Master Thesis

Assessing the Role of Gesture Information in NLG: a Case Study with LLMs and Multimodal AMR

Sanne van den Berg

Supervisor	Lucia Donatelli
2nd reader	Antske Fokkens

a thesis submitted in fulfillment of the requirements for the degree of

MA Linguistics

(Text Mining)

Vrije Universiteit Amsterdam

Computational Linguistics and Text-Mining Lab
Department of Language and Communication
Faculty of Humanities

Date	June 27, 2025
Student number	2672693
Word count	16181

Abstract

Abstract Meaning Representation (AMR) is a notation scheme developed to encode the meaning of English sentences in written text and has gained popularity because of its easy-to-understand structure. AMRs are represented as rooted, directed, labeled graphs, where concepts are represented by nodes, relations between nodes are represented as edges, and coreferences by recurrent nodes. Originally, it was designed for written English sentences, and recent research has adapted the AMR structure to attempt to represent the semantics of gesture. Given that gestures often convey meaningful information alongside speech, this thesis aims to investigate whether large language models (LLMs) can interpret and use the information provided by the gestures. To explore this, the SAGE-AMR dataset was created and used to prompt the Llama model. The input consists of three types: speech AMR only, gesture information, and a combination of both. The model is then tasked with generating a corresponding sentence to assess what impact gesture information has on the model's ability to interpret this gesture information and generate sentences using it. Results indicate that Llama struggles to extract and integrate information from both the speech AMR and gesture information, often failing to produce accurate or complete outputs. These findings suggest limitations in the model's ability to interpret gesture-based semantics within this framework and highlight the possibility that unimodal models cannot grasp gesture meaning. The code for this thesis can be found on Github[1].

[1] https://github.com/sanne-jpg/Master-thesis

Declaration of Authorship

I, author, declare that this thesis, titled *Assessing the Role of Gesture Information in NLG: a Case Study with LLMs and Multimodal AMR* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: June 27th 2025

Signed: Sanne van den Berg

Acknowledgements

First and foremost, I would like to thank my supervisor and mentor Lucia Donatelli for her continuous guidance and support throughout this thesis, as well as the rest of the master. It has been a long and tough year, with a few too many late nights at the VU, and this thesis would not have been possible without your expertise on the topic, so thank you.

I would also like to thank the rest of the CLTL staff for teaching us and sharing your knowledge with us. It has been inspiring to see your enthusiasm. Specifically, I would like to thank Luís de Passos Morgado da Costa for providing us with funny references to pandas in almost every lecture, it really put a smile on my face.

I also want to thank my classmates, specifically Elisabetta Dentico, Hannah Goossens, Ino van de Wou, and Wayne Kuan, for their support the past year. You guys made the lectures and study sessions a little bit better.

Lastly, I want to thank my boyfriend Vince and my cat Nibbit, they really helped me during these past months by keeping me motivated, and Nibbit provided the needed distractions, reminding me it is okay to take a break sometimes.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Overview

Human communication is inherently multimodal. While spoken language serves as the primal form of conveying information, it is often accompanied by non-verbal cues. These cues often consist of head or hand movements, as well as facial expressions. They help to disambiguate meaning or to emphasize intention (McNeill and Duncan (2000)). These gestures help enhance the semantic richness of spoken language. Despite the richness of gesture in everyday communication, most existing Natural Language Processing (NLP) frameworks focus on text or speech, not taking into account how gestures contribute to meaning. Even in semantic representation frameworks like Abstract Meaning Representation (AMR), gesture was not incorporated until recently. AMR is a notation scheme developed to encode the meaning of linguistic expressions, specifically written English sentences, and will be further explained in chapter 2. Recent studies have extended AMR with gesture annotations, Gesture AMR, but these representations remain underutilized in large language models (LLMs). LLMs will also be discussed in chapter 2, and a detailed explanation of Gesture AMR can be found in chapter 3. Although models like Llama have shown remarkable capabilities in generating language, their training data often consists only of linguistic input, such as text and code from various sources. While Gesture AMR provides a mechanism for representing gestures formally, it remains unclear whether incorporating such representations into LLM prompts can improve the quality or grounding of generated text. This raises the question whether gesture information can improve a large language model's ability to understand or generate language. Moreover, can gestures provide disambiguating context that enhances the output quality when asked to generate natural spoken sentences? This thesis tries to fill this gap.

1.2 Goal and Research Questions

This thesis investigates to what extent gesture information improves Natural Language Generation (NLG) in large language models. The gesture information is represented through gesture labels that describe the intention of a given gesture and gesture AMR. I explore whether the inclusion of gesture semantics can help the model in generating more contextually grounded and meaningful sentences, particularly in multimodal communication scenarios. In doing so, I aim to shed light on the potential and limitations of integrating gesture semantics into modern LLMs.

The research question of this thesis is to determine how gesture information impacts the model's ability to perform natural language generation, compared to only speech as input. This question can be broken down into sub-questions:

- Do gesture labels and gesture AMR help predict speech and speech AMR?
- Does speech AMR help predict speech?

Specifically, a Llama model is prompted with three different inputs: speech AMR only, gesture information only, and a combination of both. The main contributions of this thesis are summarized below:

- Aligned multimodal data including speech transcripts, speech AMR, gesture labels and gesture AMR.
- Created a new dataset, SAGE-AMR, combining speech and gesture information for AMR-based tasks.
- Designed and conducted prompting experiments with a Llama model using the SAGE-AMR dataset.
- Performed a detailed error analysis of Llama's outputs to refine prompt strategies and categorize common model mistakes.

1.3 Thesis Outline

Before outlining the methodology used to address the research question, chapter 2 provides an overview of the relative literature and essential background concepts. This includes information on AMR and its framework, gesture in communication, large language models, and common prompting techniques. To investigate whether gesture information impacts the model's ability to perform natural language generation, the multimodal dataset SAGE-AMR was created, which will be discussed in chapter 3. It was created using an existing gesture corpus, which is also explained in this chapter. The three conditions and the different prompts created per prompt are explained in chapter 4, as well as the specific Llama model used for the task and its specific parameters. The results of the three conditions are explained in chapter 5. To better understand the model output, an error analysis was conducted to quantify the error types and gain deeper insight into the specific challenges faced by the model. This analysis is explained in chapter 6. Chapter 7 discusses the results and other insights gained throughout this thesis, as well as limitations. Lastly, chapter 8 draws a conclusion, and the Appendix contains supplementary material that supports the main content but is not critical to grasping the core arguments.

Chapter 2

Background and Related Work

This chapter provides an overview of the relevant literature related to the topics in this thesis and essential background concepts. This includes information on Abstract Meaning Representation (AMR) and its framework, gesture in communication, Large Language Models (LLM), and common prompting techniques. The focus of this thesis is on the use of language models to predict sentences based on AMR. Therefore, section 2.1 explains Standard AMR in detail, including its framework. Section 2.2 discusses gesture and some foundational gesture research, section 2.3 discusses (large) language models and popular architectures, steps that are needed to train a large language model, in addition to the most popular prompting techniques. Lastly, section 2.4 discusses automatic evaluation metrics, as well as AMR-to-text generation.

2.1 Abstract Meaning Representation

This section introduces Abstract Meaning Representation (AMR), a semantic representation framework designed to capture the meaning of written English sentences in a structured format. Understanding this format is essential for this thesis, since both the speech and gesture data are annotated using this formalism. This section outlines the motivation behind the development of AMR, and explains its underlying framework.

Abstract Meaning Representation (AMR), as described by Banarescu et al. (2013), is a notation scheme that has been developed to encode the meaning of English expressions. The AMR notation is easy for people to read and understand, and easy for computer programs to parse. This notation focuses on predicate-argument structure, and makes extensive use of PropBank framesets (Kingsbury and Palmer (2002)). Verbs can have multiple meanings, and these framesets help to distinguish between all possible meanings. An example is shown in Figure 2.1. Here, the frameset `look.02` is used, which means “seeming, appear/seem”. This choice is contextually appropriate because *look* does not denote visual perception, but rather the appearance of a situation. Compare this to `look.01` in Figure 2.1b, which refers to the visual sense of *look*. This difference is crucial because a misinterpretation of the verb sense would result in an incorrect interpretation, especially in NLG tasks. Chapter 6 will showcase an example of how the model handled these different framesets. Accurately resolving such ambiguities is particularly important for the task in this thesis, where gesture information might influence or help clarify verb sense.

```
(l/look-02
    :ARG1 (b/block
        :mod (n/next))
    :ARG2 (b2/be-located-at-91
        :ARG1 b
        :ARG2 (o/on
            :op1 (t/top
                :op1 (t2/that)))
        :ARG1-of (d/direct-02
            :mod (a/almost)))
    :time (t3/then))
```

(a) AMR example.

look.01 - *vision, look*

(b) Frameset definition look.01.

look.02 - *seeming, appear/seem*

(c) Frameset definition look.02.

Figure 2.1: AMR example for the sentence "Looks like the the next block is almost directly on top of that". The two framesets of the verb "look" help to disambiguate the verb's meaning.

The AMR notation was originally designed to encourage new work and research in statistical Natural Language Understanding (NLU) and NLG (Banarescu et al. (2013)). Because of its easy annotation style and comprehensibility, much research has been done on how it can be applied to different domains within different Natural Language Processing (NLP) tasks. This will be discussed in greater detail in the next section, following an overview of the AMR framework.

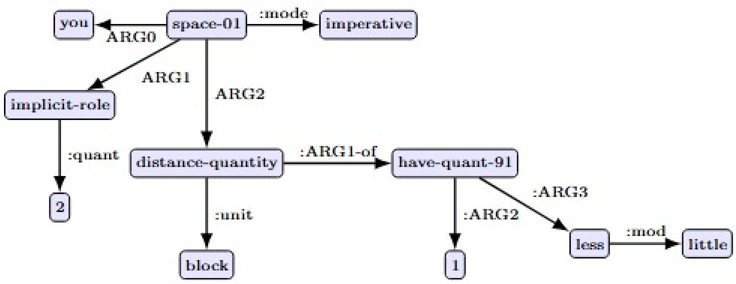
2.1.1 AMR Framework

AMRs are written down as rooted, directed, labeled graphs. There are multiple ways in which one can write down AMR: logical notation, PENMAN notation (Matthiessen and Bateman (1991)), and graph notation. In such a graph, nodes represent entities, properties, or events; leaves are labelled with concepts; and relations link entities together to represent sentences. As described above, AMR annotation makes use of PropBank framesets, which can be seen in the concept names. These are either PropBank framesets such as `put.01` or words, such as "block". PropBank framesets help to disambiguate verbs. They provide a description of the particular verb, and all possible semantic role labels it can take. A number of relations are used, ranging from: arguments (:ARG0:), general semantic relations (:location:), quantities (:quant:), date-entities (:day:), and lists (:op1:). Table 2.1 shows an example of the frameset `put.01` and `grab.01`. These examples are shown because they often appear in the corpus used for the task.

put.01 - *location*	grab.01 - *to capture, obtain, taking hold of*
ARG0: putter	ARG0: grabber
ARG1: thing put	ARG1: entity grabbed
ARG2: where put	

Table 2.1: Table with framesets for put.01 and grab.01.

Figure 2.2 shows an example of the three different notations for the sentence "Space two out a little less than a block length". Only the PENMAN notation will be used in this thesis, as it is easy to read and the format adopted by the underlying corpus on which this thesis is based. All three formats convey the same underlying semantic structure. This comparison serves to contextualize the choice of the PENMAN notation



(a) Graph notation.

```
(s / space-01
  :mode imperative
  :ARG0 (y / you)
  :ARG1 (i / implicit-role
         :quant 2)
  :ARG2 (q / distance-quantity
         :unit (b / block)
         :ARG1-of (h / have-quant-91
                    :ARG2 1
                    :ARG3 (l / less
                           :mod (l2 / little)))))
```

(b) PENMAN notation.

```
∃ s, x, d:
  instance(s, space-out-01) ∧ instance(x, two) ∧
  instance(d, distance) ∧ arg0(s, you) ∧
  arg1(s, x) ∧ arg2(s, d) ∧ modifier(d, less-than) ∧
  quantity(d, block-length) ∧ degree(d, little)ttle)))))
```

(c) Logic notation.

Figure 2.2: Different AMR notations for "Space two out a little less than a block length."

as both accessible and expressive for representing meaning in natural language and gesture. Moreover, because the AMR notation abstracts away from surface syntax, with the goal to assign the same AMR to sentences with the same meaning, there are multiple possible sentences for one AMR graph. For example, the sentences "She called him a liar.", "Her accusation toward him: liar." and "According to her, he was a liar." would all have the same AMR representation. This creates challenges for the evaluation of the model's output, as generated sentences might be semantically correct but differ from the reference sentence in wording or structure.

```
(p / push-01
  :mode imperative
  :ARG0 (y / you)
  :ARG1 (b / block
         :mod (t / that))
  :ARG2 (l / left))
```

(a) Speech AMR for "push that block left".

```
(i / icon-GA
  :ARG0 (s / signaler)
  :ARG1 (p / push-01
          :direction (l / left))
  :ARG2 (a / actor))
```

(b) "Push left" gesture.

Figure 2.3: The speech and gesture AMR associated with the communicative act of pushing a block to the left. The colors indicate coreference relations.

As mentioned previously, much research has been done on how the AMR notation could be applied to different domains within different NLP tasks. O'Gorman et al. (2018) created a multi-sentence AMR (MS-AMR) corpus, because when a document needs to be analyzed, it is important to access information that spans multiple sentences, such as coreference. In the MS-AMR corpus, existing AMRs were annotated with information about coreference, implicit roles, and information that bridges relations across multiple AMRs. This is relevant for this thesis because it allows for the establishment of connections between the content of speech and gesture modalities. An example showing the use of MS-AMR for this task can be found in Figure 2.3, adapted from Lai et al. (2024). These links allow for the alignment of gesture components with corresponding elements in the speech AMR. In this example, the gesture AMR reinforces and complements the command "push that block left" by encoding directionality and referential information, thereby supporting more grounded and context-aware interpretation.

To explain in more depth what was mentioned above, coreference is when two or more expressions refer to the same entity, and coreference resolution (Lee et al. (2013)) is the task of finding all these expressions. It is important for tasks such as summarization, question answering, and information extraction. Implicit roles are roles that do not have an explicit antecedent, and are often understood through context or world knowledge. For example in Figure 2.3a, `ARG0 you` refers to the actor of the gesture. Recognizing these implicit roles improves tasks such as event comprehension and question answering, as it allows for a more complete understanding of the participants and relations involved, even when not explicitly stated. Lastly, two examples of common bridging relations are part/whole and set/member relations. The former describes how a part is a component of a bigger whole, for example *the Residence* and *the Office Wings* are both parts of the White House. The latter describes items that belong to the same category; they are independent entities that share characteristics. For example, Italy and Spain are members of the set *countries in Europe.*

These phenomena are all related to multimodal coreference, which aims to identify when different mentions across modalities, e.g. speech and gesture, refer to the same entity or event. Coreference in this context involves resolving when entities mentioned in speech match gestures. For example someone might say "That one" while making a gesture which points to a specific object. Implicit semantic roles are important when certain references in speech are not explicitly mentioned but must be inferred, for example, in the sentence "On the other side the two closest to the middle get stacked" one has to infer what exactly *the two closest to the middle* refer to. Multimodal coreference resolves this implicit role by linking the unspoken reference to the objects indicated by the gesture. Lastly, bridging relations across AMRs is critical when dealing with multiple modalities. In the dataset used for this thesis, speech and gesture form separate AMRs, each conveying its own semantic representation. Bridging these representations ensures that the coreferential connections are made between them. Together, coreference, implicit roles, and bridging relations enable the integration of speech and gesture into a coherent understanding of the same referent across modalities. This is also why it is important for this thesis, because the addition of gesture might improve the model's ability to understand these referents.

2.2 Gesture

AMR was originally designed for English sentences in written text, and recent research has begun to incorporate gesture into AMR to capture the meaning of gesture. Gesture refers to a speaker's movements of the hand, arm, or other body parts, to communicate with others. It is thought that speech and (non-verbal) gestures contribute to meaning, and this thesis aims to investigate this question. The table below outlines examples of specific gesture types to illustrate how different gestures may encode distinct semantic contributions.

