.2995	0.3881	0.3868	3065	3972	3959
DBD (G: 5, γ: 0.2)	0.2862	0.3630	0.3598	2963	3758	3724
DBD (G: 10, γ: 0.2)	0.2713	0.3488	0.3464	2823	3629	3605
DBD (G: 2, γ: 0.8)	0.1690	0.1991	0.1942	1756	2069	2018
DBD (G: 5, γ: 0.8)	0.1483	0.1729	0.1690	1564	1821	1779
DBD (G: 10, γ: 0.8)	0.1595	0.1896	0.1867	1681	1997	1967
DBD (G: 2, γ: 0.01)	0.3373	0.4444	0.4439	3440	4534	4529
DBD (G: 5, γ: 0.01)	0.3948	0.5303	0.5192	4054	5448	5334
DBD (G: 10, γ: 0.01)	0.3921	0.5442	0.5343	4067	5645	5543

Table 5.8: Distinct n-grams with 10 samples. For 100 easy difficulty games with an utterance length of 10. This scenario shows the same pattern as 5.7. DBD produces the most diverse samples with a diversity penalty of 0.01. A new pattern is shown, where a larger group value results in more diverse samples, this correlates with the findings in (Vijayakumar 2016).

Distinct n-grams with 25 samples

Decoding method	**Normalized**			**Absolute**		
	n=1	**n=2**	**n=3**	**n=1**	**n=2**	**n=3**
Multinomial Sampling	0.1856	0.3208	0.3679	4111	7104	8150
BD (B: 25)	0.2271	0.3318	0.3422	5669	8285	8544
DBD (G: 2, γ: 0.2)	0.2195	0.3344	0.3492	5940	9049	9449
DBD (G: 5, γ: 0.2)	0.2276	0.3511	0.3658	5935	9149	9532
DBD (G: 25, γ: 0.2)	0.1943	0.2871	0.3002	5159	7619	7968
DBD (G: 2, γ: 0.8)	0.1237	0.1642	0.1670	3376	4477	4555
DBD (G: 5, γ: 0.8)	0.0983	0.1286	0.1310	2595	3392	3453
DBD (G: 25, γ: 0.8)	0.1054	0.1414	0.1459	2798	3752	3872
DBD (G: 2, γ: 0.01)	0.2359	0.3653	0.3814	6369	9860	10292
DBD (G: 5, γ: 0.01)	0.2664	0.4210	0.4334	6938	10960	11282
DBD (G: 25, γ: 0.01)	0.3090	0.5181	0.5326	8157	13674	14057

Table 5.9: Distinct n-grams with 25 samples. For 100 easy difficulty games with an utterance length of 10. In this scenario the same pattern is present as in 5.7 and 5.8. DBD with a diversity penalty of 0.01 still produces the most diverse utterances. However, the results also show that the normalized uni-gram values are far lower for all decoding methods than in 5.7 and 5.8. This could imply that using 25 samples does not necessarily result in more diverse samples compared to using 10 samples.

5.2.2 Hard Reference Games

The following results evaluate the diversity in the hard reference games. Similar to the easy games, the diversity results for the hard reference games show that DBD with a diversity penalty of 0.01 produces the most diverse utterances, with multinomial sampling being the second most diverse method. The consistency in diversity with accuracy in the hard games aligns with the findings in the easy games.

An interesting observation when comparing the hard and easy results is that the diversity of n-grams remains quite consistent across decoding methods in both cases, despite the significant differences in accuracy between the game difficulties.

