On Polysemy and Bias

A Master's Thesis

Presented to

The Faculty of the Graduate School of Arts and Sciences
Brandeis University

Graduate Program in Computational Linguistics
Department of Computer Science

James Pustejovsky, Advisor

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by

Sunny Zhou

May 2025

ACKNOWLEDGMENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed dapibus metus in erat varius tincidunt. Nulla diam neque, fringilla a interdum id, vehicula id erat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Mauris elementum odio orci. Nam at velit id augue tempus hendrerit. Fusce vestibulum purus eget eros semper, et consectetur libero pellentesque. Phasellus ultricies malesuada erat fringilla luctus. Sed vel sapien vitae diam vestibulum imperdiet non eget nisi.

Fusce suscipit eros ligula. Mauris faucibus in massa nec malesuada. Mauris sit amet vestibulum nibh. Phasellus libero nisl, viverra id tempor vel, ornare id tortor. Nunc at sem a leo tempus dictum. In sed dignissim ipsum. Aenean condimentum vestibulum ligula, et viverra urna. Mauris in risus et nulla lobortis pretium ac a neque. Etiam porttitor enim nec ornare placerat. Proin faucibus nibh eu magna convallis pellentesque.

ABSTRACT

On Polysemy and Bias

A thesis presented to the Faculty of the
Graduate School of Arts and Sciences of Brandeis University
Waltham, Massachusetts

By Sunny Zhou

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed dapibus metus in erat varius tincidunt. Nulla diam neque, fringilla a interdum id, vehicula id erat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Mauris elementum odio orci. Nam at velit id augue tempus hendrerit. Fusce vestibulum purus eget eros semper, et consectetur libero pellentesque. Phasellus ultricies malesuada erat fringilla luctus. Sed vel sapien vitae diam vestibulum imperdiet non eget nisi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Since the beginning of the study of natural language processing, this conversation about capturing the meaning in language has sprouted many attempts to translate words into numbers, now adopted in the state-of-the-art models today. A big struggle of these attempts has been how to accurate capture the ambiguous nature of language, with only one representation per word. Through the advancement of LLMs, there have been promising results on their understanding abilities especially when it comes to understanding of word meaning. In this thesis, I aim to explore aspects of probing LLM knowledge on the different use cases of words, through a series of benchmarking tasks.

Word senses in natural language often refers to what a word means in the context, which, in many cases, are the source of ambiguity. The ambiguous senses of words are categorized into two ways: homonymy and polysemy. Homonymy includes words that sound or spelled the same, but mean different things (e.g. a dog's bark and a tree's bark). On the other hand, polysemy refers to words that can mean different, but related things. The distinction between polysemous meaning can be extremely subtle.

In obvious cases, this could be a noun meaning different things, like in "I went to the bank," meaning the "financial institution" or the "bank of a river", or this could be a noun and a verb being the same word, like in "Cinnamon is a type of tree bark" and "The bark was heard from miles." However, consider the following examples for the word "fall":

(1) a. fall of a tree

b. the season of Fall

c. fall of the Roman Empire

Here, in all of the use cases, "fall" refers to a noun. Like the previous example with "bark" the comparison between a. and the other two is fairly obvious, the difference between b. and c. is subtle, with the instances in "fall of a tree" and "fall of the Roman Empire" both referring to a state

of ”falling.” Although both refer to an entity decreasing in some way, the latter is more abstract, referring to a subjective decline in prominence or power of an individual instead of a positional decrease. This difference in subtly boils down to the difference between polysemy and homonymy. It should be clear that this is incredibly important when talking about meaning.

Due to the nature of how LMs process words and the complexity distinguishing between such words, the distinction of these two are often ignored when talking about language competence, and is often taken for granted when it comes analyzing the informativeness of LLM performance. In fact, it is often assumed that LLMs understand language and, therefore, different word senses. However, considering how important this is to communication, it is crucial to establish to what extent they understand these intricacies and whether different representations of words should even matter. It is worth mentioning that there the argument to be made on whether these reasoning abilities are the result of language exposure, hinting at possible innate logic behind the structure and nature of language. Regardless of the answer to this question, it is clear (at least intuitively) that the context of the word determines largely the word's meaning. Tests for this knowledge has been focused on homonymy.

At the heart of this question of word senses is the question of determining and extracting LLM knowledge. It should goes without saying that it is essential that large language models (LLMs) are masters of the different aspects of language. But how can we be sure LLMs know what a certain word means? With the advancements in LLMs today, it is becoming increasing harder to understand what the models are learning. While most researchers understand the black box nature that comes with the field, this inability to know what they know and how they think gets in the way as we try to justify and prove increasingly advanced functions of LLMs, such as pragmatic reasoning (e.g. Theory of Mind) (van Duijn et al., 2023), mathematical reasoning (Mirzadeh et al., 2024), and logical reasoning (Wei et al., 2023). Ideally, model weights and architectures would be built to be interpretable, meaning built with extractable learning patterns, but due to the sheer size of the weights, number of heads, and proprietary gate-keeping, this is not realistic or feasible.