Gesture Type	**Examples**
Iconic	The action of smoking a cigarette.
Deictic	Pointing to a location.
Emblematic	The OK sign or a thumbs up.
Metaphoric	Tossing the hand over the shoulder while saying “That is in the past”.

Table 2.2: Overview of the different gesture types, including examples.

Two researchers, Adam Kendon and David McNeill, have published foundational gesture research, focusing on the role of gestures in speech. Kendon (2004) discusses different gesture types, their functions, and how they interact with spoken language, arguing that gestures can function as meaningful utterances. Furthermore, Kendon (1988) depicts how gestures, also described as “gesticulation”, are closely integrated with speech, giving emphasis to what is being said, or pointing out a specific object or location in the referred environment. For example, one mentions a person smoking a cigar, and simultaneously imitates a person smoking. This gesture is referred to as “iconic”, and these gesture types will be discussed in chapter 3. Moreover, gestures like pointing can help to anchor meaning in space. If a speaker says “That book there is mine” and points to a specific book, it helps to ground what “That book” refers to. Kendon also describes how gestures are used as a mode of communication, used to convey meaning. Communicative functions translate to: disambiguating ambiguous words, conveying aspects of meaning spoken words cannot capture, or only partly, completing an utterance, or substituting a non-spoken part of the utterance. Recipients in conversation pay attention to these gestures, and derive information from them. Gestures can contribute important semantic context that clarify or enrich an utterance, especially when speech alone may not fully cover the speaker’s intended meaning. Therefore gestures often provide information about the semantic contexts of an utterance. This highlights why it is of interest to explore whether gestures and their semantic representations, such as Gesture AMR, can provide additional information that complements or enhances the interpretation of speech. In the context of this thesis, this forms the basis for exploring how gesture-enriched AMR representations can inform or improve NLG.

McNeill (1985) challenges the view that gestures are merely complements of speech, arguing that speech and gesture share a computational phrase, and that they are part of the same psychological structure. One reason for this is that patients with Broca’s or Wernicke’s aphasia show similar patterns of what is preserved and what is lost after brain injury, in both speech and gesture domains. Furthermore, McNeill argues that gestures only occur during speech, stating that about 90% of all gestures only occur during speech. The 10% of gestures that occurred during silence were immediately followed by further speech. McNeill goes on to say that these gestures symbolize the function of silence in those specific situations.

In summary, while Kendon focuses more on the social and communicative aspects of gesture, McNeill focuses more on the internal cognitive processes related to gestures. This research is foundational in gesture research, and provides crucial insights in how gestures can be defined and what they might portray. The foundational concepts introduced above, such as gesture type, will be used throughout this thesis to evaluate whether gestures contribute additional information to language models. In the following section, I will introduce language models more broadly.

2.3 Language Models

Language models play an important role in Natural Language Processing (NLP), performing tasks including machine translation, text generation, or summarization. Recently, language models are also capable of processing and generating multimodal input such as audio or images, enabling them to perform tasks such as speech recognition. An example of a language model is a next-word prediction model, which is a model that predicts the next word based on the preceding tokens (Jurafsky and Martin (2025)), it does this by assigning a probability to words that might follow. This is useful for tasks that require text generation. Language models can generally be put into two categories: statistical and neural. In general, statistical models estimate the likelihood that different linguistic units, such as words or sentences, will occur (Rosenfeld (2000)). This approach falls short for tasks that are more complex, e.g. longer texts or large corpora, because a large number of probabilities needs to be calculated, which requires a lot of memory (Zhao et al. (1993)). An example of statistical language models are n-gram models. These models predict the probability of a word based on the preceding n-1 words, allowing it to assign probabilities to a whole sequence (Jurafsky and Martin (2025). Neural language models use neural networks, which consist of numerous small computational units, where each unit processes a set of input values represented as a vector and produces a single output (Jurafsky and Martin (2025)). Neural language models represent words in a continuous vector space, where words that appear in a similar context are mapped to similar representations. Neural language models include RNNSs, LSTMs, and transformer models. There are many different language models, and this thesis will focus on generative language models.

In this thesis I will prompt a generative language model with both gesture and speech input to examine the impact gesture has on the NLG ability of the model. This type of model is suited for this task due to its ability to generalize from a wide range of linguistic patterns and integrate contextual information, making it a promising tool for exploring the semantic contribution of gesture in multimodal communication.

2.3.1 Generative Language Models

Generative language models, also known as auto-regressive transformer-based models, can be used to generate new data, such as images or text. These models gain their general NLU and NLG abilities through training on extensive textual corpora, using billions of parameters (Minaee et al. (2025)). Examples of such transformer-based models are OpenAI's GPT and Llama. Llama will be explained in detail in section 4.2. There are three main architecture types used for transformers: encoder-only, decoder-only, and encoder-decoder. We will briefly discuss the decoder-only architecture because this is what will be used in this thesis. Decoder-only models do not make use of an encoder, they pass the input directly to the decoder, which essentially generates the output by predicting the next token. This architecture is mainly used for text generation, and examples of models that use this architecture are GPT and Llama.

The latter will be used in this thesis.

2.3.2 Prompt engineering

Prompt engineering is a recent field that focuses on the creation and development of prompts to effectively leverage language models across diverse applications and research areas (Saravia (2022)). In the context of this thesis, prompt engineering is particularly relevant because of the complex and multimodal input, as it combines structured semantic representations, AMR, with gesture information. Since the model must interpret and generate natural language based on this multimodal input, how the task is presented through the prompt plays a central role in shaping the output. This thesis builds on recent research into effective prompting strategies by experimenting with different prompts for the AMR-to-text generation task.

Zero-shot Prompting Zero-shot prompting entails that the prompt given to the LLM will not contain any examples or demonstrations. They do not need any examples to learn from because they have been trained on large amounts of data which allows them to generalize and infer appropriate responses without task-specific examples. An example, taken from the SAGE-AMR dataset, of a zero-shot prompt is shown below.

```
Generate a possible sentence from the following AMR:
(p / put-01
    :mode imperative
    :ARG0 (y / you)
    :ARG1 (t / them)
    :ARG2 (t2 / together))
Sentence:
```

Few-shot Prompting If zero-shot prompting is ineffective, it is helpful to include examples in the prompt, which leads to few-shot prompting. This is beneficial for more complex tasks, as it allows for in-context learning. This is what will be used in this thesis, and an example is shown below.

```
First read the following examples, then try to generate a
sentence from the given AMR:
Example:
AMR:
(s / side
        :mod (b / both))
Sentence: "Both sides"
Now generate a possible sentence from the following AMR:
(p / put-01
    :mode imperative
    :ARG0 (y / you)
    :ARG1 (t / them)
    :ARG2 (t2 / together))
Sentence:
```

This is effective for many tasks, especially tasks that require quick adaptation or real-time reasoning such as solving anagrams, doing arithmetic, or incorporating newly defined words into sentences after encountering them just once (Mann et al. (2020)). However, few-shot prompting has limitations, particularly when applied to more complex reasoning tasks, such as arithmetic reasoning. Overall, providing examples can help with certain tasks, but when zero-shot and few-shot prompting are not enough, it might indicate that the model lacks the necessary knowledge or reasoning ability for the task. If this is the case, it can be beneficial to consider fine-tuning the model or exploring more advanced fine-tuning strategies. To place this thesis within a broader context, I want to briefly discuss the paper by Webson and Pavlick (2022). The authors question whether prompt-based models understand the meaning of the prompts, and they examine this by constructing five different categories of templates to apply in a zero-shot and few-shot setting. Webson and Pavlick found that

Chain-of-Thought Prompting Chain-of-Thought (CoT) prompting, introduced by Wei et al. (2022), enhances a model's ability to handle complex reasoning tasks, such as math or logic problems, by encouraging it to generate intermediate reasoning steps before reaching a final answer. This can be combined with few-shot prompting to improve performance on tasks that require multistep reasoning.

In this thesis, CoT prompting might help by encouraging the model to break down how different inputs combine to convey meaning because interpreting both gesture and speech requires the model to reason through how the two relate. Furthermore, encouraging the model to reason explicitly could help ensure more accurate and semantically grounded sentence generation.

Wei et al. do state that this ability emerges with sufficiently large language models. They found that smaller models had semantic understanding errors of which a big portion could be fixed by upscaling the model to contain more parameters.

Prompt Chaining Lastly, prompt chaining is when a task is broken down into subtasks, to create a chain of prompts. This helps to enhance the reliability and effectiveness of LLMs. Each subtask is then treated individually by prompting the model, and the resulting output is used as input for the next prompt. Prompt chaining is an effective method for handling complex tasks an LLM might struggle with when presented in a single, detailed prompt. In this approach, a sequence of prompts is used, where each prompt builds upon the response from the previous one, enabling intermediate processing or transformation steps that guide the model toward a final output. This also helps to increase transparency of the LLM and makes it easier to troubleshoot issues in model responses.

2.4 Evaluation of NLG

An important aspect of automatic text generation is how the generated text should be evaluated. With the rapid development of Natural Language Generation (NLG), the question of how the progress of the system can be tracked has also become more relevant. It is also relevant for this thesis because evaluating the model's NLG ability is crucial to understanding the contribution of gesture information. Therefore, reliable evaluation metrics are needed to assess the quality of the outputs generated under different input conditions.

There are two main approaches concerning the evaluation of automatic text generation: automatic metrics and human evaluation. Several automatic metrics have been developed, notably in the last few years. From 2002 to 2014 there were approximately 10 NLG evaluation metrics, and since 2015 at least 36 new metrics have been developed (Sai et al. (2022)). These metrics offer a fast and convenient way to score system outputs, compare different models, as well as monitor advancements. However, these metrics might fail to capture the nuances of a diverse set of tasks. Human evaluation, on the other hand, is time-consuming and expensive and requires special annotation skills and detailed guidelines.

2.4.1 Automatic Evaluation Metrics

As mentioned above, in the past few years, several automatic evaluation metrics have been proposed. They are most commonly used because of their fast and convenient nature. This section will discuss three metrics that are frequently used to evaluate NLG systems and which will be used in this thesis to evaluate the LLM output. An overview of the metrics discussed can be found in Table 2.3.

The first metric, BLEU (Bilingual Evaluation Understudy), was proposed by Papineni et al. (2002) and was originally designed to evaluate machine translations. It compares n-grams of the machine-generated text with n-grams of the reference text, and it counts the number of matches. The more matches there are, the better the candidate translation is. As described above, an n-gram is a sequence of n words. The precision is then calculated by counting and adding all n-gram overlaps and dividing by the total number of n-grams. To account for inconsistencies where the machine translation overgenerates reasonable words, the authors count how often the target word appears in the reference sentence, and this is the maximum number of times this word can be present in the candidate translation. Moreover, a sentence brevity penalty is introduced for high-scoring candidate translations, which means it has to match the reference translation in length, word order and word choice. This is done across the entire corpus and a BLEU score is calculated. Because the score is calculated on the basis of exact n-gram matches, things such as capitalization influence the score.

The second metric that is often used is METEOR, designed by Banerjee and Lavie (2005). It was also developed for machine translation evaluation and challenges several observed weaknesses in the BLEU metric. These weaknesses include a lack of recall and the lack of explicit word matching between reference and machine translation. METEOR creates an alignment between the translation and the reference, an alignment meaning a mapping between unigrams, where each unigram in the translation maps to zero or one unigram in the reference and vice versa. Instead of considering an exact match, METEOR takes into account morphological variants, i.e. words that have the same stem, and synonyms. Then, unigram precision (P) is calculated by dividing the number of mapped unigrams in the system translation by the total number of unigrams in the system output. Similarly, unigram recall (R) is calculated by dividing the number of mapped unigrams in the system translation by the total number of unigrams in the reference translation. Then, Fmean is calculated by combining precision and recall via a harmonic-mean of P and 9R. The formula to calculate this is shown in Figure 2.4.

$$Fmean = \frac{10PR}{R + 9P}$$

Figure 2.4: Formula to calculate Fmean.

Moreover, to take into account longer matches, a penalty is calculated. Mapped unigrams that are adjacent are grouped in chunks; the more chunks there are the higher the penalty is. For example, if the machine translation is “The principle spoke to the students” and the reference translation is “The principle then spoke to the students”, there are two chunks: “The principle” and “spoke to the students”. If the number of chunks increases, the penalty also increases. This formula and the formula for the METEOR score are shown in Figure 2.5.

$$Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams\_matched} \right)^3$$

(a) Penalty formula.

$$Score = Fmean * (1 - Penalty)$$

(b) METEOR score formula.

Figure 2.5: Formula to calculate penalty and METEOR score.

Banerjee and Lavie found that METEOR Scores correlate better with normalized human judgements on Arabic and Chinese to English translations.

More recently, BERTScore was proposed as an automatic evaluation metric for text generation by Zhang et al. (2019). It computes a similarity score between tokens in the candidate sentence and tokens in the reference sentence. Instead of considering exact matches, similarity is calculated using pre-trained contextual embeddings. BERTScore addresses two common drawbacks of n-gram based evaluation metrics. First, these evaluation methods often struggle to robustly recognize paraphrases. As a result, the true performance of a system may be underestimated when valid, semantically equivalent expressions are penalized for not matching the reference wording exactly. BERTScore uses contextualized token embeddings to calculate similarity, which have been shown to be successful for paraphrase detection (Devlin et al. (2019)). Second, n-gram based models struggle to account for long-range dependencies and often penalize changes in word order that are crucial for meaning (Isozaki et al. (2010)). In contrast, contextualized embedding models are specifically trained to recognize and represent such long-distance relationships and ordering more effectively. BERTScore returns a dictionary that contains precision, recall, and F1 score for each reference and candidate. Because contextual embeddings are trained to capture long-range relationships and word sequences, these models might still find weak associations between sentences that seem unrelated. Zhang et al. found that BERTScore correlates highly with human judgement on machine translation and image captioning tasks.

In addition to the three automatic evaluation metrics mentioned above, there are other metrics such as ROUGE (Lin (2004)), chrF (Popović (2015)), as well as chrF++, (Popović (2017)).

2.4.2 AMR-to-Text Generation

One component of this thesis is the generation of text based on AMR, also known as AMR-to-text generation. Currently, AMR-to-text generation systems are most often assessed using automatic metrics that compare the generated sentence to a reference sentence, which in this context is typically the original sentence from which the AMR graph was derived. This reference sentence is human-approved (Manning et al. (2020)). However, these metrics might not accurately represent human judgments, and, addi-

Metric	**Description**	**Type of Matching**	**Strengths**	**Weaknesses**	**Use Case**
BLEU	N-gram precision based	N-grams	Fast, widely used	Ignores synonyms, recall	Machine translation
METEOR	Harmonic mean of precision and recall	Unigrams	Better alignment, recall aware	Computationally intensive	Machine translation
BERTScore	Contextual embeddings to compare tokens	Embeddings	Paraphrase aware	Sensitive to model choice	Generative tasks

Table 2.3: Overview of automatic evaluation metrics.

tionally, different sentences can be represented by the same AMR graph. This makes reference-based metrics like BLEU and METEOR less reliable for this task, as they penalize valid outputs that do not match the gold reference exactly. The addition of gesture to AMR may introduce additional nuances or disambiguations which are not present in the original reference sentence. Furthermore, most AMR-to-text research deals with written text only. No prior work has tried to evaluate gesture AMR-to-text, presenting a methodological and theoretical gap this thesis aims to explore.

Chapter 3

Dataset

To evaluate the impact of gesture information on a large language model's ability to generate speech and speech AMR, a new dataset was created. In order to do this a number of steps had to be taken, which are shown in Figure 3.1.