Distinct n-grams with 5 samples

Decoding method	**Normalized**			**Absolute**		
	n=1	**n=2**	**n=3**	**n=1**	**n=2**	**n=3**
Multinomial Sampling	0.4670	0.6187	0.6124	2073	2746	2718
BD (B: 5)	0.3988	0.4735	0.4580	2045	2428	2348
DBD (G: 5, γ: 0.2)	0.3390	0.4042	0.3896	1773	2114	2038
DBD (G: 5, γ: 0.8)	0.2285	0.2525	0.2406	1222	1349	1285
DBD (G: 5, γ: 0.01)	0.5008	0.6261	0.5948	2553	3196	3038

Table 5.10: Distinct n-grams with 5 samples. For 100 hard difficulty games with an utterance length of 10. In this scenario DBD produces the most diverse sample utterances with a diversity penalty of 0.01. However multinomial sampling does show higher normalized bi-gram and tri-gram values. This is likely due to the random aspect of the multinomial distribution, which occasionally results in non-fluent utterances, unlike the fluent decoding of DBD.

Distinct n-grams with 10 samples

Decoding method	**Normalized**			**Absolute**		
	n=1	**n=2**	**n=3**	**n=1**	**n=2**	**n=3**
Multinomial Sampling	0.3208	0.4794	0.5008	2843	4248	4439
BD (B: 10)	0.3109	0.4014	0.4014	3131	4037	4036
DBD (G: 2, γ: 0.2)	0.2997	0.3897	0.3897	3083	4007	4008
DBD (G: 5, γ: 0.2)	0.2811	0.3592	0.3576	2931	3746	3729
DBD (G: 10, γ: 0.2)	0.2704	0.3473	0.3463	2834	3638	3627
DBD (G: 2, γ: 0.8)	0.1739	0.2040	0.1986	1811	2123	2066
DBD (G: 5, γ: 0.8)	0.1585	0.1865	0.1836	1674	1969	1938
DBD (G: 10, γ: 0.8)	0.1543	0.1822	0.1801	1647	1943	1920
DBD (G: 2, γ: 0.01)	0.3355	0.4386	0.4389	3429	4481	4485
DBD (G: 5, γ: 0.01)	0.3970	0.5296	0.5164	4087	5453	5318
DBD (G: 10, γ: 0.01)	0.3980	0.5461	0.5327	4122	5657	5520

Table 5.11: Distinct n-grams with 10 samples. For 100 hard difficulty games with an utterance length of 10. In this scenario the same pattern as in 5.8 and 5.10 is shown. DBD produces the most diverse utterances with a diversity penalty of 0.01. A difference compared to 5.10 is that the normalized tri-gram values of DBD with a diversity penalty of 0.01 and a group size of 5 and 10 are greater than the multinomial sampling tri-gram values.

Distinct n-grams with 25 samples

Decoding method	Normalized			Absolute		
	n=1	**n=2**	**n=3**	**n=1**	**n=2**	**n=3**
Multinomial Sampling	0.1877	0.3238	0.3697	4170	7190	8208
BD (B: 25)	0.2226	0.3267	0.3380	5557	8147	8425
DBD (G: 2, γ: 0.2)	0.2190	0.3338	0.3492	5904	8991	9406
DBD (G: 5, γ: 0.2)	0.2269	0.3481	0.3621	5925	9087	9450
DBD (G: 25, γ: 0.2)	0.1975	0.2924	0.3072	5239	7755	8148
DBD (G: 2, γ: 0.8)	0.1277	0.1713	0.1754	3481	4664	4774
DBD (G: 5, γ: 0.8)	0.1043	0.1377	0.1402	2762	3642	3706
DBD (G: 25, γ: 0.8)	0.1030	0.1379	0.1431	2750	3683	3819
DBD (G: 2, γ: 0.01)	0.2322	0.3540	0.3709	6256	9536	9991
DBD (G: 5, γ: 0.01)	0.2677	0.4229	0.4356	6965	10998	11331
DBD (G: 25, γ: 0.01)	0.3066	0.5093	0.5244	8073	13411	13808