Thus, current methods of testing rely on using performance metrics to equate to what to some type of lower-level competence. The justifications for how well these methods depict understanding is a continuous question, but they can provides good hints as to what might be happening.

Measuring the extent of this knowledge can be approached in two ways: 1. prompting or 2. embeddings. We can view prompting as asking the LLM to do a task for us and interpreting its output. The simplest way of extracting what LLMs know about word senses is directly asking the LLM, "What does this word mean in this context?" Current methods of benchmarking and testing LLM knowledge has largely comprised of testing on questions with fixed answer (e.g. multiple choice) and evaluating scores such as accuracy and F1. From education in general, we know that this method of testing can demonstrate a certain level of competency of certain tasks, but often plays the risk of memorization. When it comes to testing human knowledge this tends to be less of a concern. In this case, LLMs have been shown to be able to memorize structures and answers of questions (). Additionally, performance on multiple choice questions relies on the whether correct answers are provided and

The use of iterative approaches to test this have since carried the assumption that LLMs can incorporate previous knowledge and judgments into future predictions. However, literature suggests that this may not be the case (Dziri et al., 2023). Although the studies present convincing arguments as to why we should be skeptical towards the reasoning and judgment abilities of the LLMs, the tasks chosen (e.g. graph coloring) uncover less about LLM reasoning abilities towards language tasks. Then how does one break apart what an LLM knows?

The second way of gathering information from LLMs is through extracting embeddings. Embeddings are vectors that capture a unit of language's meaning. Often times these represent subwords, but can represent words, sentences, etc. depending on the implementation and usage. Embeddings can be thought of as a mapping from language to some geometric space, supposedly preserving the relationships between language. The training of word embeddings relies heavily on the mantra "You shall know a word by the company it keeps." By relying on building these

representations on what other words are near, embeddings by design are capturing a collection of contexts. In particular, current models utilize similar methods to calculate a base embedding, which goes through additional layers of contextualization (i.e. attention heads). As methods of calculating and training embeddings develop in sophistication, more contextual information is used to captured, such as positional information and what neighboring words are important (i.e. Key, Query, and Value in attention) (Vaswani et al., 2023). Mathematically there are a variety of ways to to compare vectors, such as clustering and Euclidean distance. Considering all of this, it is stands to reason that comparing these embeddings after contextualization through a model architecture should give a glimpse into how a model might process meaning.

This thesis will comprise of two experiments, utilizing these existing methods to uncover lower-level understanding: prompting and contextual embeddings. First, I use iterative prompting to show how open-ended tasks can be used to isolate the possible gaps in LLM understanding. The goal of this approach is suggest a departure from the traditional fixed-answer benchmarking. With this approach, I aim to address a concern regarding of data contamination, which as LLM training data increases becomes more of an issue. I will discuss how the tasks in this experiment are designed to minimize the chances accessing of memorized content and structure, minimizing data contamination. Thus, I designed the prompts used in this section to ask for an open-ended response. A continuous consideration throughout this experiment is justifying whether the LLM responses reflect this underlying knowledge of the content or the knowledge of a task. In other words, how can we be confident that an incorrect answer demonstrates the LLMs inability to understand the task or the meaning of the words? Thus, the challenge when evaluating open-ended responses is determine whether content, such as explanations, efficiently. As a solution, I utilize LLM-as-a-judge for evaluation, which posed uniquely difficult prompting challenges. I found that, despite what some literature suggests, some models were unwilling to provide ratings for the response.

Second, I used the words, examples, and prompts used in experiment 1 and extracted contextual embeddings of the words in an example sentence. These embeddings were then dimension reduced

using PCA to 2 dimensions and clustered using K-Means. The goal of this section is to find possible justifications as to why the models do better on particular tasks than others.

The challenging part of is question is determining and justifying what knowledge performance on these tasks may tell us. We can first observe how polysemy is captured on understood in humans. It has been observed in children that new context or understand to polysemic words results in changes to the understanding (Srinivasan et al., 2019). Although LMs do not necessarily update their weights based on new information, this draws similarities to contextual embeddings and how they will changes with new information in the context window.

This work has many different implications about language, intelligence, and learning, which I will discuss later. In particular, the debate regarding how much context should influence word meaning has been debated (Grindrod, 2024) and how much insight LLMs can provide on the nature of language. Like Grindrod (2024), I would not like to fully dismiss any one of these stances; however, I aim to provide insight as to how each approach may be used.