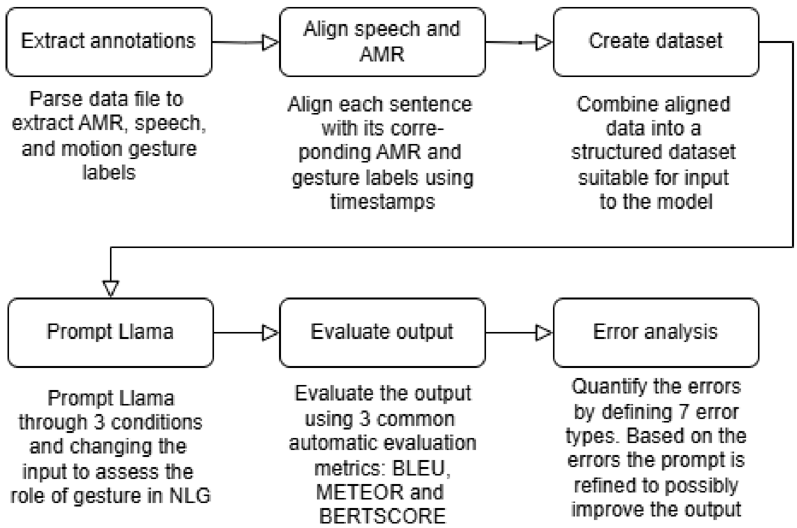


Figure 3.1: Overview of the pipeline.

This chapter will discuss the first three steps in the pipeline. The fourth step will be discussed in chapter 4, the fifth step will be discussed in chapter 6 and the final step is discussed in chapter 5 and chapter 7. In order to create the dataset, information from an existing gesture AMR corpus was extracted and aligned. Before this specific corpus is explained in subsection 3.1.1, along with some statistics, Gesture AMR and its framework are first discussed in section 3.1. Lastly, section 3.2 outlines the processing steps required to construct the dataset used for prompting the model.

3.1 Gesture AMR

Originally, AMR was designed to capture sentences in written English. However, recent studies have adapted the AMR structure to create Gesture AMR (Brutti et al. (2022), Donatelli et al. (2022), Lai et al. (2024)) to represent the semantics of gesture. The original AMR design excludes several crucial elements which are required to comprehend linguistic meaning in context. For example, the AMR shown in Figure 3.2 represents a reference to a specific location. However, the Standard AMR alone lacks spatial information to identify this location. Including gesture information can help provide this missing contextual grounding, which is what will be explored in this thesis.

(a) Deictic gesture.

```
(t/there
    :mod (r/right))
```

(b) AMR conveying “Right there”.

Figure 3.2: AMR for the sentence “Right there” along with the accompanying gesture.

Moreover, standard AMR lacks characteristics of speech such as intonation, pauses, or disfluencies, therefore lacking clear alignment of gesture to the speech represented in the AMR, as well as meaningful information contained in these signals. Extending Standard AMR to gesture does have its obstacles, such as the question whether AMR can capture the morphology of gesture, which is underspecified and may abstract away from standard principles of linguistic compositionality. (Cassell et al. (2007), McNeill (2005)). To elaborate, some gestures can be interpreted numerous ways, and could be difficult to interpret outside of the specific task. Therefore, it is difficult to claim that a gesture possesses semantics the same manner as language interpreted within its linguistic context.

According to Kendon (1994), who has published foundational gesture research, gestures play an important role in conveying meaning and managing interaction during speech. Recipients in conversation pay attention to these gestures, and derive information from them. Gestures can contribute important semantic context that clarify or enrich an utterance, especially when speech alone may not fully cover the speaker’s intended meaning. Therefore gestures often provide information about the semantic contexts of an utterance. This highlights why it is of interest to explore whether gestures and their semantic representations, such as Gesture AMR, can provide additional information that complements or enhances the interpretation of speech. In the context of this thesis, this forms the basis for exploring how gesture-enriched AMR representations can inform or improve NLG.

Gesture AMR framework

In order to develop the Gesture AMR schema, the authors annotated gestures from the EGGNOG corpus (Wang et al. (2017)), which is focused on a block building task in which one participant guides the other to build block structures. A detailed description of the EGGNOG corpus can be found in subsection 3.1.1.

The design of Gesture AMR is structured to be adaptable beyond task-oriented contexts. The authors suggested that future studies could investigate gestures that

```
(g / [gesture]-GA
  :ARG0 (s / signaler)
  :ARG1 [content]
  :ARG2 (a / actor))
```

Figure 3.3: Template used to create gesture AMR

convey expressive or extensive meanings. The annotation of Gesture AMR follows a similar pattern as Standard AMR, additionally specifying the type of gesture that is performed. This template is shown in Figure 3.3, taken from Lai et al. (2024). ARG0 represents the one who performs the gesture, the signaler, ARG1 represents the content of the gesture, and ARG2 represents the person addressed, the actor.

As discussed in chapter 2, there are four different gesture types: icon, deixis, emblem, and metaphor (Ekman and Friesen (1969), McNeill (1992)). Metaphoric gestures, which represent abstract aspects of ideas or concepts, did not appear in the corpus so these will not be discussed further. Iconic gestures are movements that represent the form of an object or illustrate how an action is performed, see Figure 3.4. The signaler makes a gesture that indicates the number one with his hand. Even though numbers are not objects nor actions, the gesture depicted still conveys the meaning, therefore it is considered an iconic gesture.

(a) Iconic gesture.

```
(i / icon-GA
  :ARG0 (s / signaler)
  :ARG1 1
  :ARG2 (a / actor))
```

(b) PENMAN notation

Figure 3.4: Example of the iconic gesture for the sentence: “Take one block.”

Deictic gestures refer to familiar pointing, referring to objects or locations in space. In Figure 3.5 the signaler is pointing twice to a location on the table. Although the direction of the gesture and what hand the signaler is using are different, both gestures are represented with the same AMR. The argument ARG1 is labelled as “location”, but its meaning remains very underspecified, as it does not indicate the precise location referenced.

(a) Deictic gesture.

(b) Deictic gesture.

```
(d / deixis-GA
  :ARG0 (s / signaler)
  :ARG1 (l / location)
  :ARG2 (a / actor))
```

(c) PENMAN notation

Figure 3.5: Example of the deictic gesture for the sentence: “Almost like that.”

Lastly, emblematic gestures are gestures with standard properties and are culturally agreed upon to have a specific meaning, for example the OK sign or a thumbs up. The meaning of a gesture may vary per country if there is a difference in culture. In Figure 3.6 the signaler makes a thumbs up gesture to the researcher next to her, not to the actor.

(a) Emblematic gesture.

```
(e / emblem-GA
  :ARG0 (s / signaler)
  :ARG1 (y / yes)
  :ARG2 (r / researcher))
```

(b) PENMAN notation

Figure 3.6: Example of the emblematic thumbs up gesture.

Gesture AMR provides a structured and symbolic way to represent the meaning of gestures, rather than just their form, like the gesture labels which are also included. This supports deeper integration of gesture with linguistic meaning in multimodal systems. However, many gestures, especially deictic ones, are difficult to semantically write down, leading to vague or non-informative Gesture AMR representations. The meaning of these gestures is highly contextual, and it is difficult to fully encode this using only AMR. This might prove to have an impact on the model's ability to interpret the given gesture information. In the next section the gesture AMR corpus will be discussed.

3.1.1 Gesture AMR Corpus Description

The gesture AMR corpus used was created by Lai et al. (2024), and is built on top of the EGGNOG, Elicited Giant Gallery of Naturally Occurring Gestures, corpus (Wang et al. (2017)). The EGGNOG corpus consists of 360 videos, 8 hours in total, where one participant, the signaler, instructs one other participant, the actor, to build a block structure as indicated. The participants were in separate rooms and could communicate through video and audio. The participants were between the ages 19 and 64. The videos are annotated with gesture intent, e.g. “think” or “rotate”, and a gesture label describing the motion of the relevant body part. Each label starts with a specific body part, and an overview can be found in Table 3.1.

Body part	**Description**
Body	Refers to broad movements involving the upper body, such as stepping forward or standing still.
Head	Captures head movements, like nodding or shaking.
Arms	Refers to either both arms, or specifically the left (LA) or right arm (RA). Small, unintentional arm movements are not included.
Hands	Refers to either both hands, or specifically the left hand (LH) or right hand (RH). These labels describe hand orientation, gestures, and motion. The palm is considered the “front” of the hand, while the opposite side is called the “back”.

Table 3.1: Overview of the gesture labels used to describe body movements.

There are other existing gesture datasets, however the EGGNOG dataset includes gesture-related elements of communication that were not covered in earlier datasets. Moreover, it is novel because it includes continuous data on people working collaboratively on a task (Wang et al.). Recognizing specific gestures requires distinguishing them from surrounding motions. Most of the earlier datasets focus only on gestures that have already been segmented in advance, such as Ren et al. (2011), who introduce a hand gesture recognition system using pre-defined hand gestures.

From the 360 videos, Lai et al. selected 21 one-minute long videos, and annotated both speech and gesture AMR, besides the already existing gesture morphology provided by EGGNOG. The participants and annotators are English-speaking students and professors, from American universities. Lai et al. added speech transcripts using the Coqui speech-to-text toolkit with the English STT v1.0.0-huge-vocab model (Coqui (2021)). The speech and gesture AMR were annotated using the program ELAN (Wittenburg et al. (2006)). An example of the interface can be seen in Figure 3.7. It includes a video of the signaler and the actor, the annotated speech and gesture AMR, the gesture intent, the gesture label describing motion, as well as the transcribed speech. In the upper right corner the full speech transcript is displayed, and additional annotations can be selected from this panel as well. Everything combined resulted in a rich multilayered dataset, which can be used to study multimodal communication.

Table 3.2 includes an overview of some statistics of the Lai et al. dataset. The first row indicates the number of signalers, broken down by gender. The corpus contains a total of 342 sentences, of which 247 include at least one gesture. A singular speech AMR was defined as one sentence. The average sentence length is approximately 8 words. The table also shows the distribution of gesture types, and in total there are 1637 annotated gestures. The final row lists the three most frequently occurring gesture labels, along with their counts.

As can be seen in Table 3.2, the majority of sentences include gestures, indicating that gestures are a core component of communication in this dataset. It is also to be noted that there are significantly more gestures than there are sentences. This implies that multiple gestures often accompany a single sentence, and that gestures are closely intertwined with speech. Iconic gestures are the most common, suggesting that many

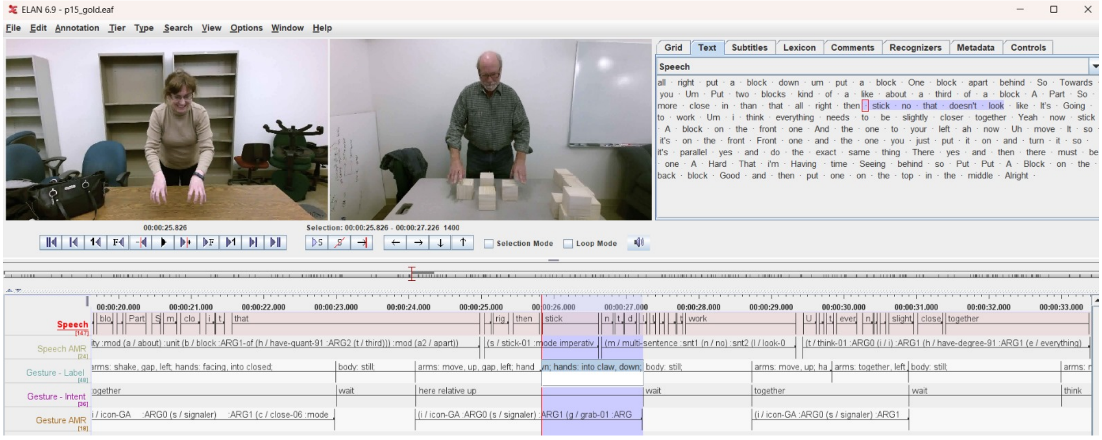


Figure 3.7: ELAN environment.

Number of signalers	male: 13 female: 7
Total number of sentences	342
Number of sentences with gesture	247
Average sentence length (words)	8.08
Iconic gestures count	245
Deictic gestures count	127
Emblematic gestures count	39
Total number of gestures	1637
Top 3 most common gesture labels	body: still (count: 199) Unknown (count: 155) RA: move, up (count: 53)

Table 3.2: Statistics of the gesture corpus

gestures in the corpus are used to visually represent shapes or movements relevant to the spoken context. Deictic gestures are also frequent, but because these gestures can be quite ambiguous this might prove to be difficult to interpret by the model.

3.1.2 Human Evaluation

In order to measure the quality of the annotated speech and gesture AMRs, the inter-annotator was calculated. This measure shows to what degree the annotators agree, and is generally calculated using Cohen's Kappa, Fleiss Kappa co-efficient, or Krippendorff's alpha (Sai et al. (2022)). It can be difficult to acquire a high value due to subjectivity within a task. Moreover, human-error, mistakes or ambiguity in the text, and unclear annotator guidelines also play a role. For the gesture AMR corpus inter-annotator agreement was calculated using Smatch (Cai and Knight (2013)) and s^2match (Opitz et al. (2020)), which are evaluation metrics for semantic feature structures, such as AMR. This was done for the annotated speech and gesture AMR. An overview can be found in Table 3.3, adapted from Lai et al. (2024). The parentheses show the 95% confidence intervals.

The Smatch scores for both speech and gesture are significantly lower than the typical range of 70-80 reported for text-based AMRs (Bonial et al. (2020)). This is due

	Smatch	S^2match
Speech	48.9	64.8
Gesture	57.5	71.5

Table 3.3: Inter-annotator agreement for the annotated speech and gesture AMR of the gesture corpus.

to a few reasons. There is notable lexical variation per annotator in how the semantic content of gestures is expressed. For example, words that indicate positive acknowledgement were annotated using different terms such as "good" or "great". These kinds of variation are less common in text-based AMRs. Even when annotators agreed on the underlying concept, differences sometimes appeared in the specific word forms used, e.g. "blocks" vs "block". In such cases where the intended meaning remains consistent, s^2MATCH supports more flexible evaluation by allowing soft matches between related concepts, as discussed by Lai et al.. This is why the s^2MATCH score better reflects the inter-annotator agreement of this task. Because the s^2MATCH score allows for soft matching, it shows there is agreement on underlying concepts, which is why it can be concluded that this gesture corpus is a valid one on top of which this thesis can be based.

3.2 Data Pre-processing

In order to prompt Llama the SAGE-AMR dataset was created; Speech Aligned with Gesture using English AMRs. To accomplish this, the annotations were exported from ELAN. While several export formats are available, the tab-delimited text format was chosen for its simplicity and ease of processing. This resulted in a .txt file consisting of timestamps and their corresponding words, speech and gesture AMR, and gesture labels. An example can be seen in Figure 3.8.

```
1.08    1.18    all
1.22    1.4     right
1.44    1.56    put
1.62    1.74    a
1.78    2.02    block
2.32    5.0     down
1.08    2.32        (p / put-01     :mode imperative    :ARG0 (y / you)     :ARG1 (b / block        :quant
1           :direction (d / down))  :mod (a / all-right))
0.959   1.992           RA: move, front left; RH: into claw, down;
1.992   4.025           body: still;
4.026   4.592           Unknown
1.44    2.32                (g / gesture-unit   :op1 (i / icon-GA       :ARG0 (s / signaler)
:ARG1 (p / put-01)          :ARG2 (a / actor))  :op2 (d / deixis-GA         :ARG0 s             :ARG1 (l /
location)           :ARG2 a))
```

Figure 3.8: Example that shows the tab-delimited format for the sentence "All right put a block down". It includes speech, speech AMR, gesture labels, and gesture AMR.

