Lost in Disambiguation: How Instruction-Tuned LLMs Master Lexical Ambiguity

Luca Capone[1,*], Serena Auriemma[1], Martina Miliani[1], Alessandro Bondielli[1,2] and Alessandro Lenci[1]

[1] *CoLing Lab, Dipartimento di Filologia, Letteratura e Linguistica, Università di Pisa, Via Santa Maria, Pisa, 56126, Italy*

[2] *Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo, 3 Pisa, 56127, Italy*

Abstract

This paper investigates how decoder-only instruction-tuned LLMs handle lexical ambiguity. Two distinct methodologies are employed: Eliciting rating scores from the model via prompting and analysing the cosine similarity between pairs of polysemous words in context. Ratings and embeddings are obtained by providing pairs of sentences from Haber and Poesio [1] to the model. These ratings and cosine similarity scores are compared with each other and with the human similarity judgments in the dataset. Surprisingly, the model scores show only a moderate correlation with the subjects' similarity judgments and no correlation with the target word embedding similarities. A vector space anisotropy inspection has also been performed, as a potential source of the experimental results. The analysis reveals that the embedding spaces of two out of the three analyzed models exhibit poor anisotropy, while the third model shows relatively moderate anisotropy compared to previous findings for models with similar architecture [2]. These findings offer new insights into the relationship between generation quality and vector representations in decoder-only LLMs.

Keywords

Lexical ambiguity, Decoder models, Transformer, LLM, Cosine similarity, Human rating, Anisotropy, Model generation, Model ratings, Polysemy

1. Introduction

Lexical ambiguity (LA) is a peculiar characteristics of human language communication. Words often carry multiple meanings, and discerning the intended sense requires nuanced comprehension of contextual cues. LA is a broad concept subsuming several semantic phenomena, such as regular and irregular polysemy, homonymy, and the coinage of new senses. Humans handle such ambiguity effortlessly, leveraging contextual information, prior knowledge, and pragmatic inference. However, for Large Language Models (LLMs), which rely on statistical patterns in text data, accurately resolving lexical ambiguity remains a challenging task.

Despite their remarkable capability of using words appropriately in context, one critical aspect that requires deeper investigation is whether such models possess human-like lexical competence, enabling them to generalize from multiple instances of the same phenomenon, or if they are simply mimicking these instances.

In this paper, we aim to investigate how LLMs handle LA. Specifically, we challenged three decoder-only instruction-tuned models to generate lexical similarity ratings for word pairs used in two different contexts, with various degrees of sense similarity. To achieve this, we employed a chain-of-thought approach, prompting the models to produce a step-by-step reasoning process before assigning their ratings, allowing them to better distinguish between different senses of the same term.

For this task, we used the dataset released by Haber and Poesio [1], which includes human similarity judgments. The models' generated ratings were correlated with human similarity judgments to determine whether their lexical disambiguation competence aligns with that of humans. Additionally, we computed the cosine similarity between the models' internal representation of the ambiguous target words. Our research question is twofold: i.) to **assess if the models' generated ratings are consistent with their internal representations of the target words**; ii.) to **determine whether the internal representations have a more similar distribution to human ratings than the generated responses**.

We are aware that context-sensitive word embeddings, like those of LLMs, can suffer from a *representation degeneration problem* (see Section **??** for further details), which limits their semantic representational power. Hence, we included in our analysis a brief overview of how this phenomenon affects the internal representational space of the models under our investigation.

To the best of our knowledge, this is the first study in

CLiC-it 2024: Tenth Italian Conference on Computational Linguistics, Dec 04 — 06, 2024, Pisa, Italy

[†] For the specific purposes of Italian Academy, Luca Capone is responsible for Sections 2, 3.4 and 3.5, Serena Auriemma for sections 1 and 4.1, Martina Miliani for sections 3.1, 3.3 and 3.6, Alessandro Bondielli for sections 3.2 and 4 and Alessandro Lenci for section 5

luca.capone@fileli.unipi.it (L. Capone)
0000-0002-1872-6956 (L. Capone); 0009-0006-6846-5826 (S. Auriemma); 0000-0003-1124-9955 (M. Miliani); 0000-0003-3426-6643 (A. Bondielli); 0000-0001-5790-4308 (A. Lenci)

which different decoder-only models were tested on their metalinguistic competence regarding LA. Understanding how LLMs manage this type of complex semantic phenomenon, based on the interplay of multiple contextual factors, can guide new improvements in training methodologies for the development of more sophisticated and robust models that better mimic human-like language understanding.