RELATED WORK

In-Context Learning

LLMs have become the classic black box problem. Although, we understand what they learn from, through the careful design and annotation of training data, and how they learn, through the design of training pipelines and algorithms (Kingma and Ba, 2017; DeepSeek-AI et al., 2025), there is still much to know about what they learn. The general approach of interacting with LLMs to probe their knowledge have been mainly about what information can we prompt out of them. A lot of work has been trying to figure out what the best way of prompting the information. The naive approach of asking simple questions, such as ”What is the weather like today?” or ”What is the meaning of life?” can yield varying degrees of performance. For more compute heavy tasks, such as tasks that require a higher level of reasoning, these simple prompts are subpar.

Few-shot learning means providing LLM an example of an answer within the prompt (Brown et al., 2020). This has been shown to help LLMs to provide better formatted, more accurate response

Additionally, LLMs have been used to critique viability of answers (Stechly et al., 2023; Zheng et al., 2023). These studies suggest that LLM prompts can

Contextualism in LLMs

Well for starters, it is intuitive to think about sentences as groupings of words and, thus, the meaning of a sentence should be understood as the combinations of the included words. Starting from 2000's, work on semantic natural language processing have reasoned that word meanings can be captured within vectors (Bengio et al.).

Transformers, namely attention mechanism, has revolutionized how LLMs process inputs (Vaswani et al., 2023).

Embeddings have thus spearheaded this contextualism argument.

LLM Benchmarks

Many of the current LLMs – for the benefit of increased grammaticality and increased connectivity between previous predictions and future ones – have been designed to perform tasks categorized as CausaulLM, meaning they predict the most probable words in the sequence. These LLMs generate words (or tokens)

An interesting idea posed by differences between LLMs and humans is this idea that we utilize both in different ways. In other words we can classify their intelligences differently.

We use these tests to argue the knowledge of an LLM, extrapolating to capabilities of reasoning or thought. Similar studies, then make the argument that the solution to these performance shortcomings are using methods of in-context learning and prompt engineering (Brown et al., 2020; Wei et al., 2023). Given the well know black-box nature of LLMs, it is nearly impossible to disprove this notion. With that being said, I find it, also, difficult justify and be convinced that such benchmarking tasks signify more than just embedding level understanding. Here, I attempt not to disprove either approach, but to suggest linguistically motivated advances to begin to understand how LLMs process the word meaning and the questions presented to them.

This first experiment is to gather an understanding of what aspects of a word's definition do LLM know. Through this, we can then determine how much LLMs know about polysemy and homonymy. The difficulty of this question determining what sort of task to prompt LLMs with and what the chosen tasks tell use about polysemy and homonymy.

Corpus

The corpus used for this experiment comes from WordNet (Miller et al.). What makes WordNet so powerful and useful for my purpose is that it is a dictionary built on lexical entries that are organized by word meaning; in other words, semantic roles, instead of alphabetical. Additionally by design WordNet is written in an easily computer searchable format. Words are organized into four categories: nouns, verbs, adjectives, and adverbs. Words in WordNet are organized in sets of synonyms, known in WordNet as synsets. Each synset is categorized by a ”main” word acting as the head of the set. This creates a hierarchy, in which the noun case is a topical. This mean the top of the noun synset tree is the most representative of that topic. This design of WordNet allows easy look up of all of the synsets a certain word belongs to. For many words, the synsets they belong to spans multiple parts of speech (POS). For example, the word ”fall” belongs to synsets Synset('spill.n.04') and Synset('descend.v.01'), only to name a couple.

Due to computational limitations, this experiment was limited to the lookup of the nouns of WordNet. Per the version of WordNet that I am using there are 82,115 total synsets. Specifically, these nouns were then filtered for the 1000 most common nouns according to Corpus of Contemporary American English's (COCA) word frequency data (). Nouns were chosen as a good starting point since they are widely used as examples for ambiguous sentences and give a good representation for how an LLM understands meaning. Additionally, since the synsets listed for each word includes synsets belonging to a variety of POS, this means the data is not specifically limited to nouns.

Tasks

Fixed answer tasks, such as True or False questions, limit how the LLM can answer and play the risk of memorization. Unless the answers a carefully chosen to avoid data contamination, certain constructions, such as the definitions and sentences used here, can lead the LLMs to the sources and regurgitate their training data. Thus, the goal of these tasks should enforce a sort of novel-ness to the LLMs answer, minimizing any mention to possible source data in order to test for what the LLM has learned. To address this concern, I decided to pursue an open ended approach, asking the LLM to generate any answer. By doing so, this should allow the models more freedom to justify and reason about the problem. However, this poses several challenges. First, in cases of poor responses, it is impossible to be certain whether the response is due to a lack of task understanding As mentioned in the introduction, it is extremely important to choose

A series of tasks were chosen to narrow down the possible response interpretations. Thus, conclusions made about the LLMs performance does not rely on only one failed task. The tasks chosen were he following tasks[1]:

1. Determine if a definition fits a word in a given context.
2. Provide a definition for a word in a given sentence
3. Identify what words are important for a definition for a word.
4. Replace a word in a given sentence.