For readability I put the examples in Figure 3.8 together, however the exported .txt file first lists all separate words, then the speech AMR, the gesture labels, and lastly the gesture AMR. Because all items are listed separately, I ensured that the speech and corresponding AMRs and labels were aligned by looking at the timestamps. Aligning AMRs to speech via timestamps makes it possible to fuse gesture and speech semantics

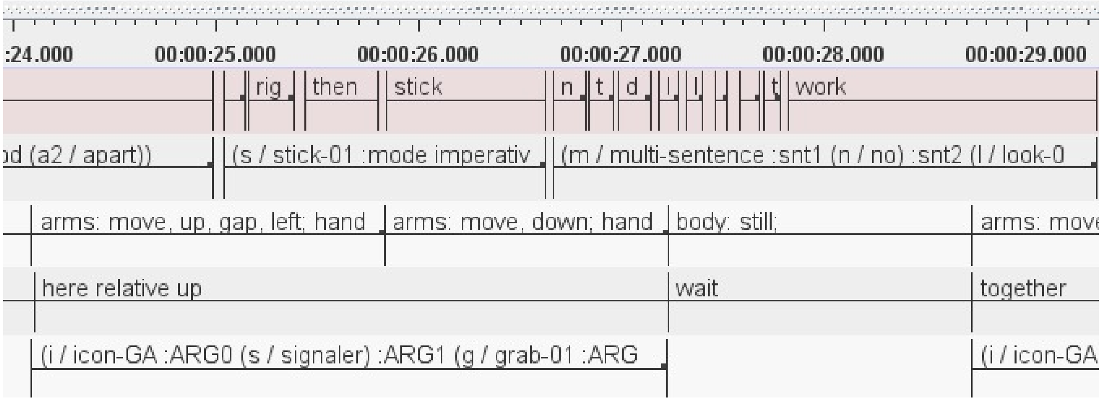
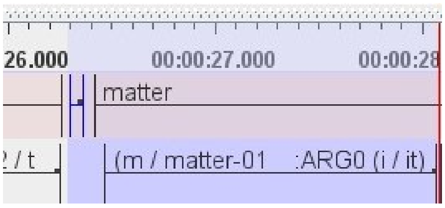


Figure 3.9: Example that shows gesture AMR overlap.

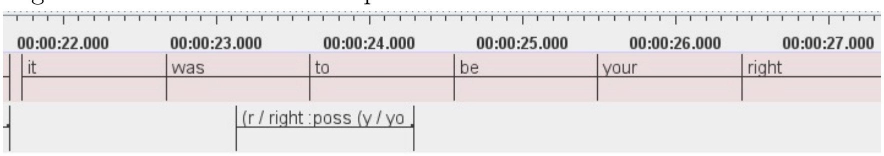
in a principled way. It allows each segment of speech to be directly associated with its corresponding AMR. Because the AMR is exported linearly, I had to write code that transforms the AMR into a readable format, following the correct indentations. This code can be found on Github[1]. Furthermore, some gesture labels or AMR overlap with multiple sentences; in these cases, the labels and AMR are included with both sentences. An example is shown in Figure 3.9; the first gesture AMR is included with both the `stick-01` speech AMR and the multi-sentence AMR.

```
5.28    7.04    um
5.28    7.03        (u / um :mode expressive)
```

Figure 3.10: Speech and speech AMR for the utterance “um”.



(a) Example where speech does not align with the AMR time stamps.



(b) Example where speech AMR does not align with the speech.

Figure 3.11: Two examples where speech and AMR do not align.

Challenges Some sentences are not accompanied with a gesture, these are removed. To align speech with its corresponding speech AMR a margin of 0.017 seconds was

[1]https://github.com/sanne-jpg/Master-thesis

applied. Without this margin, some utterances would be excluded because not all speech utterances have the exact same ending time stamp as their corresponding AMR. An example is shown in Figure 3.10. Without the margin this utterance would not have been included. However, this occasionally resulted in the inclusion of words that do not belong to the intended AMR, due to their timestamps falling within the margin. Although not optimal, this approach proved to be the most effective, as the timestamps of adjacent sentences were sometimes very close, leading to potential misalignment. This highlights a common challenge in working with dialogue data. It is inherently messy and significantly more difficult to handle than clean, pre-processed text-only data. Moreover, because the speech transcripts are not always perfectly aligned with the speech or the corresponding speech AMR, some sentences may be incomplete or slightly inaccurate. Two such examples are illustrated in Figure 3.11. In a) the signaler says "Doesn't matter" but "doesn't" starts well before the start time of the corresponding speech AMR, and "matter" falls outside the 0.017 seconds margin. As a result, this sentence was manually added to the dataset, along with one other sentence. These two sentences are specified in the code on Github. In b), a transcription error results in only the word "to" matching the speech AMR. Since this sentence lacks meaningful content, it was removed from the dataset. The code can be found on Github[2], along with the SAGE-AMR dataset that includes the speech, speech and gesture AMR and gesture labels, and the gold files.

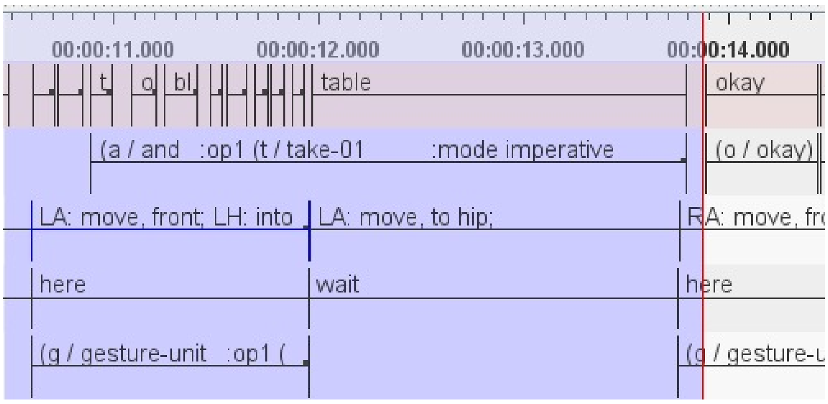


Figure 3.12: ELAN data for the sentence "Take one block and put it on the table".

It is important to note that the gold data contains some typos, and the time stamps do not always precisely align with the spoken words, leading to some occasional misalignments, see Figure 3.12. For example, due to the code structure, the third gesture label "RA:move" is associated with the first speech AMR shown in Figure 3.12, as its time stamp falls between the AMR time stamps. However, it only belongs to the second AMR, but it will be included with both. Due to time constraints, the alignment process was based solely on time stamps and a fixed margin. While this method is not perfect, it was the most practical choice. Future work could explore alignment methods based on semantic relevance to improve accuracy. Moreover, as you can see in Figure 3.12, the AMR seems to be cut off, however when the AMR box is selected, the full AMR will be displayed. This can be seen in Figure 3.13

[2]https://github.com/sanne-jpg/Master-thesis

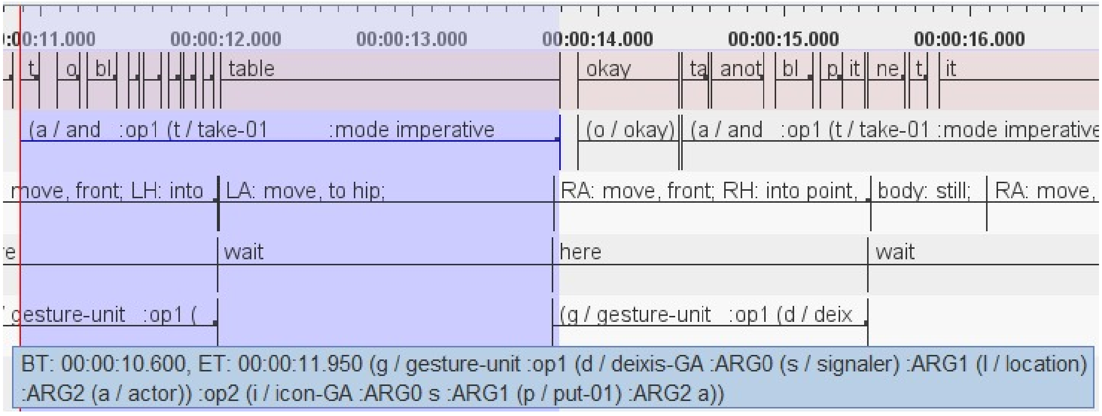


Figure 3.13: Expanded gesture AMR for the sentence 'Take one block and put it on the table".

Division of the data To evaluate whether the inclusion of gesture adds value, the data was split into a training and test set. Each file represents a different signaler, and for each file, the sentences were shuffled and 70% were assigned to the training set to provide Llama with illustrative examples to learn from. The remaining 30% formed the test set, which was used to prompt Llama to generate responses based on the specific condition. This per-file division ensures that the model is exposed to a variety of speakers and speaking styles, minimizing the risk that speaker-specific characteristics influence the results. In total, the test set contains 66 unique sentences, and the train set contains 180 sentences. To ensure that Llama is prompted with minimal pairs, each unique sentence in the test set appears three times, once for each different prompt type: speech, gesture, or speech and gesture.

Number of signalers	male: 13 female: 7
Total number of sentences	246
Average sentence length (words)	8.11
Iconic gestures count	308
Deictic gestures count	153
Emblematic gestures count	48
Total number of gestures	1415
Top 3 most common labels	body: still (count: 139) Unknown (count: 119) RA: move, up (count: 52)

Table 3.4: Statistics of the SAGE-AMR dataset

SAGE-AMR statistics Lastly, Table 3.4 shows some statistics of the SAGE-AMR dataset. Presenting these statistics ensures transparency, and illustrates the diversity and balance within the data. Compared to the gesture corpus, the SAGE-AMR dataset contains fewer gesture labels. This reduction is due to the removal of all sentences without accompanying gesture AMRs. While these excluded sentences did include gesture labels, their lack of gesture AMRs made them unsuitable for this dataset, resulting in a lower total count. These gesture labels comprise labels like "body: still", or labels that are a continuum of previous gestures, such as "arms: move, down". This removal introduces an important theoretical consideration: it narrows the dataset down to only those instances where gestures and speech co-occur in a form that can be structurally annotated. Therefore the dataset might under-represent cases where gesture is absent, ambiguous or misaligned with the speech. This raises the question whether models using this data might overestimate the role or clarity of gesture in everyday communication. The gesture types have more instances than the original gesture corpus because some gesture AMRs were duplicated and included with multiple utterances, thus increasing the count.

To conclude, the process of data extraction and alignment posed several challenges, particularly in ensuring the accurate synchronization of speech and gesture information. These challenges were addressed through iterative refinement. The resulting dataset revealed a significantly higher number of gestures compared to sentences, highlighting the central role of gestures in communication within this task. This observation reinforces the relevance of investigating how gesture contributes to NLG. The following chapter will outline the methodology employed in this thesis.

Chapter 4

Methodology

4.1 Approach of the thesis

The goal of this thesis is to explore the impact of gesture information on the NLG ability of a large language model. This gesture information is represented using Abstract Meaning Representation (AMR) and gesture labels describing the motion of the relevant body part. The study compares the model's ability to generate speech and speech AMR when using both gesture and speech input, compared to only speech AMR as input. In order to explore the impact of gesture information I create the SAGE-AMR dataset, which is based on an existing corpus. An LLM is then prompted using this new dataset. The datasets were discussed in chapter 3, and the chosen large language model, Llama, will be discussed more thoroughly in this chapter. In section 4.2 I will discuss the Llama model, outline the specific settings used for prompting, and explain the process of prompt construction in detail. Section 4.3 discusses the specific metrics used to evaluate the generated output, as well as the approach for measuring statistical significance. To construct the dataset and begin prompting Llama, a number of steps had to be taken. This chapter will discuss the fourth step, shown in Figure 4.1. The bottom row of the figure emphasizes the iterative nature of prompt engineering and evaluation. Prompts may be refined based on insights gained from the error analysis, creating a feedback loop aimed at improving generative performance.

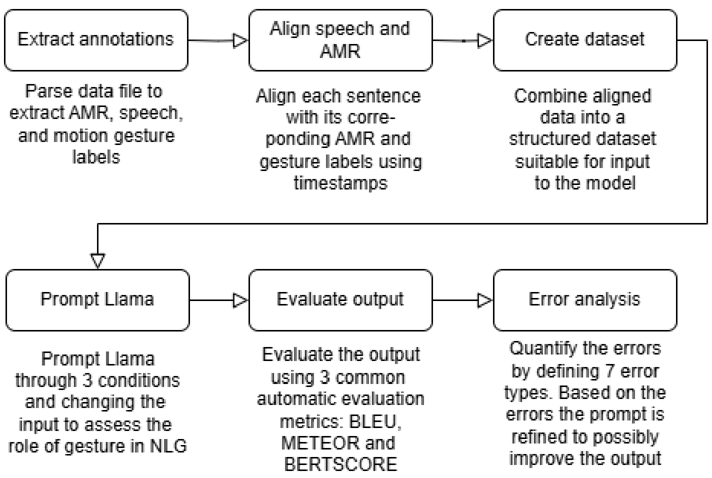


Figure 4.1: Overview of the pipeline.

Model	**Training Data**	**Params**	**Input**	**Output**	**Token Count**	**Knowledge Cutoff**
Llama 3.1 (text only)	A mix of publicly available online data.	**8B**	Multilingual Text	Multilingual Text and Code	15T+	December 2023
		70B	Multilingual Text	Multilingual Text and Code		
		405B	Multilingual Text	Multilingual Text and Code		

Table 4.1: Llama 3.1 model specifications

4.2 Llama

Llama, short for Large Language Model Meta AI, is an LLM developed by Meta (Touvron et al. (2023)). It is a state-of-the-art LLM built to facilitate research and innovation, and allows researchers that do not have access to the large amounts of computing power and other resources that are required to run LLMs to study these models. As can be seen in Table 4.1, which was adapted from Huggingface[1], Llama models range from 7B to 65B parameters, and they are competitive with other large-scale language models such as GPT-4 across a range of NLP tasks such as question answering. Furthermore, the model is trained on publicly available data unlike most models.

The specific Llama model that is used for this thesis is Llama 3.1, trained on 8B parameters[1]. It is an auto-regressive language model built on an enhanced transformer architecture for improved performance. Additionally, it is a quantized version, which means model weights are more efficiently represented by using lower-precision data types, such as 8-bit integers instead of the standard 32-bit floating-point format (Jurafsky and Martin (2025)). This makes the model inference more efficient. By reducing the bit-width of these values, the model's memory footprint is significantly lowered, energy consumption can be reduced, and computations can be executed more quickly using integer arithmetic. Furthermore, it is optimized for multilingual use cases, but in this research, only English will be used. It was trained on more than 15 trillion tokens, and trained on publicly available online data. This Llama model was chosen for this thesis because of its accessibility through Huggingface and its performance-to-size ratio. Since it is quantized, it can run more easily using only a CPU. Exploring models with more parameters was not feasible in this research due to time constraints, although future work could investigate this further.

4.2.1 Llama parameters

The goal of using an LLM for this task is to evaluate its NLG ability from structured inputs, specifically speech AMR and gesture information. A key question is whether the inclusion of gesture information enables the model to disambiguate or enrich underspecified elements in the speech AMR. To meaningfully evaluate how gesture contributes to generation quality, it is essential to understand how the model produces its output. This is governed by decoding strategies, which determine how the model selects each next word based on its probability distribution. To recap from section 2.3, a next-word prediction model is a model that tries to predict the next word based on the preceding tokens (Jurafsky and Martin (2025)), it does this by assigning a probability to words that might follow. The process of selecting a word based on a model's probability dis-

[1]https://huggingface.co/QuantFactory/Meta-Llama-3.1-8B-Instruct-GGUF

tribution is known as decoding. When this is done step-by-step from left to right, it is referred to as autoregressive generation. One of the most widely used decoding strategies in LLMs is sampling, which involves randomly selecting words in proportion to the probabilities assigned by the model. The sampling techniques described below each come with parameters that allow for balancing two key aspects of generation: quality and diversity. Techniques that focus more on the most likely words usually create text that is accurate, coherent, factually correct, and therefore have higher quality. This is often at the expense of creativity. On the other hand, methods that include less likely words often produce more varied and imaginative outputs, which are therefore more diverse, but these can sometimes be less factual, less coherent, and generally lower in quality (Jurafsky and Martin (2025)). Additionally, the parameters play a role in balancing accuracy and generative flexibility, making their careful selection essential to the quality of the output. In this thesis, I aim for outputs with high accuracy, therefore allowing limited flexibility and creativity. I do not want the model to generate overly diverse responses because it should not divert too much from the given AMR and gesture information.