2. Related works

One of the main reasons for the success of Transformer-based LMs is their ability to represent context-dependent meaning. The specific meaning a token assumes in a given context is encoded within the internal layers of these models and is reflected in the spatial distribution of the produced embeddings, where unique context vectors for each token occurrence are placed distinctly [2].

Yenicelik et al. [3], extending Ethayarajh [2]'s study, sought to obtain a general overview of BERT's [4] embedding space concerning polysemous words. They confirmed that BERT does indeed form contextual clusters, which nevertheless obey semantic regularities in a broad sense. These clusters may fulfill denotative, connotative, or syntactic criteria, with converging groups consistent with the idea of polysemy as a gradual continuum. However, the embedding space of such models shows regularities influenced not only by linguistic factors but also by one of the model's training objectives, i.e., Next sentence Prediction [5]. This confirms the flexibility and richness of contextual representations but raises questions about their representativeness of proper linguistic features. Several studies compared the contextual vectors of encoder models like BERT and ELMO with human similarity judgments, demonstrating that human judgments usually correlate with the cosine similarity of polysemous word pairs [1, 6], and even more with homonyms pairs [7].

Recently, the correlation between human similarity judgments and model competence regarding LA was also explored for larger decoder-models, such as GPT-4 [8]. However, this analysis only considers GPT's generated ratings, without examining the internal representations of polysemous words. Hu and Levy [9] pointed out that prompting might not be the most reliable way to evaluate models, as the generated responses are not always consistent with the model's probability distribution.Their work primarily addresses two tasks: token prediction and sentence pair selection. In their evaluations, token prediction is determined by identifying the token with the highest probability from the entire vocabulary, while sentence pair selection is based on the perplexity of two competing propositions. While their methodology yields strong results, it is not directly applicable to our study due to the non-deterministic nature of model outputs in response to the task we propose. Specifically, presenting the model with two alternative sentences is not feasible in our experiment, as the objective is to have the model generate a chain-of-thought output that differentiates between the distinct senses of an ambiguous term and subsequently produces a rating. One alternative would be to have the model directly predict the rating and check which vocabulary token (among the numbers in the rating scale) has the highest probability. However, this approach would not generate the contextual embeddings for the target term necessary for our comparisons. Furthermore, as discussed in section 3.3, ratings produced without the chain-of-thought approach were inconsistent.

Since we are dealing with word similarities, the most straightforward way to measure a model's internal knowledge about polysemic words is by using cosine-similarities. However, given the contextual nature of these models, embeddings might not transparently reflect semantic properties, as they can be influenced by other superficial contextual factors. This makes it challenging to discern whether a high value of cosine similarity is due to word sense similarity or to a general closeness of the word embeddings in the space, the so-called *anisotropy*.

Anisotropy can indeed negatively affect the representational power of embeddings, and several methods have been proposed to mitigate its effect [10, 11, 12]. Nevertheless, it has been demonstrated that anisotropy does not have a negative impact on model performance [12].

Given these complexities, we decided to further investigate LA with large decoder-only models to highlight differences with results obtained from smaller encoders and to determine whether their behaviour aligns with the human competence on LA. We compared the performance of different instruction-tuned decoders to obtain a more comprehensive overview of how these models handle this phenomenon. To ensure a thorough evaluation, we consider both the models' generated ratings for polysemous words and their cosine similarities. Additionally, in our analysis, we took into account the level of anisotropy exhibited by these models.