Parallels from these tasks can be made to what a good understand of a word's meaning should reflect: definition suitability judgments, definition generation, definition importance, and eventual word usage.

[1]Templates for these tasks can be found in the Appendix

Architectures

I tested four LLM architectures using the approach listed above: Llama 3.2 4B, Mistral 7B v0.3, Gemma 3 4B, and DeepSeek R1 Distill Llama. All of the architectures used are instruction tuned, except for DeepSeek R1 Distill Llama, to maximize the chances of task comprehension. These models were utilized using vLLM and the existing off-the-shelf pretrained weights from Huggingface. Although larger models would have been more desired, due to computational limitations, I had to resort to smaller models. These models were also used as LLM-as-a-judge. It is worth noting that results judged by the same model that generated the responses will be inflated due to a self answer bias. For the sake of curiosity and limited by compute, Gemma-3-12b-it was ran on a portion of the tasks and judged in the same way as the others.

Evaluation

The problem with using the open ended responses to test LLM knowledge is evaluation. Since there are innumerable ways to answer any given question, ”correctness” can be subjective; especially, considering the tasks at hand. For instance, responses for the task of identifying the keywords from a definition could differ drastically, depending how it quantifies the importance of a word. This could include considering the consquences on the definition if the keyword is omiited or the considering if the keyword is a synonium of the target word. Thus, the longstanding method of evaluation for these tasks has required manual annotation. However, human annotation is costly, with respects to money and time. Instead, I break up my evalutation in two parts. I chose a subset of the responses, randomly sampling 20 from each task, and enlisted volunteers manually annotate the responses. The goal for annotation a small subset of responses was by no means meant to indicate any final conclusion about performance. Instead, this provided a good baseline judgement of the common pitfalls of model performance, used as sanity check for the second part.

Second, I utilize LLM-as-a-Judge (Zheng et al., 2023) to judge the model outputs, which allowed me to avoid manual annotation. This idea of using LLM-as-a-Judge is relatively new and requires substantial confidence testing. Thus I used multiple models to illustrate consensus among judges. In order to empirically evaluate the responses, the judges were asked to provide a rating of each response on a scale of 1 to 10 (i.e. 1 being the worst response and 10 being a perfect response). A perfect response must satisfy the following criterion: helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response (Zheng et al., 2023). These criterion along with a strict format guide was fed into the LLM as a system prompt, modeled after Zheng et al. 2023. This choice of utilizing a premade prompt was because it was already tested to yield human like response.

The responses of the four models were tested as judges: Llama 3.2 4B, Mistral 7B v0.3, Gemma 3 4B, and DeepSeek R1 Distill Llama. Please note that these are the same models used to create

Architecture	Bad Outputs (0-shot)	Bad Outputs (1-shot)
Llama-3.2-3B-Instruct'	78,879	57
Mistral-7B-Instruct-v0.3	189,697	1,772
Gemma-3-4b-it	42,784	1,635
DeepSeek-R1-Distill-Llama-8B	274,336	69,998

Table 1: Table of Number of Output Missing Ratings

the responses being judged.[2]

Getting viable outputs in of itself proven to be a struggle even with the templated prompts. The key to a good LLM-as-a-judge is writing a prompt that is able to tease out a rating. From initial prompting, I found that the models had a preference of outputting only explanation. Even after adding stronger modals, such as "must," and emphasizing excluding explanation all together, I found that this led to inconsistent performance and some models were not willing to generate a rating. For these tests many of the LLM-as-a-Judge responses lacked a rating. So, I added a reasonable response to act as an example answer, serving as one-shot learning example. Moreover, each response was checked for the rating using the same method as the evaluation. If no rating was gathered, the LLM was prompted again with the same prompt. This helped tremendously at reducing the bad outputs as indicated in 1.

[2]Ideally, this would not have been the case and an independent fourth model would have been chosen. However, due to computational resource limitations, larger models were not used for this.