In order to test what parameter settings yielded the most accurate and coherent outputs, Llama was prompted with one or two example sentences from the dataset, while systematically varying temperature, number of examples given, max number of tokens, and frequency penalty. For this thesis, Llama was run using the OpenAI API, and the full list of possible parameters and their default values can be found on the OpenAI website[2].

Frequency penalty I chose to leave out this parameter. It controls the repetitiveness of the model's responses, where a higher value encourages more diverse responses. Since this task involves NLG as well as generating AMRs, encouraging excessive diversity was not considered beneficial, as some AMRs may be very similar, therefore yielding similar sentences.

Temperature To prevent the model from generating overly diverse or unpredictable responses, it was decided to keep the temperature low, specifically set to 0.3. In this task, where outputs are expected to follow the structure and meaning of specific AMR graphs or instructional patterns, excessive diversity can lead to hallucinations, paraphrasing that strays too far from the intended meaning, or inconsistent use of terminology. Therefore, a lower temperature helps maintain consistency to the source meaning and reduces the chance of introducing semantic errors.

Top-p This parameter, also known as nucleus sampling, limits the next word selection to the top tokens whose combined probability adds up to at least a threshold value P, with a default value of 1. A higher value allows for more diverse output. It was observed that a value of 0.9 led Llama to provide more detailed explanations of its reasoning compared to a value of 0.1, although the generated sentences remained largely unchanged. Since diversity in output is not the main goal of this thesis, it was decided not to change this parameter, which means the default value of 1 was used. A value of 1 indicates that the model samples from the full probability distribution. Furthermore, the OpenAI website[2] recommends altering either the top-p or temperature parameter,

[2]https://platform.openai.com/docs/api-reference/chat/create

Prompt type	Inputs	Outputs	Prompt
Speech	Speech AMR + examples	Sentence + explanation	Given the following speech Abstract Meaning Representation (AMR), generate a corresponding sentence in natural spoken English. Provide a short explanation. First, read the examples to understand the AMR format:
Gesture	Gesture labels, Gesture AMR + examples	Sentence, Speech AMR + explanation	Given the following gesture label(s) and gesture Abstract Meaning Representation (AMR), generate a corresponding sentence and speech AMR in natural spoken English. Provide a short explanation. First, read the examples to understand the AMR format:
Speech_gesture	Speech AMR, Gesture AMR, Gesture labels + examples	Sentence + Explanation	Given the following gesture label(s), gesture Abstract Meaning Representation (AMR), and speech AMR, generate a corresponding sentence and speech AMR in natural spoken English. Provide a short explanation. First, read the examples to understand the AMR format:

Table 4.2: Prompt types and their corresponding inputs/outputs.

but not both. Since the temperature was already chosen, top-p was deliberately left at its default.

Number of examples After some experiments, two examples were chosen as the optimal number to provide to Llama. A single example was insufficient, while three risked overwhelming the prompt.

4.2.2 Prompting Llama

To evaluate the impact of gesture information on an LLM's ability to generate speech and speech AMR, I prompted Llama using three different input conditions: speech AMR alone, gesture information alone, and gesture information and speech AMR combined. The prompt looks different per condition and an overview can be found in Table 4.2. A complete example prompt for the speech condition can be found in Table 4.3, the full prompt for the other conditions can be found in the Appendix.

The first condition, **speech-only**, is designed to evaluate whether Llama can generate an appropriate sentence based solely on the speech AMR. This condition serves as a baseline to assess the model's NLG ability with only the AMR derived from spoken input. Importantly, this condition provides a minimal and controlled input condition that mimics the setup of AMR-to-text generation tasks. By supplying only the AMR graph I establish a clear, input-constrained scenario against which other, richer input conditions can be compared. This makes it an effective baseline, because any improvements observed in the other conditions can be attributed to the additional input. In this condition, the model is first given two example pairs to familiarize itself with the AMR notation and how it maps to natural language. It is then presented with a new speech-derived AMR from which it must generate an appropriate sentence, along with a brief explanation of its reasoning. This explanation encourages the model to explicitly reflect on the generation process, which may in turn lead to more accurate or coherent outputs. It also provides insight into its reasoning, which is useful for the error analysis.

The second condition, **gesture-only**, is designed to evaluate whether Llama can generate both an appropriate sentence and speech AMR using only gesture-derived inputs. Specifically, gesture labels describing the motion of the relevant body part and their associated gesture AMR. This condition isolates gesture-based semantics to determine whether they can serve as an appropriate substitute for spoken language in guiding NLG. Asking the model to generate both a sentence and a corresponding speech AMR helps assess whether Llama can reconstruct high-level linguistic meaning from gestures alone, in both textual and structured semantic forms. In this condition, the model is first given two example pairs to familiarize itself with the gesture labels and gesture AMR notation and how it maps to natural language. It is then presented with a new gesture-derived AMR as well as the corresponding gesture labels, from which it must generate an appropriate sentence and its corresponding AMR, along with a brief explanation of its reasoning. Strong performance in this condition would suggest that gestures alone can carry enough compositional and contextual meaning to derive both surface realization and deeper semantic abstraction. This performance is measured through three automatic evaluation metrics: BLEU, METEOR, and BERTScore.

The third condition is **multimodal**, where speech and gesture inputs are combined. Specifically, the model reveives gesture labels, as well as AMRs derived from both speech and gesture. This condition is designed to test whether the integration of multiple modalities enhances Llama's NLG ability. Multimodal learning reflects how humans often process language. Speech is combined with non-verbal cues such as gestures to enrich communication. This condition mirrors that process, aiming to determine whether Llama can leverage semantic information from both modalities. The central question is whether gesture information adds value beyond what is already present in the speech AMR. If this condition outperforms the speech-only baseline, it would suggest that Llama is capable of meaningful multimodal fusion. In contrast, if no improvement is observed, it may indicate that gesture information does not significantly enhance Llama's understanding or generation capabilities in this context, or that current gesture representations do not effectively convey additional useful information. The three conditions are also summarized in Table 4.2.

In order to test whether the different input has an impact on the performance of the model, I make sure I keep minimal pairs when prompting Llama with test examples. This means that the same test sentence appears once for every condition with different input. Additionally, each prompt is run three times with different seeds. By collecting multiple generations per input, I can better assess the consistency and variability of the model's responses within a condition. This also supports a more nuanced evaluation. Moreover, to draw meaningful and reliable conclusions from the experiment, it is essential to ensure that the evaluation setup is robust and methodologically sound. One key concern when working with LLMs is their sensitivity to prompt formulation.

Even minor differences can influence the model's behaviour (Errica et al. (2024)). To mitigate this, I ensure all prompts are designed with a consistent and parallel structure across the three experimental conditions. This consistency ensures that the only variation between prompts is the type of input data provided, rather than any confounding factors introduced by prompt wording or layout. Maintaining this structural alignment is crucial for internal validity: it allows for the isolation of the effects of modality on the model's output, rather than testing the effects of prompt design. Together, these methodological choices, prompt parallelism, minimal pairs, and repeated sampling, are critical for establishing a robust experimental foundation. They help ensure that any observed differences in Llama's performance across conditions are attributable to the intended experimental manipulation, rather than to uncontrolled factors in the prompting setup or randomness in generation.

The size of the dataset necessarily restricts the number of examples that can be given to Llama during prompting. Llama is prompted with two examples per prompt, but since each sentence is presented under three conditions, the example pool is not large enough to support fully balanced prompting across all conditions. This means that certain examples are reused throughout the prompts. The training set was composed randomly, and for each prompt two consecutive examples were chosen, iteratively. Future research could explore choosing examples based on informativeness. During the experiments I observed that not all sentences are as informative, and some might have been too long and complex for the model to parse. These variations in input complexity may have led to different types of generation errors, an to better understand where and why the model fails, an error analysis was conducted. This analysis aims to discover systematic patterns in the model's output and will be discussed in chapter 6.

4.3 Evaluation

In order to evaluate the sentences generated by Llama, I make use of the three automatic evaluation metrics explained in section 2.4; BLEU, METEOR, and BERTScore.

BLEU BLEU is included as a widely used standard for assessing machine-generated text. However, it is not expected to perform well in this context, since it is based on exact n-gram matches. Because multiple sentences can correspond to the same AMR graph I believe BLEU will underestimate the true quality of the generated text. Especially because Llama's outputs are unlikely to closely match the gold references. I will be using the SacreBLEU implementation (Post (2018)), which most closely replicates the original implementation.

METEOR METEOR builds on BLEU by incorporating stemming, synonym matching, and a harmonic mean of precision and recall, enabling a more linguistically informed evaluation. Given its ability to recognize morphological variations, I believe it will provide a more nuanced comparison. Therefore I think METEOR will yield higher scores than BLEU, particularly since the generated sentences may differ from the references while still being semantically appropriate. In order to calculate the METEOR score I will be using the NLTK[3] implementation.

[3]https://www.nltk.org/api/nltk.translate.meteor_score.html

BERTScore BERTScore complements these by leveraging contextual embeddings to assess similarity between generated and reference texts, offering a fundamentally different perspective on similarity. Unlike BLEU and METEOR, which rely on surface-level token matches, BERTScore captures deeper contextual meaning, making it well-suited for the evaluation of NLG tasks where exact wording may differ, but underlying semantics are preserved. Given the flexible nature of AMR-to-text generation and the probable variability in correct outputs, I expect BERTScore to provide the highest scores. To calculate the BERTScore I will be using the implementation as explained by Zhang et al. (2019).

Paired Bootstrap Sampling To test whether the scores obtained from these metrics are statistically significant and not due to chance, I make use of the paired bootstrap test. Bootstrapping is a method that involves repeatedly taking many random samples *with replacement* from an original dataset (Jurafsky and Martin (2025)). *With replacement* means that each instance is returned to the pool after being picked before drawing the next sample, therefore they can be picked again. The idea behind the bootstrap test is that by resampling from the observed test set, one can create many alternative versions of the test set. This allows us to estimate whether the obtained results are due to chance.

Prompt type	Prompt
Speech	Given the following speech Abstract Meaning Representation (AMR), generate a corresponding sentence in natural spoken English.Provide a short explanation. First, read the examples to understand the AMR format: Example 1: Sentence: okay Speech AMR: (o/okay-04) Example 2: Sentence: space two out a little less than a block length Speech AMR: (s/space-01 :mode imperative :ARG0 (y/you) :ARG1 (i/implicit-role :quant 2) :ARG2 (q/distance-quantity :unit (b/block) :ARG1-of (h/have-quant-91 :ARG2 1 :ARG3 (l/less :mod (l2/little))))) Now, generate a sentence from the following speech AMR and explain your reasoning. Please provide the output only in json format: `[{"sentence": "Your generated sentence here.", "explanation": "Your explanation here."}]` Speech AMR: (p/put-01 :mode imperative :ARG0 (y/you) :ARG1 (b/block :quant 1) :ARG2 (b2/block :mod (b3/back)))

Table 4.3: Full prompt for the speech condition, test sentence “so put put a block on the back block good”.

Chapter 5

Results

This thesis investigates whether gesture influences the NLG ability of an LLM. This chapter contains the results of the experiments described in chapter 4. First the overall scores for all conditions are presented, followed by an analysis of results from each metric separately accompanied with two or three examples to showcase and explain specific obtained results. The three metrics, BLEU, METEOR, and BERTScore, capture different aspects of the generated output. BLEU and METEOR focus more on surface-level similarity, while BERTScore measures semantic similarity using contextual embeddings. Examining all three provides a more nuanced view of how gesture input affects NLG quality.

5.1 Speech vs. Gesture NLG Evaluation

To score the performance of the model, three automatic evaluation metrics were used. The three methods that were described in detail in section 2.4 are BLEU, METEOR, and BERTScore, which will be discussed in the following paragraphs. An overview of all the scores can be found in Table 5.1.

Scenario	**BLEU**	**METEOR (average)**	**BERTScore**
All Scenarios	0.1096	0.2855	Precision: 0.5371 Recall: 0.5455 F1: 0.5382
Speech	0.1918	0.4671	Precision: 0.6609 Recall: 0.6653 F1: 0.6604
Gesture	0.00697	0.0715	Precision: 0.4035 Recall: 0.4134 F1: 0.4053
Speech+Gesture	0.1117	0.3180	Precision: 0.5469 Recall:0.5578 F1: 0.5488

Table 5.1: Evaluation scores for the metrics BLEU, METEOR, and BERTScore across different prompting conditions. The scores range from 0 to 1, where a higher score means there is more overlap and is therefore more accurate.

5.1.1 BLEU

In order to calculate the BLEU score, the reference and candidate sentences were extracted, which were then used to calculate the BLEU score over the whole corpus. BLEU scores are the lowest overall, as shown in Table 5.1. The BLEU score for the speech scenario is highest, with a score of 0.1918, which is 0.08 higher than the speech and gesture scenario. Compared to state-of-the-art (SOTA) AMR-to-text-generation results, the BLEU scores of the Llama model are significantly lower across all conditions, particularly for the gesture-only condition, see Table 5.2. For context, leading systems such as BiBL and AMRBART achieve BLEU scores between 47 and 51.5, while the Llama's best performance remains below 20. This difference is expected, as Llama was not fine-tuned on any AMR-to-text datasets, whereas the SOTA models were specifically trained for this task.

	BLEU
BiBL	47.0
BiBL+Silver	51.5
AMRBART large	49.8
Llama (mine)	speech: 19.2 gesture: 0.7 speech+gesture: 11.2

Table 5.2: SOTA BLEU scores for AMR-to-text-generation compared with the Llama model, reported as percentages.

Bidirectional Bayesian learning (BiBL), proposed by Cheng et al. (2022), is a multitask sequence-to-sequence model that jointly performs AMR-to-text generation and Text-to-AMR parsing. It is trained on the corpora AMR 2.0 and AMR 3.0 which consist of 36.521 and 55.635 sentence-graph pairs, respectively. The BiBL+Silver variant extends this by incorporating additional silver data. Specifically, the authors used the best performing Text-to-AMR model to automatically annotate sentences from the unlabeled English Gigaword corpus. This yielded an extra 200.000 sentence-graph pairs which were used for training. This led to a performance gain over BiBL without silver data.

The paper by Bai et al. (2022) addresses a gap in pretrained language models (PLMs). While pretrained sequence-to-sequence models excel at AMR parsing and generation when fine-tuned, they are not structurally aware of graph data. The authors investigate whether pretraining directly on AMR graphs improves performance on both parsing and graph-to-text tasks. Bai et al. also make use of the AMR 2.0 and AMR 3.0 corpora.

Example Sentences To better understand the BLEU score, I present two in-depth examples. Since the corpus-level BLEU score aggregates performance over the entire dataset, sentence-level BLEU scores are used to analyze performance at the level of individual examples, where corpus statistics are not applicable.

```
reference: put two blocks
scenario: speech
candidate: put two blocks.
score: 0.431
```

The sentence only has three tokens, therefore the computation of the n-gram weights is changed, which leads to a lower score, even though intuitively the candidate is almost identical to the reference. This is because BLEU is precision based, not semantic based, and because of the punctuation there are unmatched n-grams, which causes the score to drop.

```
reference: one block on top of each of those two
scenario: speech+gesture
candidate: put the block on top of each other.
score: 0.446
```

This score is close to the score above, even though this candidate is not close to the reference sentence. This is because there is enough n-gram overlap, and even 4-gram overlap, unlike the previous sentence, namely "block on top of each". There is a semantic mismatch, "each of those two" is not the same as "each other", however, BLEU does not register this, and rewards surface overlap. Because of this focus on surface-level evaluation, I do not think BLEU is an accurate metric for evaluating this NLG task, especially given its AMR focus and the potential for multiple valid sentences.