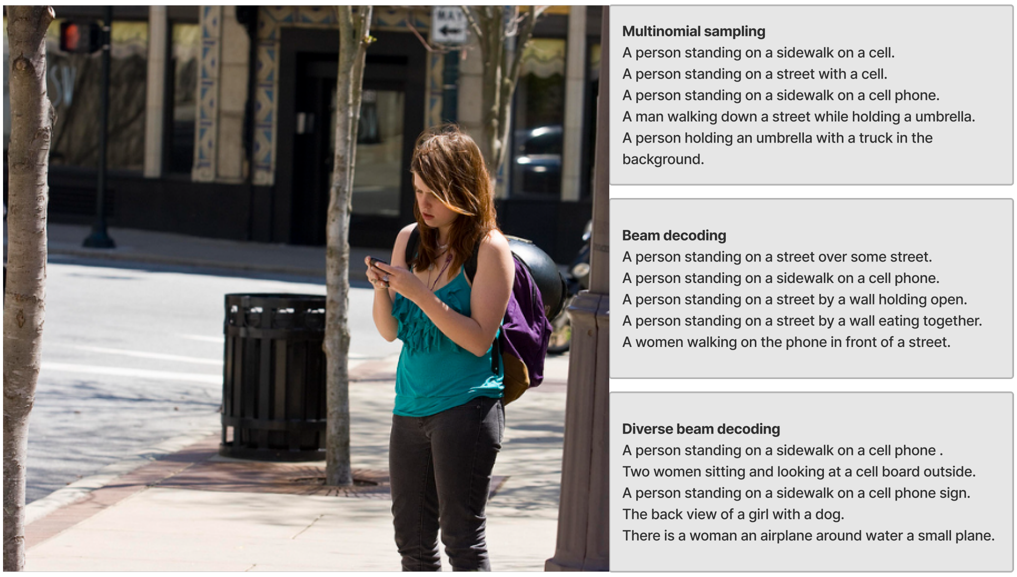
Table 5.12: Distinct n-grams with 25 samples. For 100 hard difficulty games with an utterance length of 10. In this scenario, the same pattern as in 5.12 is shown, where the values of the normalized distinct n-grams is lower than the values in the table with 10 samples 5.11. This again implies that using 25 samples does not necessarily result in more diverse samples when compared to 10 samples.

5.3 Qualitative Results

This section will evaluate the quality of the generated utterances. Since there are no implemented metrics in this paper for this type of evaluation, as their implementation was out of scope for this study, the assessment relies on our own human perception. The utterances shown in figure 5.1 and figure 5.2 are drawn from the same generated batch that corresponds to the other results in this paper, the DBD utterances are based on DBD with a diversity penalty of 0.01 and separate group for each sample utterance.

Both figures 5.1 and 5.2 demonstrate that multinomial sampling and beam decoding generate similar sample utterances, suggesting that generating more samples with these methods could be computationally wasteful as mentioned in (Vijayakumar 2016). Diverse beam decoding produces diverse samples but sometimes loses track of the target referent, resulting in diverse utterances that no longer accurately describe the target. For example, the utterance "There is a women an airplane around water a small plane" shown in figure 5.1 generated by DBD, displays this issue as such utterances do not contribute to solving the reference game. The multinomial results also include non-fluent utterances, likely due to the randomness introduced by the multinomial distribution when selecting the next word of an utterance. This fluency issue is shown in example utterances such as, "A baseball player holding child playing with a frisbee".

Overall, when examining the generated utterances from all three decoding methods, they generally describe the target image but lack the pragmatic reasoning necessary when reference games involve very similar distractors. Specific objects from the target and color-based descriptions are often missing. Therefore, it is understandable why the hard reference game results, as shown in Tables 5.10, 5.11, and 5.12, display lower accuracy compared to the results from the easy reference games.