Results

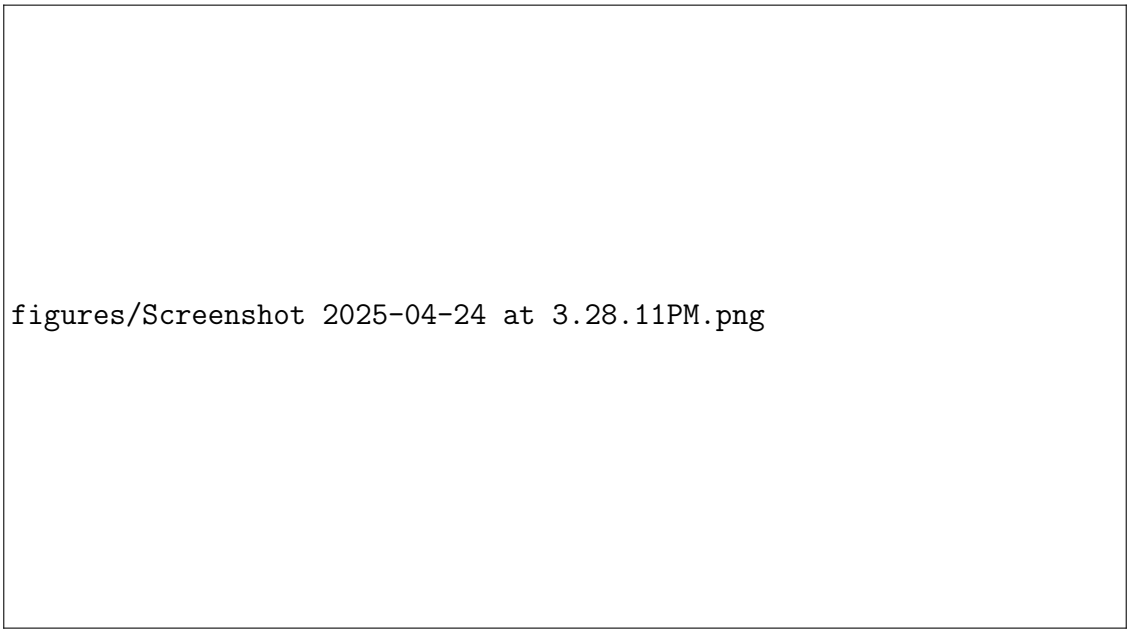

Figure 1: Graph of LLM-as-a-Judge with Llama-3.3-3b, Gemma-3-4b, DeepSeek-R1, and Mistral-7b (One-Shot and Grouped by Task) Results

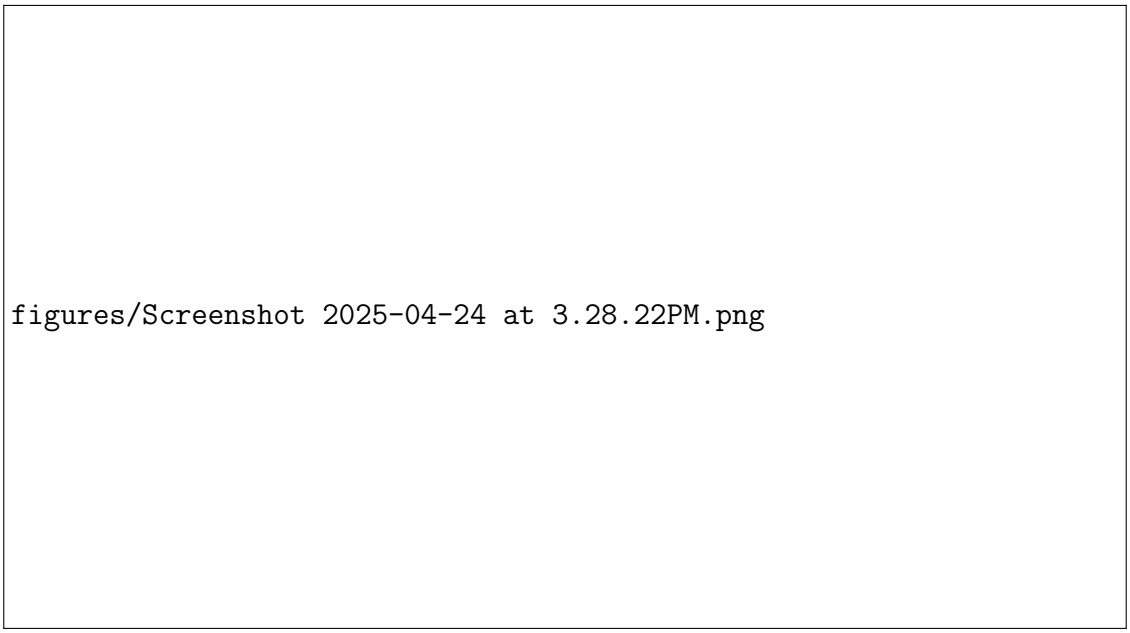

Figure 2: Graph of LLM-as-a-Judge with Gemma-3-4b and Gemma-3-12b (One-Shot and Grouped by Task) Results