5.1.2 METEOR

In order to calculate the METEOR score, the average was calculated for each scenario separately, as well as across all scenarios combined. Unlike BLEU, METEOR is not directly additive across a corpus, since it calculates scores on a sentence-level. That is why the average was manually calculated. As can be seen in Table 5.1, the speech scenario scores the highest, with an average score of 0.4671. This is 0.15 higher than the speech and gesture scenario. Gesture has again the lowest score, and reasoning for this will be discussed in chapter 7. When compared to SOTA AMR-to-text generation models such as BiBL, BiBL+Silver, and AMRBART, the speech-only scenario actually slightly exceeds their best METEOR scores. This is particularly interesting because while the BLEU scores were much lower, the METEOR results suggest that the model's outputs for speech inputs are better at preserving semantic content, paraphrasing, and lexical flexibility. In other words, although the exact word overlap, which is captured by BLEU, is lmited, the generated sentences often capture the same meaning with different wording or synonyms, which METEOR is better at detecting.

	METEOR
BiBl	43.2
BiBL+Silver	45.2
AMRBART large	42.6
Llama (mine)	speech: 46.7 gesture: 7.2 speech+gesture: 31.8

Table 5.3: SOTA for METEOR AMR-to-text-generation compared with the Llama model, reported as percentages.

Example Sentences To better understand the METEOR scores, I present three in-depth examples.

```
reference: so put put a block on the back block good
scenario: speech
candidate: put the back block.
score: 0.395
```

This sentence has four unigram matches with the reference sentence: put, the, back, and block. There are two chunks: “put” and “the back block”. Therefore the penalty is calculated as follows:

$$0.5\left(\frac{2}{4}\right)^3 = 0.0625 \tag{5.1}$$

Then, precision, recall, and Fmean are calculated.

$$\begin{aligned} precision &= \frac{4}{5} = 0.8 \\ recall &= \frac{4}{10} = 0.4 \\ Fmean &= \frac{10 \times 0.8 \times 0.4}{0.4 + 9 \times 0.8} = 0.421 \end{aligned} \tag{5.2}$$

Lastly, the METEOR score is calculated.

$$score = 0.421 \times (1 - 0.0625) = 0.395 \tag{5.3}$$

This is a moderate METEOR score, some lexical alignment is captured however too much of the reference sentence is omitted in the candidate sentence. The candidate sentence contains four correct unigrams and two contiguous chunks, which results in a relatively low penalty score. However, the recall is low, which means that more than half of the reference sentence is omitted. METEOR’s use of recall and the penalty reflects this imbalance, resulting in a final score of 0.395. Subsequently, we will discuss a high METEOR score.

```
reference: put two blocks
scenario: speech
candidate: put two blocks.
score: 0.950
```

The same calculations as above will be repeated to show how the final METEOR score is calculated.

$$\begin{aligned} penalty &= 0.5\left(\frac{1}{3}\right)^3 = 0.0185 \\ precision &= \frac{3}{4} = 0.75 \\ recall &= \frac{3}{3} = 1 \\ Fmean &= \frac{10 \times 0.75 \times 1}{1 + 9 \times 0.75} = 0.968 \\ score &= 0.968 \times (1 - 0.0185) = 0.950 \end{aligned} \tag{5.4}$$

All three words in the reference sentence are present in the candidate sentence, therefore the recall is 1. Precision is also close to 1, resulting in a very high Fmean. Due to punctuation and the penalty, the score is not 1. Lastly, I want to discuss a counterintuitive example, which is shown below.

```
reference: okay
scenario: speech
candidate: okay
score: 0.500
```

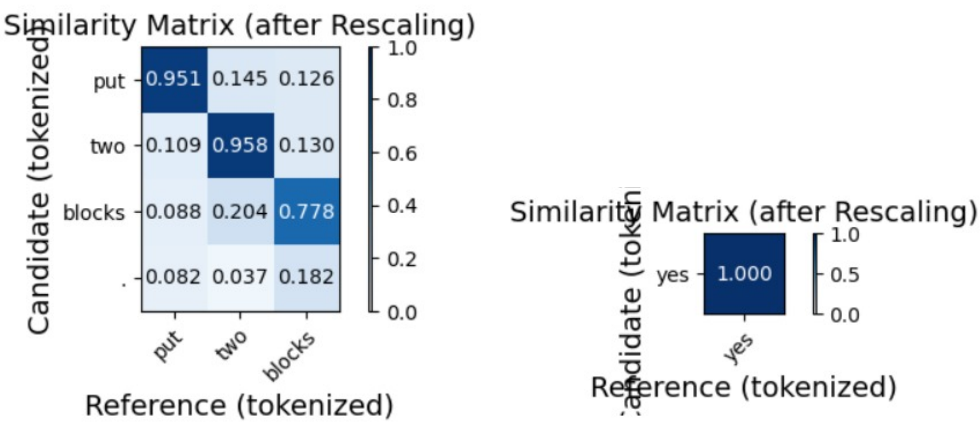
This sentence only contains one word, which leads to a penalty of 0.5 and consequently to a total score of 0.5.

Overall, I think METEOR is better than BLEU for evaluating this specific NLG task because it is more sensitive to sentence structure, as well as missing content because of its incorporation of recall. In addition, its consideration of morphological variants, such as synonyms, makes it a more suitable metric for this task.

5.1.3 BERTScore

Lastly, BERTScore returns the precision, recall, and F1 score for each reference and candidate. Precision in this context refers to how relevant the generated words are compared to the reference, and recall refers to how much of the reference meaning is captured by the generated sentence. Again, the average was calculated for each scenario separately, as well as across all scenarios combined. The contextual embedding model "bert-base-uncased" is used to calculate the scores. The BERTScores are higher than the other reported scores, with the speech scenario achieving the highest precision, recall, and F1 scores, 0.6609, 0.6653, and 0.6604 respectively.

Example Sentences BERTScore makes use of contextual embeddings to calculate precision, recall, and F1 score, which is why it is not possible to explain how exactly these scores were determined. However, I will show two similarity matrices which include the assigned score per token, this can be found in Figure 5.1.



(a) Reference: "put two blocks". (b) Reference: "yes".

Figure 5.1: Plot showcasing the BERTScore similarity matrix for two sentences.

As can be seen, the diagonal values are close to 1, indicating that the candidate sentence contains the same words as the reference sentence. All off-diagonal scores are low, indicating that BERTScore correctly distinguishes between unrelated tokens. Its output is harder to analyze than BLEU and METEOR scores because it is based on contextual embeddings and does not give a straightforward score that can be checked, however I think it is a better suitable metric for this task. Compared to BLEU and METEOR, BERTScore takes context into account, it captures semantic similarity, and it is robust to paraphrasing.

5.2 Paired Bootstrap Test

In order to test whether the differences between the scenarios are statistically significant, I performed paired bootstrap resampling, as explained in section 4.3. I did this between all scenarios, to check whether the results obtained are statistically significant, and not due to chance. The results can be found in Table 5.4.

Comparison	BLEU	METEOR	BERTScore
Speech vs. Gesture	Δ = 0.1856 p = 0.0000	Δ = 0.3954 p = 0.0000	ΔP = 0.2573, p = 0.0000 ΔR = 0.2517, p = 0.0000 ΔF1 = 0.2549, p = 0.0000
Speech vs. Speech+Gesture	Δ = 0.0765 p = 0.0000	Δ = 0.1492 p = 0.0000	ΔP = 0.1142, p = 0.0000 ΔR = 0.1076, p = 0.0000 ΔF1 = 0.1117, p = 0.0000
Gesture vs. Speech+Gesture	Δ = -0.1091 p = 1.0000	Δ = -0.2461 p = 1.0000	ΔP = -0.1431, p = 1.0000 ΔR = -0.1440, p = 1.0000 ΔF1 = -0.1433, p = 1.0000

Table 5.4: Paired bootstrap sampling performed on all scenario pairs.

BLEU For the first comparison, Δ indicates the average difference between the BLEU score for speech, and the BLEU score for gesture. This difference of 0.1856 means that on average, the speech corpus score is 0.1856 points higher than the gesture corpus score. The value p = 0.0000 indicates that the probability that gesture outperforms or equals speech just by chance is essentially zero. Therefore, we can say the difference between these two scenarios is statistically significant.

For the second comparison, Δ = 0.0765 indicates that on average, the speech corpus score is 0.0765 points higher than the speech and gesture corpus. Again, p = 0.000 therefore this difference is statistically significant.

Lastly, between gesture and speech and gesture there is a difference of -0.1091. This means that on average, the gesture corpus score is 0.1091 points lower than the speech and gesture corpus score. Since p = 1.0000, it means that in all samples, gesture performed worse than or equal to speech and gesture. Again, this difference is statistically significant, and we can say that gesture underperforms compared to speech and gesture combined.

METEOR For the first comparison, on average, the METEOR score for speech is 0.3954 points higher than the METEOR score for gesture. This is quite a big difference, and because p = 0.0000 we can say that this difference is very unlikely to be due to chance.

For the second comparison, on average, the METEOR score for speech is 0.1492 points higher than the METEOR score for speech and gesture. Because p = 0.0000 we can say that this difference is unlikely to be due to chance.

Lastly, between gesture and speech and gesture there is a difference of -0.2461. This means that on average, the gesture score is 0.2461 points lower than the speech and gesture score. Since p = 1.0000, it means that in all samples, gesture performed worse than or equal to speech and gesture. Again, this difference is statistically significant, and we can say that gesture underperforms compared to speech and gesture combined.

BERTScore For the first comparison, across all BERTScore metrics, speech shows a statistically significant improvement over gesture, by approximately 0.25 points. For the second comparison, speech scores again statistically significantly higher than gesture, by approximately 0.11 points. Lastly, gesture underperforms compared to speech and gesture combined across all BERTScore metrics, by approximately 0.14 points. Because p = 1.0000, we can say that gesture performed worse or equal to speech and gesture in every sample.

5.3 Text-to-AMR Evaluation

For the gesture-only condition, Llama was prompted to first generate a sentence based on gesture information. This generated sentence was then use as input to produce a corresponding speech AMR. The resulting AMRs were compared to the gold speech AMRs using the Smatch metric Cai and Knight (2013)). See subsection 3.1.2 for the explanation of this metric. The average precision, recall, and F1 score are shown in Table 5.5.

Precision	Recall	F1
0.275	0.242	0.257

Table 5.5: The average Smatch scores for the speech AMRs generated by Llama over the SAGE-AMR dataset.

These scores are expectedly low. Since Llama generated sentences solely based on gesture information, it often produced output that diverged significantly from the gold references. Consequently, the AMRs derived from those sentences frequently failed to match the gold speech AMRs. Moreover, during evaluation, many of the generated AMRs were invalid, making Smatch evaluation impossible for 62 out of 198 sentences.

Chapter 6

Error Analysis

This chapter presents the error analysis conducted following the initial experimental runs. This is the second to last step in the pipeline, as shown in Figure 6.1. Based on the observed mistakes, the original prompt was refined and a subset of test sentences was selected to evaluate the revised prompt. In section 6.1 I categorize and quantify the errors by defining several error types and systematically identifying their occurrences in the data. Representative examples of these errors are also provided. In section 6.2 I describe the revised prompts and assess their effectiveness by examining improvements in the model's output.

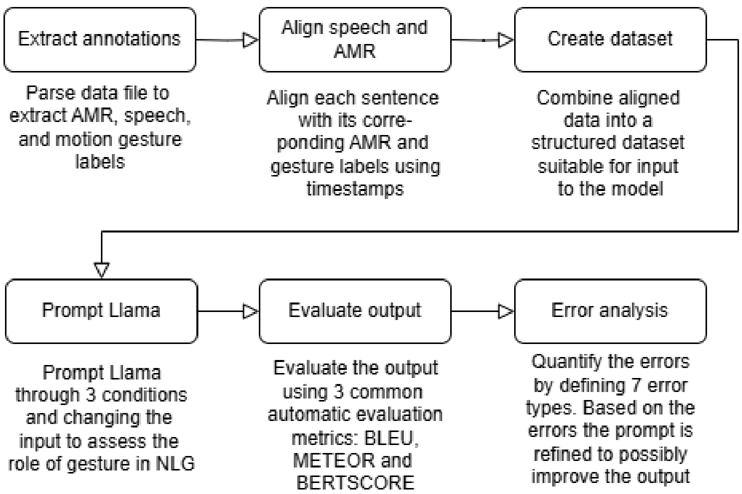


Figure 6.1: Overview of the pipeline.

6.1 Overview Error Analysis

In order to quantify the errors, seven error types were identified and will be defined below: Propbank, Polarity, Deictic Gestures, Gesture Information, AMR Misinterpretation, Incoherent and Others. During the analysis, it was noted that the model occasionally failed to follow the expected response format. As a result, due to time constraints, some sentences could not be parsed and were excluded from the final results and error analysis. An example of a response that did not follow the expected format is

```
[
  {
    "sentence": "It matters.",
    "explanation": "In the speech AMR, ':ARG0 (i/it)' indicates that 'it' is the subject of the sentence. The verb 'matter-0
1' implies a sense of importance or significance, which corresponds to the positive polarity indicated by '-)).'"
  }
]
[
  {
    "sentence": "It matters.",
    "explanation":
      The speech AMR indicates that the word "matter" is in its base form, and it has a polarity of -), which typically corre
sponds to a negative or downward tone. This suggests that the sentence should convey a sense of importance or relevance being
diminished. In this case, the generated sentence "It matters." conveys a neutral statement about something being important, b
ut the original speech AMR implies a more nuanced meaning.
  }
]
```

Figure 6.2: Example of correct and incorrect model outputs. The first example is a correct sentence and explanation pair, the second one is incorrect.

Error Type	**Mistakes**	**Percentage**
Propbank	25	3.77%
Polarity	23	3.46%
Deictic Gestures	46	6.93%
Gesture Information	231	34.79%
AMR Misinterpretation	176	26.51%
Incoherent	44	6.63%
Other	119	7.92%

Table 6.1: Error types with their error count, also shown in percentages.

shown in Figure 6.2, along with a correct response. The second explanation begins on a new line, which breaks the regex pattern and prevents it from matching. As a result, the generated sentence is also omitted. Because of time constraints this issue was not solved. In total, 573 sentences were evaluated for the error analysis. As previously noted, the test set contains 66 unique sentences. Each unique sentence is evaluated under three conditions, resulting in 198 prompts. Each prompt was run three times using different seeds to capture variation in Llama's output, yielding 594 generated sentences overall. However, due to parsing issues, 21 outputs were excluded, leaving 573 sentences for analysis. In the following subsections the identified error types will be further explained, and examples are shown. An overview of the error types and their frequencies is shown in Table 6.1.

6.1.1 Propbank

Propbank errors refer to issues related to the interpretation of Propbank structures and meaning. This entails errors interpreting semantic roles or meanings of verbs, including incorrect interpretation of verb senses or argument assignments. Of the 573 sentences evaluated, there were 664 mistakes. Mistakes are identified by examining the entire sentence and categorizing each error according to the error type it most closely resembles. A single sentence can contain multiple mistakes, each corresponding to different error types. Of the 664 mistakes, 25 were labelled Propbank, which is 3.77%. An example from the speech condition is shown in Figure 6.4. The yellow highlights the part that is relevant for the error, the red shows that there can be multiple errors per sentence.

```
Speech AMR:
(m/multi-sentence
    :snt1 (n/no)
    :snt2 (l/look-02
        :ARG0 (t/that)
        :ARG1 (w/work-09)
        :polarity -))
Reference: "no that doesn't look like it's going to work"
Candidate: "no, look that work"
Explanation: "The speech AMR indicates a multi-sentence structure
with two sentences. The first sentence is 'n/no', which
corresponds to the word 'no' in English. The second sentence has
a subject 't/that' and an action 'l/look-02', which means 'to
look'. The object of the verb 'work-09' is not explicitly
mentioned, so it can be inferred that the speaker is referring to
something related to work. Therefore, the generated sentence is
'No, look that work.'"
```

Figure 6.3: Example from the speech condition showcasing the Propbank error type.

Llama interprets `look-02` as meaning "to look", whereas according to Propbank, look-02 corresponds to the sense "to appear" or "to seem". The same goes for work-09, which refers to "to function" or "to operate", whereas Llama interprets it as a noun. This misunderstanding of the correct frameset led to the model generating incorrect sentences, highlighting the model's difficulty in distinguishing different verb and noun senses.