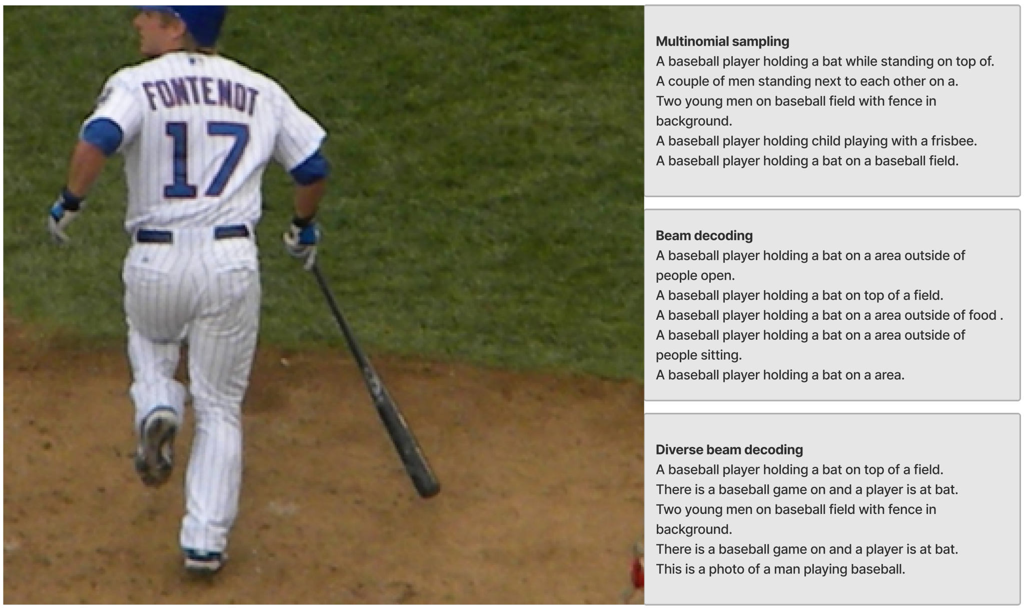


Multinomial sampling
A person standing on a sidewalk on a cell.
A person standing on a street with a cell.
A person standing on a sidewalk on a cell phone.
A man walking down a street while holding a umbrella.
A person holding an umbrella with a truck in the background.

Beam decoding
A person standing on a street over some street.
A person standing on a sidewalk on a cell phone.
A person standing on a street by a wall holding open.
A person standing on a street by a wall eating together.
A women walking on the phone in front of a street.

Diverse beam decoding
A person standing on a sidewalk on a cell phone .
Two women sitting and looking at a cell board outside.
A person standing on a sidewalk on a cell phone sign.
The back view of a girl with a dog.
There is a woman an airplane around water a small plane.

Figure 5.1: Utterances generated to describe an image of a girl using a cell phone with the three implemented decoding methods. Multinomial sampling generates grammatical mistakes due to the randomness in its distribution. Beam decoding seems to produce non-diverse utterances, while diverse beam decoding sometimes loses focus on the target due to extreme diversity



Multinomial sampling
A baseball player holding a bat while standing on top of.
A couple of men standing next to each other on a.
Two young men on baseball field with fence in background.
A baseball player holding child playing with a frisbee.
A baseball player holding a bat on a baseball field.

Beam decoding
A baseball player holding a bat on a area outside of people open.
A baseball player holding a bat on top of a field.
A baseball player holding a bat on a area outside of food .
A baseball player holding a bat on a area outside of people sitting.
A baseball player holding a bat on a area.

Diverse beam decoding
A baseball player holding a bat on top of a field.
There is a baseball game on and a player is at bat.
Two young men on baseball field with fence in background.
There is a baseball game on and a player is at bat.
This is a photo of a man playing baseball.

Figure 5.2: Utterances generated to describe an image of a running baseball player with the three implemented decoding methods. Multinomial sampling generates grammatical mistakes due to the randomness in its distribution. Beam decoding seems to produce non-diverse utterances, while diverse beam decoding sometimes loses focus on the target due to extreme diversity.

Chapter 6

Discussion and Conclusion

This chapter will discuss and conclude the results presented in chapter 5. The discussion will primarily focus on the findings and limitations of this study, and will offer recommendations for future research to address these limitations. The conclusion will provide a final reflection on the study as a whole.