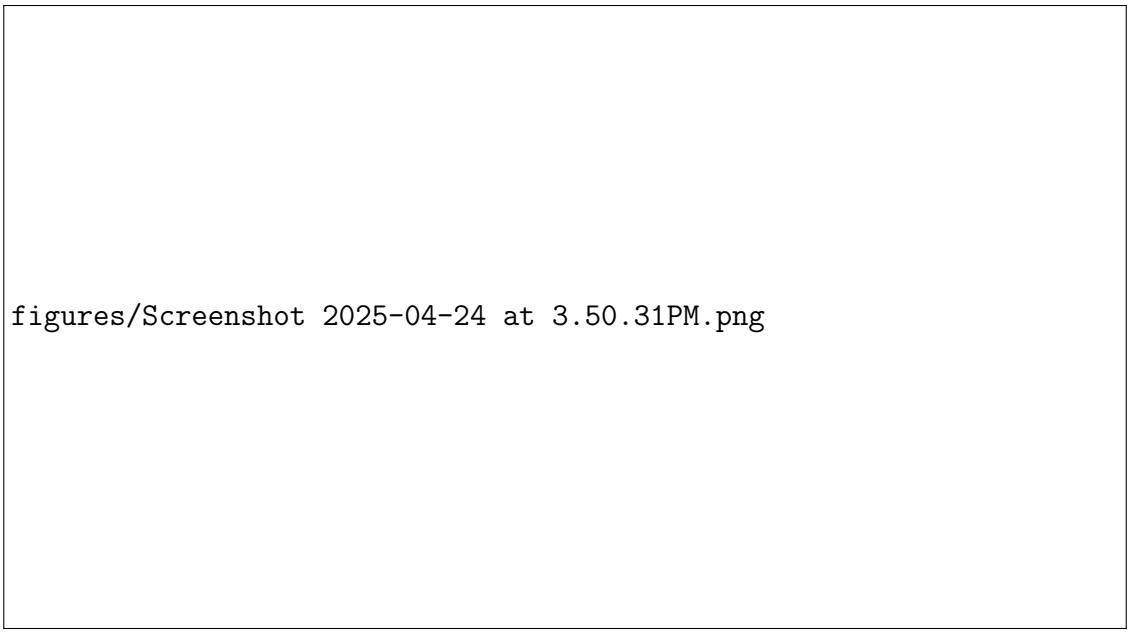

Figure 3: LLM-as-a-Judge with Task 2, 3, and 4 (One-Shot and Grouped by Assistants) Results

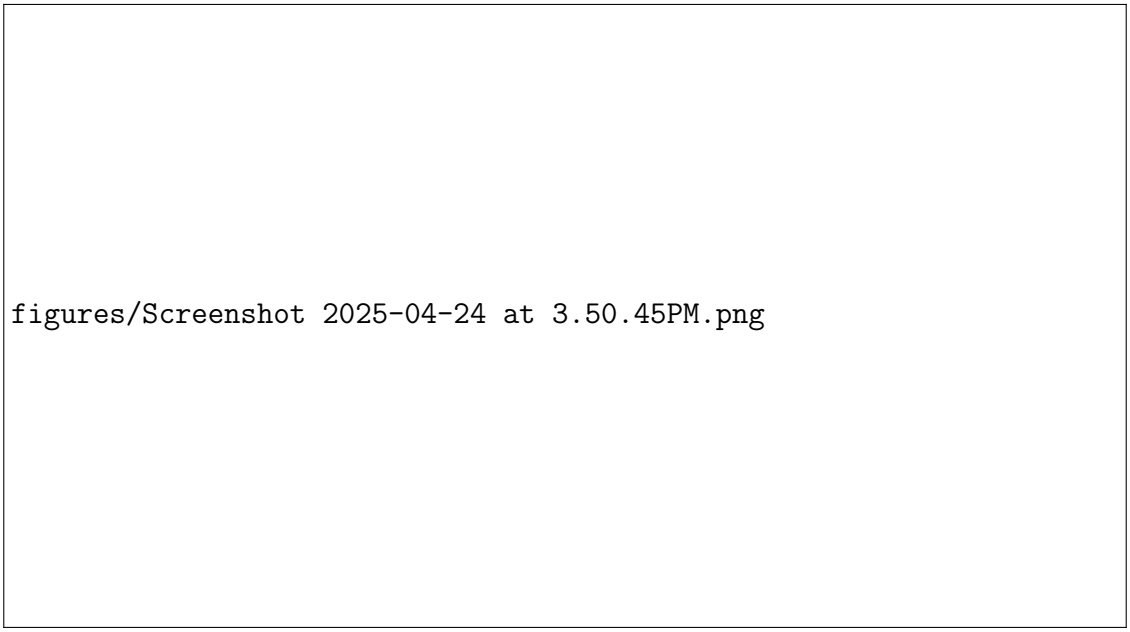

Figure 4: LLM-as-a-Judge with Task 1 (One-Shot and Grouped by Assistants) Results

Judging Performance

Using the human annotations, I found a few interesting characteristics about the model's preferences. On the surface, it would seem as though thte s

As shown by Figure 1, there are two clear winners and two clear losers across the board, regardless of LLM judge. Consistently the Gemma and Mistral both scored the higher than DeepSeek and Mistral. As expected, we see that some LLMs (i.e. Gemma and Mistral) had preferences towards their own responses. While DeepSeek, for all of the tasks judged itself to be the worst in

Task Performance

When evaluating the results of the models, it must be considered the bias of the judges. To minimize the effect the concern for judge validity, comparison across judges has to be scaled

Llama perfromance

Looking at DeepSeek's over all performance, it performed the worst on task 1

As mentioned earlier, I also compared Gemma 3 4B with Gemma 3 12B. The results suggest that both models performed similarly.