6.1.2 Polarity

Polarity errors involve the misinterpretation or omission of negation, as indicated by the presence of a polarity relation in the speech AMR. Of the 664 mistakes, 23 were categorized as Polarity, which is 3.46%. While one instance of a polarity error is already illustrated under the Propbank error category, an additional example is shown in Figure 6.4, taken from the speech condition:

```
Speech AMR:
(m/matter-01
    :ARG0 (i/it)
    :polarity -))
Reference: "it doesn't matter"
Candidate: "it matters."
Explanation: "The speech AMR indicates that 'matter' is the main
concept, with 'it' as the subject (:ARG0) and a positive polarity
(-)). This suggests that the sentence should convey a neutral or
affirmative statement about something being important or
significant."
```

Figure 6.4: Example from the speech condition showcasing the Polarity error type.

In this example, the speech AMR explicitly includes a negative polarity, indicating that the intended meaning is negated. However, the generated candidate sentence fails to capture this negation, instead expressing the opposite sentiment. This reversal results in a polarity error, as the output contradicts the meaning conveyed by the original AMR.

6.1.3 Deictic Gestures

Deictic Gestures errors denote mistakes stemming from the model's misinterpretation of deictic gestures in the gesture AMR. These gestures often carry ambiguous spatial references, such as pointing to a specific location, which can lead to incorrect sentence generation. Out of the 664 mistakes, 46 were labelled as deictic, which is 6.93%. An example taken from the gesture condition is shown in Figure 6.5.

```
Gesture labels:
RA: move, up; RH: into point, front;
Unknown

Gesture AMR:
(d/deixis-GA
    :ARG0 (s/signaler)
    :ARG1 (l/location)
    :ARG2 (a/actor))
Reference: "and then put one on the top"
Candidate: "point to the location"
Explanation: "The gesture label indicates that the speaker is moving
their hand up and pointing forward, which corresponds to a deixis
gesture. The speech AMR reflects this by indicating that the speaker
is performing a deixis action, with the location as the target of the
action."
```

Figure 6.5: Example from the gesture condition showcasing the Deictic Gestures error type.

This occurred frequently: whenever the gesture AMR included a deictic gesture, Llama tended to combine information from both the gesture labels and the gesture AMR. In this example, the candidate sentence includes the verb "to point", which is present in the gesture labels, and the noun "location", which is referenced in the AMR. However, these terms are not intended to be used as a verb or standalone noun, but rather serve as semantic roles within the AMR structure and gesture labels. In this particular example this behaviour is unsurprising, as the AMR provides little additional informative content beyond the deictic gesture.

6.1.4 Gesture Information

Gesture Information errors refer to any mistake made with regards to the gesture labels and gesture AMR. This category was separated from the Deictic Gestures error type because the model's outputs exhibited a high variety of distinct issues related to gestures, beyond just deictic gestures. Notably, this error type accounted for the highest number of occurrences, namely 231 out of 664 mistakes, which is 34.79%. An example taken from the gesture condition is shown in Figure 6.6.

```
Gesture AMR:
(e/emblem-GA
    :ARG0 (s/signaler)
    :ARG1 (o/ok)
    :ARG2 (a/actor))
Reference: "good enough"
Candidate: "you're okay"
Generated speech AMR:
(e/emblem-GA :ARG0 (s/signaler) :ARG1 (o/ok) :ARG2 (a/actor))
Explanation: "The gesture label indicates that the arms move up
and hands open in front, which is a common emblem for 'okay'. The
speech AMR reflects this interpretation."
```

Figure 6.6: Example from the gesture condition showcasing the Gesture information error type.

This example demonstrates one of many gesture-related errors. The model generated the sentence "you're okay" based solely on an emblematic gesture representing an "OK" sign. While the sentence is plausible in isolation, it fails to accurately reflect the intended meaning of the accompanying speech, resulting in an incorrect interpretation. Furthermore, in its explanation the model states, "The speech AMR reflects this interpretation", suggesting that it treats its own generated AMR as the gold standard, since the input did not contain speech AMR. This indicates that for the gesture condition the model bases its responses on its own output, rather than only considering the input gesture information.

6.1.5 AMR Misinterpretation

AMR Misinterpretation includes cases where the model fails to accurately interpret the meaning of the speech AMR, leading to incorrect or irrelevant outputs. This error type accounted for the second highest number of occurrences, namely 176 out of 664 mistakes, which is 26.51%. An example taken from the speech condition is shown in Figure 6.7.

```
Speech AMR:
(a/and
    :op1 (j/jiggle-01
        :time (t/then)
        :mode imperative
        :ARG0 (y/you)
        :ARG1 (i/implicit-role
            :quant 2
            :location (f/front
                :poss (t2/them)))))
Reference: "and then jiggle the two in front of them"
Candidate: "you should jiggle them in the front then"
```

Figure 6.7: Example from the speech condition showcasing the AMR Misinterpretation error type.

The AMR includes a location relation, indicating that something is positioned in front, along with a possessive modifier specifying that it is in front of them. However, the model fails to capture this nuance, entirely overlooking the possessive modifier.

6.1.6 Incoherent

Incoherent refers to all outputs that are nonsensical or fail to exhibit a logical flow, such as a disorganized structure, making them difficult to understand. This category excludes grammatically well-formed sentences that are irrelevant to the task, as the model was not explicitly informed of the task context. Of the 664 mistakes, 44 were labelled Incoherent, which is 6.63%. Two examples from the speech condition are shown in Figure 6.8.

```
Reference: "just like that but closer together"
Candidate: "you should just resemble that, but you're
having it close together more"

Reference: "the first block you put it the one nearest and
the right side of it"
Candidate: "you put the first block near most, and you have
the right side of it"
```

Figure 6.8: Two examples from the speech condition showcasing the Incoherent error type.

6.1.7 Other

Other includes all remaining errors not covered by the previous categories. This may involve annotation mistakes in the speech, or words that are missing in the speech AMR but are present in the spoken words, all of which can lead to wrongly generated sentences. Of the 664 mistakes 119 were labelled Other, which is 17.92%. Two examples are shown in Figure 6.9 and Figure 6.10.

```
Speech AMR
(g/get-01
    :ARG0 (y/you)
    :ARG1 (t/tower
        :quantity 2
        :location (o/on
            :op1 (s/sides))))
Reference: "and then you got two towers like that on the sides"
Candidate: "you get two towers on the sides"
Explanation: "The speech AMR indicates that the subject is 'you'
(ARG0), and the action is 'get' (g/get-01). The object of the
action is 'two towers' (t/tower with quantity 2), which are
located 'on the sides' (o/on with location on the sides)."
```

Figure 6.9: An example from the speech condition showcasing the Other error type

Based solely on the AMR alone, this sentence would have been correct. However the spoken words include some words that are not included in the AMR, leading to a wrong answer.

```
Gesture label(s):
arms: apart, left; hands: inch, down;
LA: move, right; LH: into closed, down; RH: into closed, left
Gesture AMR:
(i/icon-GA
    :ARG0 (s/signaler)
    :ARG1 (s2/space-01)
    :ARG2 (a/actor))
Reference: "only two of them"
Candidate: "i don't understand"
Generated speech AMR:
(g/dont-know :mode declarative :ARG0 (i/i) :mod (n/no))
Explanation: "The gesture labels indicate a lack of understanding
or confusion, with arms apart and hands inching down. The LA
moving right and LH/RH closing down suggests a sense of uncertainty
or puzzlement. In the speech AMR, 'I don't understand' is used to
convey this sentiment in a declarative mode."
```

Figure 6.10: An example from the gesture condition showcasing the Other error type

As in the example shown in Figure 6.6, the model bases its response on its own generated output. This is undesirable behaviour, highlighting that the model should not have been tasked with generating the corresponding speech AMR as well.

Out of the 573 evaluated sentences, only 41 sentences had no mistakes, which is a very low number. This indicates that Llama struggles to reliably generate speech that fully captures the intended meaning of the input. Most notably, Gesture Information and AMR Misinterpretation errors account for around 60% of all mistakes, suggesting that the model has difficulty integrating or correctly interpreting information derived from gestures, as well as AMR structure. The relatively high number of Deictic Gestures errors also reinforces this, as well as the challenge of grounding expressions such as “this” or “there” in gesture-based input. Although some error types such as Propbank and Polarity are less frequent, they still reflect critical information, often resulting in a complete reversal or distortion of meaning. The number of Incoherent outputs and Other errors highlights broader issues in fluency and model generalization.

In order to try to improve the outputs, the prompts for the three conditions were refined, which will be discussed in the next section.

6.2 Prompt Refinement

To guide Llama more effectively and with the aim of producing more accurate outputs, I constructed three new prompts, one for each condition. I informed Llama about the specific task at hand; a block-building task, and provided a short explanation of the given inputs. The three new prompts can be found in Table 6.2.

Prompt type	Prompt
Speech	You are interpreting verbal communication in a collaborative block-building task. One participant is verbally explaining how to construct a specific block structure. Your input includes: Speech AMR: A structured semantic representation of what was spoken. Your task is to: 1. Generate a sentence in natural spoken English using speech AMR that reflects what the speaker is trying to communicate. 2. Provide a short explanation describing how you interpreted the speech AMR to generate the sentence. First, read the examples to understand the Abstract Meaning Representation format:
Gesture	You are interpreting gesture information from a collaborative block-building task. One participant is explaining how to construct a specific block structure using both speech and gestures. In this task, you are given only the gesture information: Gesture Labels: Descriptions of hand, arm, or body movements. Gesture AMR: A structured semantic representation of the gestures. Although the participant was also speaking, only the gesture information is provided here. Your task is to: 1. Generate a sentence in natural spoken English using gesture information that reflects what the speaker was likely trying to communicate with the gestures 2. Provide a short explanation describing how you interpreted the gesture information to generate the sentence. First, read the examples to understand the Abstract Meaning Representation format and the gesture labels:
Speech_gesture	You are interpreting multimodal communication in a collaborative block-building task. One participant is verbally and gesturally explaining how to construct a specific block structure. Your input includes: Gesture Labels: Descriptions of hand, arm, or body movements. Gesture AMR: A structured semantic representation of the gestures. Speech AMR: A structured semantic representation of what was spoken. Your task is to: 1. Generate a sentence in natural spoken English using speech and gesture information that reflects what the speaker is trying to communicate 2. Provide a short explanation describing how you interpreted both the gesture and speech AMRs and the gesture labels to generate the sentence. First, read the examples to understand the Abstract Meaning Representation format and the gesture labels:

Table 6.2: First refined prompt created for the error analysis.

I ran these prompts on twelve sentences picked from the test set to compare them with the original prompts. To select these twelve sentences, I manually reviewed the SAGE-AMR dataset, examining both the speech AMR and the gesture AMR, along with the corresponding gesture labels. I aimed to include gesture AMRs that were relatively non-ambiguous, in order to assess whether it would lead to improved output. An example of gesture AMR that is relatively non-ambiguous is shown in Figure 6.11. Additionally, I ensured a balanced selection by including examples where the speech-only condition performed well, as well as cases where it did not.

```
(i/icon-GA
    :ARG0 (s/signaler)
    :ARG1 (s2/slide-01
        :direction (b/backward))
    :ARG2 (a/actor))
```

Figure 6.11: Example of relatively non-ambiguous gesture AMR.

I again went through the output manually, and I noted the different errors. Mistakes are again identified by examining the entire sentence and categorizing each error according to the error type it most closely resembles. For the first refined prompt, there were 107 evaluated sentences and in total there were 134 mistakes. An overview can be found in Table 6.3. The percentages are calculated based on the outputs corresponding to the twelve selected sentences.

Error Type	**Mistakes**	**Percentage**
Propbank	10	7.46%
Polarity	4	2.99%
Deictic Gestures	5	3.73%
Gesture Information	57	42.54%
AMR Misinterpretation	39	26.51%
Incoherent	5	29.1%
Other	14	10.45%

Table 6.3: Error types with their error count, also shown in percentages, for the first refined prompt.

For the second prompt I added some information regarding the speech AMR, specifying that it consists of Propbank framesets and arguments. This resulted in 131 mistakes, the scores shown in Table 6.4. The second prompt refinement showed minimal improvement over the first one.

Error Type	**Mistakes**	**Percentage**
Propbank	11	8.4%
Polarity	6	4.58%
Deictic Gestures	2	1.53%
Gesture Information	56	42.75%
AMR Misinterpretation	30	22.90%
Incoherent	7	5.34%
Other	19	14.5%

Table 6.4: Error types with their error count, also shown in percentages, for the second refined prompt.

A notable improvement was observed in the first refined prompt when a brief explanation of the task was added to the prompt. This additional context helped the model generate sentences with more task-relevant vocabulary. For example, in the initial run, the model frequently referred to the blocks using vague terms like "things", likely due to a lack of understanding about the specific objects involved. In contrast, after the prompt refinement, the model more consistently used the word "block", reflecting a better grasp of the task. For the second refined prompt, specifying that the speech AMR is based on Propbank appears to have slightly reduced the number of AMR misinterpretation errors. However, this observation should be interpreted with caution, as the error classification was performed by a single annotator, which may introduce subjectivity and affect the reliability of the results.

Chapter 7

Discussion

This thesis aimed to investigate the impact of gesture information, represented using AMR and gesture labels, on the ability of an LLM to generate speech and speech AMR, compared to only speech AMR as input. This was done by prompting Llama using three different conditions; speech, gesture, and speech combined with gesture. The output was then evaluated using three automatic evaluation metrics, as described in chapter 5. This chapter will recap these results, discuss the limitations encountered during this thesis, and what they mean with regards to the research question.

7.1 Summary

Evaluation Results This section summarizes the key evaluation outcomes discussed in chapter 5. Among the three input conditions tested, speech-only, gesture-only, and speech and gesture combined, the gesture-only condition consistently performed the worst across all evaluation metrics, while the speech-only condition outperformed the others in every condition. The overall corpus-level BLEU score across all conditions was 0.1096, the average METEOR score was 0.2855, and the average F1 BERTScore was 0.5382. Importantly, paired bootstrap sampling confirmed that these results were significant and not due to chance, giving confidence to the observed trends.

The BLEU score, which strictly relies on exact n-gram overlap between generated and reference texts, was the lowest of the three metrics. This reflects a limitation of using BLEU to evaluate NLG tasks. Even though the model can often produce semantically appropriate or paraphrased sentences, it rarely reproduces the exact reference sentence. As a result, the generated outputs often fail to match the reference sentence at the n-gram level, causing BLEU to underestimate the true quality of the output. In contrast, METEOR offers a more flexible measure by accounting for morphological variants. This leads to higher scores when the model generates paraphrases or alternative phrasing that still convey the correct meaning. In this evaluation, METEOR produced scores that were more than twice as high as BLEU, suggesting better semantic alignment than BLEU captures. Finally, BERTScore, which compares contextualized embeddings, yielded the highest scores among the three metrics. This is expected, as BERTScore is specifically designed to capture semantic similarity at the representation level, allowing for greater tolerance of paraphrasing and surface-level variation. It can assign relatively high scores even when the lexical overlap is low, as long as the underlying meanings are aligned. As with the other metrics, the speech-only condition achieved the highest BERTScore, reinforcing the conclusion that this input provides the richest and most effective information for Llama's NLG ability.

Error Analysis To further explore the results obtained in the evaluation, an in-depth error analysis was conducted on a total of 573 generated sentences, covering all three experimental conditions. This analysis aimed to discover recurring patterns of failure, and provide insight into how Llama interprets different types of (AMR) input. In order to quantify the errors, seven error types were identified. The most frequent error type was Gesture Information, which accounted for 34.8% of the errors. These cases typically involved Llama failing to meaningfully incorporate or interpret the gesture AMR and gesture labels, often resulting in vague outputs. The second most common error type was AMR Misinterpretation, accounting for 26.5% of the errors. These cases typically involved Llama failing to accurately interpret the meaning of the speech AMR, leading to semantically inaccurate sentences. The results indicate that Llama struggles with interpreting AMR and gesture information, as well as input that is ambiguous. Based on these findings the prompt was first refined to include more details about the task, i.e. that it is a block-building task, with the goal of anchoring the model's understanding and providing more context. A selection of sentences was chosen to test the refined prompt, resulting in sentences with more task-relevant vocabulary. For the second refined prompt it was specifically stated that speech AMR consists of Propbank framesets and arguments, this did not improve the first refined prompt.