6.1 Discussion

The results in chapter 5 demonstrate that diversity considerably impacts pragmatic reasoning within a reference game setting. Traditional beam decoding is shown to be less optimal compared to a more diverse variant such as diverse beam decoding. Surprisingly, even a simple method like multinomial sampling outperforms beam decoding in many areas, including accuracy and confidence.

Diverse beam decoding and multinomial sampling do show near equal performance. However, the diverse beam decoding method requires significantly more computation time compared to multinomial sampling, which raises questions about its feasibility when accuracy improvements are minimal. Diverse beam decoding only slightly outperforms multinomial sampling in harder reference games, and the difference is less or equal to 1%. Nevertheless, the results indicate that diverse beam decoding is a promising method, combining the fluency of beam decoding with the diversity of multinomial sampling. With further adjustments, it could potentially achieve even better results.

Due to the limitations of this paper, we could not measure the fluency of the generated sample utterances thoroughly. If diverse beam decoding proves to be significantly more fluent, it could outperform multinomial sampling by a greater distance, providing better pragmatic descriptions of the target referent. Further research is needed to explore different pruning methods within the diverse beam decoding algorithm. While this study used top-B beam selection, other techniques like nucleus or multinomial beam selection within the diverse beam decoding algorithm could produce different results. Additionally, experimenting with other parameters, such as altering the temperature and maximum sequence length, could provide further changes in results.

6.2 Conclusion

Diverse beam search improves traditional beam search, but it consistently falls behind multinomial sampling in performance due to significantly longer runtimes, which raises questions about its computational feasibility. Research indicates that diverse beam search improves fluency, showing promising potential. Since, fluency scoring was not feasible within the scope of this study, the exact impact of it could not be measured.

Both diverse beam search and multinomial sampling demonstrate greater diversity compared to traditional beam decoding, highlighting that their improved diversity contributes to higher accuracy in referential games.

In conclusion, while diverse beam decoding offers advantages in terms of fluency and diversity, its practical implementation is restricted by computational inefficiencies compared to multinomial sampling. Future research should explore optimizations to resolve these trade-offs and further evaluate its impact on pragmatic reasoning in reference games. Additionally, efforts to achieve higher accuracy with diverse beam decoding would justify its computational costs, which currently limit its feasibility.

Bibliography

Andreas, Jacob and Dan Klein (Nov. 2016). “Reasoning about Pragmatics with Neural Listeners and Speakers”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, pp. 1173–1182. DOI: 10.18653/v1/D16-1125. URL: https://aclanthology.org/D16-1125.

Degen, Judith (2023). “The rational speech act framework”. In: *Annual Review of Linguistics* 9, pp. 519–540.

Frank, Michael C and Noah D Goodman (2012). “Predicting pragmatic reasoning in language games”. In: *Science* 336.6084, pp. 998–998.

Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy (2014). “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572*.

Goodman, Noah D. and Michael C. Frank (2016). “Pragmatic Language Interpretation as Probabilistic Inference”. In: *Trends in Cognitive Sciences* 20.11, pp. 818–829. ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2016.08.005. URL: https://www.sciencedirect.com/science/article/pii/S136466131630122X.

He, Kaiming et al. (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Lewis, David Kellogg (1969). *Convention: A Philosophical Study*. Cambridge, MA, USA: Wiley-Blackwell.

Lin, Tsung-Yi et al. (2014). “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, pp. 740–755. ISBN: 978-3-319-10602-1.

Liu, Andy et al. (2023). “Computational language acquisition with theory of mind”. In: *arXiv preprint arXiv:2303.01502*.

Liu, Yinhan et al. (2019). “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692*.

Reimers, Nils and Iryna Gurevych (2019). “Sentence-bert: Sentence embeddings using siamese bert-networks”. In: *arXiv preprint arXiv:1908.10084*.

Vijayakumar, A. K. (2016). “Diverse beam search: Decoding diverse solutions from neural sequence models”. In: *arXiv preprint arXiv:1610.02424*.

White, J. (2020). “Learning to refer informatively by amortizing pragmatic reasoning”. In: *arXiv preprint arXiv:2006.00418*.