Discussion

The results The subpar performance of Deepseek

TOKEN EMBEDDINGS SEMANTICS

When studying the word embeddings of pretrained models, words are typically split up into smaller segments (i.e. byte pair encodings), which makes it hard to decipher what knowledge is being captured by the embeddings. Adopting the approach given by Chronis and Erk 2020, contextualized word embeddings are gathered from layer in the LLM architectures and clustered using K-means.

Corpus

The data used in this section is the data generated by the tasks and the responses from the previous experiment. Specifically, the data used here were the definition, example sentence, the prompt used in each task, and the response that was given. Each response was used as is an not

Methods

For each data instance, two embeddings were generated for each target word: one in the context of the example sentence and one in the context of the prompt. These embeddings were extracted from passing the base embeddings of the sentence (and prompt) tokens forward through each LLMs attention heads. Then the weights from the last hidden state was extracted and treated as the contextualized embeddings of every token. To find the token that would be the most representative in the context, the first token that is contained in the target word (or vice versa) is used as the contextual embedding of the word. Additionally, I extracted an averaged embeddings of the definition, example sentence, the prompt used in each task, and the response that was given. Like the contextualized embeddings, each of these sentences were fed through the LLMs and the last hidden layer was extracted. However, instead of mapping these embeddings to their respective the encoding tokens embeddings to gather target word embeddings, these weights were averaged to generalize what how the LLM may interpret the entire sentence meaning. Of course, the same models were used to embed the data. Each model was only used to embed their own responses because the goal is to use this method to reflect LLMs may be thinking.

After gathering the embeddings, they were grouped by embedding type, LLM, and task type from the previous experiment. The number of clusters were uTo get a better visualization of the clusters, principle component analysis was used to provide a dimensionally reduced representation.

Results

Discussion

LIMITATIONS AND FUTURE DIRECTIONS

A large assumption made in this study is the sense annotation from WordNet. However, this is also not to say that WordNet annotations are perfect for this purpose. Word sense ambiguity is a complex issue, which is still not fully agreed upon in some cases. This is reflected in the WordNet definition annotation. For example for the word in ”move” in ”The director moved more responsibilities onto his new assistant”, the given definition by WordNet is ”cause to move or shift into a new position or place, both in a concrete and in an abstract senses.” Although this is a valid definition for the word, it groups together the abstract and concrete differences in the words. Some may argue that this is deserve two separate senses. With that being said, this is by no means the fault of WordNet. The purposes for WordNet is to classify words based on word usage,

ASSISTANT: A. To change position or place. B. To put something somewhere. C. To shift something from one place to another. D. To perform a physical action. The best definition is **C. To shift something from one place to another.** Here’s why: * The sentence implies the director is transferring or assigning a burden of work (responsibilities) to the assistant, effectively moving them from the director’s role to the assistant’s.

Negative Sampling

The tasks of the study is rather tailored in the LLMs' favor. To generate the instances for the first task (i.e. definition suitability judgment), I took example sentences from each synset associated with a target word and paired them with with definitions that matched the target word. For the second task (i.e. definition generation), only grammatical sentences were presented to the LLM. For the third task (i.e. keyword identification), only definitions that were already associated with the target word were used to generate the instances. For the last task (i.e. target word replacement), That means only reasonable definitions for the word were even considered. For the purposes of testing whether LLMs could distinguish and judge ambiguity study, this sufficed because limited to.

On the other hand, future studies more robust testing using negative sampling should be used. The choices made in this propose task should be motivated by the current answers to the third task.

New Word Meanings

A large amount of research has been done on children and how new word meanings are developed. (Srinivasan et al., 2019). It's been shown that children rely onSimilar tasks would be interesting to carry out on LLMs

Prompt ”Flow”

Similar to the embedding experiment presented here, suggesting information being captured into the contextualize embeddings after the attention heads, it begs the question, what information is encoding to a question as a whole. I propose that since CausalLMs take into considering previous predictions for proceeding predictions, the last contextual embeddings of a sentence should capture an compounding of information. Thus, should there be a reasoning ”trajectory” mapped by word embeddings that can be analyzed.