7.2 Limitations

Several limitations were encountered during this thesis, primarily related to the data and its annotation. These will be discussed in detail in the following section, as well as limitations regarding the representation of the gesture information.

7.2.1 Annotations

Several issues arose due to misalignments in the annotations, particularly with regard to time stamps. One frequent problem was that the end time of the speech AMR often differed slightly from that of the corresponding gesture AMR. To address this, a temporal margin had to be introduced, which in turn introduced minor inconsistencies elsewhere. Because of time constraints these inconsistencies were not fixed but had to be included. In some cases, the AMR annotations were not properly aligned with the speech itself, which also caused problems. These challenges highlight the inherent messiness of working with dialogue data. Additionally, it is often difficult to determine precisely when one gesture ends and another begins, or to assess which gestures are communicatively meaningful. Future work could aim to improve alignment accuracy, as cleaner annotations could lead to more reliable and informative analyses.

7.2.2 Gesture Representation

During the experiments, a main question emerged regarding the suitability of representing gesture using AMR. Specifically, I question whether gesture meaning should be encoded solely through AMR structures, and whether a combination of AMR and gesture labels is appropriate for this task. The model often struggled to meaningfully integrate the gesture AMR with gesture labels. When gesture information was combined with speech AMR, the model appeared to draw partially from both sources, but without fully combining them, often resulting in only partially correct outputs. This suggests that, for this task, an alternative form of gesture representation may be more effective, perhaps one that visualizes gestures more directly. The current transcription approach does not seem to adequately capture the intended meaning conveyed by the gestures. Videos or images might better capture the intended meaning of gestures. In the current task, participants were able to communicate verbally, which likely influenced the types of gestures they used. It could be valuable to compare this setting with one in which participants are restricted to gestural communication only. Such a condition might elicit more informative gestures, which could, in turn, enhance the model's ability to interpret them accurately. However, this could take away from the naturalness of the gestures, which is a key strength of the EGGNOG corpus.

7.2.3 Model

In this section, I briefly discuss the LLM used for the experiments. Specifically, the model employed was Llama, an LLM that has demonstrated strong performance across a range of NLU and NLG tasks. However, while analyzing the model's generated outputs and accompanying explanations, it became evident that Llama lacks sufficient domain-specific knowledge of Propbank and AMR frameworks. This limitation was apparent in the model's frequent misinterpretation of AMR structures, Propbank rolesets, and its tendency to produce inaccurate sentences. To confirm that there is a lack of Propbank knowledge, Llama was prompted to explain the distinction between the Propbank framesets `look-01` and `look-02`. The model responded by stating that "`look-01` represents the verb sense of looking at or observing something with one's eyes" whereas "`look-02` represents the verb sense of searching for something, often implying a more active or intentional search". While partially correct, these explanations do not reflect the two distinct verb senses defined by Propbank, suggesting that the model does not have a reliable understanding of frameset distinctions. This response reinforces the conclusion that Llama has not internalized the semantic nuances necessary to differentiate between closely related verb senses, which is a core requirement for accurate AMR-to-text generation. Consequently, it is unlikely that the model can consistently produce correct sentences based solely on AMR inputs. Given these constraints, I believe future work should explore the use of LLMs that are explicitly pre-trained or fine-tuned on AMR-to-text tasks. Such models are more likely to have internalized the structural conventions of AMR graphs and the distinctions between various Propbank rolesets. This may result in more semantically accurate NLG.

Chapter 8

Conclusion

This thesis investigates the NLG ability of an LLM, Llama, using structured semantic input in the form of speech AMRs, gesture information, or a combination of both. The study aimed to determine whether the inclusion of gesture information meaningfully contributes to the model's performance in this generation task. Across the employed evaluation metrics, BLEU, METEOR, and BERTScore, the findings consistently show that the speech-only condition yielded the highest scores, while the gesture-only condition performed the worst. The multimodal condition did not significantly outperform the speech-only baseline. These results lead to several key conclusions. Firstly, gesture information did not improve Llama's NLG ability. This outcome can be attributed in part to the limitations of the LLM itself. Llama was not trained on AMR-specific data and therefore does not show a strong understanding of Propbank rolesets. Both aspects are crucial for accurately interpreting the semantic structures presented in the prompts. Without this foundational knowledge, the model struggles to map AMRs to coherent sentences. Secondly, the way gestures were represented through gesture labels and gesture AMRs may not have been sufficiently informative or detailed for the model to leverage effectively. In particular, deictic gestures tend to be ambiguous without contextual grounding, and when isolated from accompanying visual or spatial clues, offer limited semantic value. However, these limitations do not prove that gesture lacks communicative value. Future work could explore the use of LLMs that have been explicitly trained or fine-tuned on AMR-to-text tasks.

In summary, while gesture information did not enhance the NLG ability in this thesis, it could be a consequence of model limitations. With more suitable model architectures, the integration of gesture into semantic generation remains an interesting field for future research.

Appendix A

Appendix

<table>
<tr><th>Prompt type</th><th>Prompt</th></tr>
<tr><td>Gesture</td><td>Given the following gesture label(s) and gesture Abstract Meaning Representation (AMR), generate a corresponding sentence and speech AMR in natural spoken English. Provide a short explanation. First, read the examples to understand the AMR format:
Example 1:
Sentence: and you can you can place them close to each other
Speech AMR:
<pre>(a/and
    :op2 (p/possible-01
        :ARG1 (p2/place-01
            :ARG0 (y/you)
            :ARG1 (t/them)
            :purpose (c/close-10
                :ARG1
                :ARG2 (o/other
                    :mod (e/each))))))</pre>Gesture label(s):
hands: into open, back;
hands: beckon, back;
LH: into closed, right; RH: into closed, up;
arms: apart, left; RH: into closed, left;
hands: shake, left;
Gesture AMR:
<pre>(i/icon-GA
    :ARG0 (s/signaler)
    :ARG1 (c/close)
    :ARG2 (a/actor))</pre>Example 2:
Sentence: it cannot fall down from the ground right
Speech AMR:
<pre>(p/possible-01
    :polarity -
    :ARG1 (f/fall-01
        :ARG1 (i/it)
        :ARG3 (g/ground))
    :ARG1-of (r/request-confirmation-91))</pre>Gesture label(s):
head: shake; hands: rotate, open, up;</td></tr>
</table>

Continuation of the table above

Prompt type	Prompt

Gesture	Gesture AMR:
	
	`(d/deixis-GA`
	`    :ARG0 (s/signaler)`
	`    :ARG1 (b/block)`
	`    :ARG2 (a/actor))`
	
	Now, generate a sentence and the corresponding speech AMR from the following gesture label(s) and gesture AMR and explain your reasoning. Please provide the output only in json format:
	`[{"sentence": "Your generated sentence here.",`
	`[{"speech AMR": "Your generated speech AMR here.",`
	`"explanation": "Your explanation here."}]`
	Speech AMR:
	
	`(p/put-01`
	`    :mode imperative`
	`    :ARG0 (y/you)`
	`    :ARG1 (b/block`
	`        :quant 1)`
	`    :ARG2 (b2/block`
	`        :mod (b3/back)))`

Table A.1: Full prompt for the gesture condition, test sentence “so put put a block on the back block good”.

Prompt type	Prompt
Speech+gesture	Given the following gesture label(s), gesture Abstract Meaning Representation (AMR) and speech AMR, generate a corresponding sentence in natural spoken English. Provide a short explanation. First, read the examples to understand the Abstract Meaning Representation format:

Example 1:
Sentence: on the side with three the one closest to the middle put another block on top
Speech AMR:

```
(p/put-01
    :mode imperative
    :ARG0 (y/you)
    :ARG1 (b/block
        :mod (a/another))
    :ARG2 (o/on-top
        :op1 (o2/one
            :ARG1-of (h/have-degree-91
                :ARG2 (c/close-10
                    :ARG1 o2
                    :ARG2 (m/middle))
                :ARG3 (m2/most))
            :location (o3/on
                :op1 (s/side
                    :ARG0-of (h2/have
                        :ARG1 (i2/implicit-role
                            :quant 3)))))))
```

Gesture label(s):
body: still;
LA: move, left; LH: into open, front;
body: still;
LH: into point, front;
body: still;
Gesture AMR:

```
(d/deixis-GA
    :ARG0 (s/signaler)
    :ARG1 (s2/side)
    :ARG2 (a/actor))
(d/deixis-GA
    :ARG0 (s/signaler)
    :ARG1 (b/block)
    :ARG2 (a/actor))
```

Example 2:
Sentence: now these are a little jiggled
Speech AMR:

```
(j/jiggle-01
    :ARG1 (t/these)
    :mod (l/little))
```

Gesture label(s):
head: nod; LA: move, back;
RA: move, up; RH: shake, down, open;
RA: move, front; RH: into point, down;

Continuation of the table above

Prompt type	Prompt

Speech+gesture	Gesture AMR:
	`(e/emblem-GA`
	`    :mode expressive`
	`    :ARG0 (s/signaler)`
	`    :ARG1 (y/yes)`
	`    :ARG2 (a/actor))`
	`(i/icon-GA`
	`    :mode expressive`
	`    :ARG0 (s/signaler)`
	`    :ARG1 (j/jiggle-01)`
	`    :ARG2 (a/actor))`
	`(g/gesture-unit`
	`    :op1 (i/icon-GA`
	`        :mode expressive`
	`        :ARG0 (s/signaler)`
	`        :ARG1 (j/jiggle-01)`
	`        :ARG2 (a/actor))`
	`    :op2 (a1/and`
	`        :op1 (d1/deixis-GA`
	`            :ARG0 s`
	`            :ARG1 (b1/block)`
	`            :ARG2 a)`
	`        :op2 (d2/deixis-GA`
	`            :ARG0 s`
	`            :ARG1 (b2/block)`
	`            :ARG2 a)))`
	Now, generate a sentence from the following gesture label(s), gesture AMR and speech AMR and explain your reasoning. Please provide the output only in json format:
	`[{"sentence": "Your generated sentence here.", "explanation": "Your explanation here."}]`
	Gesture label(s):
	RA: move, front; RH: point, front; RA: move, down; RH: into claw, down; RA: move, up; RH: into point, front; Gesture AMR:
	`(d/deixis-GA`
	`    :ARG0 (s/signaler)`
	`    :ARG1 (l/location)`
	`    :ARG2 (a/actor))`
	Speech AMR:
	`(p/put-01`
	`    :mode imperative`
	`    :ARG0 (y/you)`
	`    :ARG1 (b/block`
	`        :quant 1)`
	`    :ARG2 (b2/block`
	`        :mod (b3/back)))`

Table A.2: Full prompt for the speech and gesture condition, test sentence “so put put a block on the back block good”.

References

X. Bai, Y. Chen, and Y. Zhang. Graph pre-training for amr parsing and generation. *arXiv preprint arXiv:2203.07836*, 2022.

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, 2013.

S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

C. Bonial, L. Donatelli, M. Abrams, S. Lukin, S. Tratz, M. Marge, R. Artstein, D. Traum, and C. Voss. Dialogue-amr: abstract meaning representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, 2020.

R. Brutti, L. Donatelli, K. Lai, and J. Pustejovsky. Abstract meaning representation for gesture. In *Proceedings of the thirteenth language resources and evaluation conference*, 2022.

S. Cai and K. Knight. Smatch: an evaluation metric for semantic feature structures. In H. Schuetze, P. Fung, and M. Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL `https://aclanthology.org/P13-2131/`.

J. Cassell, S. Kopp, P. Tepper, K. Ferriman, and K. Striegnitz. Trading spaces: How humans and humanoids use speech and gesture to give directions, 2007.

Z. Cheng, Z. Li, and H. Zhao. Bibl: Amr parsing and generation with bidirectional bayesian learning. In *Proceedings of the 29th International conference on computational linguistics*, pages 5461–5475, 2022.

Coqui. English stt v1.0.0. technical report stt-en-1.0.0, coqui, 2021. URL `https://coqui.ai/models`.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

L. Donatelli, K. Lai, R. Brutti, and J. Pustejovsky. Towards situated amr: Creating a corpus of gesture amr. In *International Conference on Human-Computer Interaction*, pages 293–312. Springer, 2022.

P. Ekman and W. V. Friesen. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *semiotica*, 1(1):49–98, 1969.

F. Errica, G. Siracusano, D. Sanvito, and R. Bifulco. What did i do wrong? quantifying llms' sensitivity and consistency to prompt engineering. *arXiv preprint arXiv:2406.12334*, 2024.

H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 944–952, 2010.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Self-published, 3rd edition, 2025. URL `https://web.stanford.edu/~jurafsky/slp3/`. Online manuscript released January 12, 2025.

A. Kendon. How gestures can become like words. In *This paper is a revision of a paper presented to the American Anthropological Association, Chicago, Dec 1983.* Hogrefe & Huber Publishers, 1988.

A. Kendon. Do gestures communicate? a review. *Research on language and social interaction*, 27(3):175–200, 1994.

A. Kendon. *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

P. R. Kingsbury and M. Palmer. From treebank to propbank. In *LREC*, pages 1989–1993, 2002.

K. Lai, R. Brutti, L. Donatelli, and J. Pustejovsky. Encoding gesture in multimodal dialogue: Creating a corpus of multimodal amr. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5806–5818, 2024.

H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 12 2013. ISSN 0891-2017. doi: 10.1162/COLI_a_00152. URL `https://doi.org/10.1162/COLI_a_00152`.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1:3, 2020.

E. Manning, S. Wein, and N. Schneider. A human evaluation of amr-to-english generation systems. *arXiv preprint arXiv:2004.06814*, 2020.

C. M. Matthiessen and J. A. Bateman. Text generation and systemic-functional linguistics: experiences from english and japanese. *(No Title)*, 1991.

D. McNeill. So you think gestures are nonverbal? *Psychological review*, 92(3):350, 1985.

D. McNeill. *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, Chicago, 1992.

D. McNeill. Gesture and thought chicago: Univ, 2005.

D. McNeill and S. D. Duncan. Growth points in thinking-for-speaking. *Language and gesture*, 1987:141–161, 2000.

S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, and R. Socher. Large language models: A survey, 2025. URL `https://arxiv.org/abs/2402.06196`.

J. Opitz, L. Parcalabescu, and A. Frank. Amr similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538, 09 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00329. URL `https://doi.org/10.1162/tacl_a_00329`.

T. O'Gorman, M. Regan, K. Griffitt, U. Hermjakob, K. Knight, and M. Palmer. Amr beyond the sentence: the multi-sentence amr corpus. In *Proceedings of the 27th international conference on computational linguistics*, pages 3693–3702, 2018.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

M. Popović. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395, 2015.

M. Popović. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618, 2017.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-6319`.

Z. Ren, J. Meng, J. Yuan, and Z. Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760, 2011.

R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

A. B. Sai, A. K. Mohankumar, and M. M. Khapra. A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39, 2022.

E. Saravia. Prompt Engineering Guide. *https://github.com/dair-ai/Prompt-Engineering-Guide*, 12 2022.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

I. Wang, M. B. Fraj, P. Narayana, D. Patil, G. Mulay, R. Bangar, J. R. Beveridge, B. A. Draper, and J. Ruiz. Eggnog: A continuous, multi-modal data set of naturally occurring gestures with ground truth labels. In *2017 12th Ieee international conference on automatic face & gesture recognition (fg 2017)*, pages 414–421. IEEE, 2017.

A. Webson and E. Pavlick. Do prompt-based models really understand the meaning of their prompts? In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.167. URL `https://aclanthology.org/2022.naacl-main.167/`.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. Elan: A professional framework for multimodality research. In *5th international conference on language resources and evaluation (LREC 2006)*, pages 1556–1559, 2006.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

R. Zhao, P. Kenny, P. Labute, and D. O'Shaughnessy. Issues in large scale statistical language modeling. In *3rd European Conference on Speech Communication and Technology (Eurospeech 1993)*, pages 965–968, 1993. doi: 10.21437/Eurospeech.1993-227.