Human v. Machine v. Ethics

This theme of comparing human and machine intelligence looms over this thesis and many of the current ideas in this field. It's hard to ignore our human influence on these models. It should come as no surprise that machine intelligence today resembles a lot of human considering the data it is trained on. This uncanniness can sprout questions of innateness in language. Thus, is it sufficient to draw conclusions for the cognitive processes of humans from machines, or vice versa? Something to keep in mind, are how the goals for what we want language models to do has shifted countless times (Portelance and Jasbi, 2024).

Considering the current state of with LLMs, as demonstrated by the attempts to train and to retrieve more information out of them, this distinction should not be glossed over (Warstadt et al., 2023; Webson and Pavlick, 2022). There are many who believe that at a certain point, machines may be able to substitute humans in cognitive science and psychology studies.

Relating to the question suggested earlier, there are interesting applications to the theory of language. These findings should motivate how we classify knowledge. Our approaches

REFERENCES

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang

Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. ArXiv:2501.12948 [cs].

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and Fate: Limits of Transformers on Compositionality. ArXiv:2305.18654 [cs].

Jumbly Grindrod. 2024. Transformers, Contextualism, and Polysemy. ArXiv:2404.09577 [cs].

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. ArXiv:1412.6980.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to WordNet: An On-line Lexical Database.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. GSM-Symbolic: understanding the limitations of mathematical reasoning in large language models. ArXiv:2410.05229.

Eva Portelance and Masoud Jasbi. 2024. The roles of neural networks in language acquisition. *Language and Linguistics Compass*, 18(6):e70001. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/lnc3.70001.

Mahesh Srinivasan, Catherine Berner, and Hugh Rabagliati. 2019. Children use polysemy to structure new word meanings. *Journal of Experimental Psychology: General*, 148(5):926–942. Place: US Publisher: American Psychological Association.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. GPT-4 Doesn't Know It's Wrong: An Analysis of Iterative Prompting for Reasoning Problems. ArXiv:2310.12397 [cs].

Max van Duijn, Bram van Dijk, Tom Kouwenhoven, Werner de Valk, Marco Spruit, and Peter van der Putten. 2023. Theory of mind in large language models: examining performance of 11 state-of-the-art models vs. children aged 7-10 on advanced tests. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 389–402, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. ArXiv:1706.03762.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6, Singapore. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do Prompt-Based Models Really Understand the Meaning of Their Prompts? In *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. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought prompting elicits reasoning in large language models. ArXiv:2201.11903.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. ArXiv:2306.05685 [cs].

PROMPT TEMPLATE

```
### User:
Question: How well does "{definition}" describe "{word}" in this
sentence?
Sentence: {example}

<|The Start of Assistant's Answer|>
### Assistant:
{response}
<|The End of Assistant's Answer|>

### User:
Question: Can you provide justification for your answer?

<|The Start of Assistant's Answer|>
### Assistant:
{response}
<|The End of Assistant's Answer|>
```

Figure 5: Task 1 Prompts

```
### User:
Instruction: Give a definition that matches {word} in "{example}"?

<|The Start of Assistant's Answer|>
### Assistant:
{response}
<|The End of Assistant's Answer|>
```

Figure 6: Task 2 Prompt

```
### User:
Question:  What words in "{definition}" are the keywords in defining
"{word}"

<|The Start of Assistant's Answer|>
### Assistant:
{response}
<|The End of Assistant's Answer|>

### User:
Question:  How do each of these keywords contribute to the definition?
<|The Start of Assistant's Answer|>
### Assistant:
response
<|The End of Assistant's Answer|>
```

Figure 7: Task 3 Prompts

```
### User:
Instruction:  Replace "{word}" with a word that matches the meaning the
closest in the sentence:  {example}

<|The Start of Assistant's Answer|>
### Assistant:
{response}
<|The End of Assistant's Answer|>
```

Figure 8: Task 4 Prompt

```
### User:
System
Please act as an impartial judge and evaluate the quality of the response
provided by an AI assistant to the user question displayed below.  Your
evaluation should consider factors such as the helpfulness, relevance,
accuracy, depth, creativity, and level of detail of the response.  Please
rate each evaluation on a scale of 1 to 10.  On this scale, 1 is for a
response that completely does not match and 10 is for a response that
is perfect.  The beginning of your evaluation must be your rating by
strictly following this format:  "[[rating]]", for example:  "Rating:
[[5]]".  After the rating, provide a short explanation.  Be as objective
and concise as possible, using as little sentences as possible.  Please
follow the format exactly and make sure your response include your rating
above all else.
For example, your answer should look like this:
Rating:  [[6]]
Explanation:  The response provides a list of possible substitutions for
m̈inisterb̈ut lacks a clear explanation for why each option is incorrect.
The best answer is missing (answer C̈)̈.  While the final answer is somewhat
relevant to the context, it may not necessarily provide the most accurate
or helpful substitution for m̈inister;̈ and the reasoning behind the other
options is lacking.

User:  {prompt}

Assistant:  {response}
```

Figure 9: Judges Prompt Template