3. Experimental settings

3.1. Dataset

We use the dataset introduced in Haber and Poesio [1], which includes a set of target words in various contexts. Human judgments were collected on sentence pairs with the same word, by asking participants to rate the similarity of the target word meaning in the different contexts. We chose to focus only on in-vocabulary tokens, as we aimed to compare models' performances on their generated embeddings, without employing additional opera-

Table 1
Sentence pairs for each similarity class based on the distribution of human ratings. Classes "Homonym" and "Same sense & context" in boldface were manually identified [1].

Similarity class	Count
Homonym	11
Different	45
Quite different	49
Quite similar	37
Similar	19
Equal	68
Same sense & context	7
Total	236

tions (e.g., mean pooling of subword embeddings). Thus, we retain about 79% of the dataset sentence pairs (i.e., 236 out of the original 297).

We further categorized sentence pairs according to the distribution of the human ratings, dividing them into four *similarity classes* depending on their interquartile ranges.[1] We also included the two manually identified groups from Haber and Poesio [1]. One consists of sentence pairs with homonyms, and the other consists of words having the same sense in highly similar contexts. As these groups did not have human ratings, we assigned ten ratings to each data point, randomly selected around 0.01 for homonyms (indicating completely different meanings) and around 1.00 for the other group. The human ratings serve as the ground truth for the post-hoc analysis in Section 4. The final dataset counts 35 target word types (see Figure 1 for their list and token distribution), with a set of similarity judgments for each pair.

3.2. Models

To assess the capability of LLMs to capture varying degrees of LA, we selected three decoder-only open models of comparable size. We chose instruction-tuned models exclusively, as this configuration is more suitable for conditional text generation: `Meta-Llama-3-8B-Instruct` [13], hereafter referred to as LLaMA; `Gemma-1.1-7B`[2], hereafter referred to as Gemma; and `Mistral-7B-Instruct-v0.2`[3], hereafter referred to as Mistral. All models are instruction-tuned autoregressive LLMs with around 7 Billion parameters. We chose these models as they are representative of popular and widely used open-weights LLMs. We used the Huggingface implementation of the models for our experiments.

3.3. Prompting

We report experimental results using a single prompt.[4] The prompt was designed to closely follow the methodology used by Haber and Poesio [1] for modeling the LA task to collect crowdsourced data, ensuring a fair comparison between LLMs' ratings and human judgments. In our setup, we provided the models with two sentences, each containing the same target word. We then prompted the models to return a rating score indicating how similar the word's usage was in the two occurrences. The rating score ranged from 1 to 100, where 1 indicated that the word was used with completely different senses in the two sentences, and 100 indicated that the word was used with the same sense across sentences. We formulated the instructions following common rules of thumb for prompting LLMs [14].

In preliminary experiments, we asked the model to return the similarity rating first and then to return the motivation of such rating. We observed that i.) the rating was quite inconsistent with the underlying motivations given by the models, ii.) the motivations were usually more appropriate than the ratings, and that iii.) the models tended to return the same rating for all the sentence pairs. Thus, we chose to ask the model to provide the motivation first, followed by the rating. This allowed the models to provide more accurate ratings. Such a behavior is in line with the literature on "chain-of-thought" prompting [15]. Additionally, we chose *beam search* as a generation strategy, with 2 beams. The models sampled the next generated token among the 50 most probable words. We combined this strategy with *nucleus sampling*, by setting a probability threshold of 0.95.

3.4. Embedding Extraction and Cosine-similarity

Building on the experiments in Haber and Poesio [1] and Loureiro and Jorge [16], we used the embeddings generated from the last layer and the average of the embeddings from the last four layers as contextual embeddings for the generated tokens. The idea behind this approach is that the last layer embeddings represent the most contextual and generation-focused features, while the preceding layers capture more general aspects of the processed sequence. This method allowed us to obtain two sets of contextual embeddings for each generation. Due to the unidirectional design of the decoder architectures, the repetition of the input sentences across generations was necessary. The model had to process all tokens in both sentences before providing sufficient contextual embeddings, making the input vectors unsuitable for the task. Once the vectors for each generated token were obtained, we isolated the embeddings corresponding to

[1] See Appendix 4 for the interquartile ranges values and a visual representation.
[2] https://huggingface.co/google/gemma-1.1-7b-it
[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[4] The full prompt is available in Appendix A.

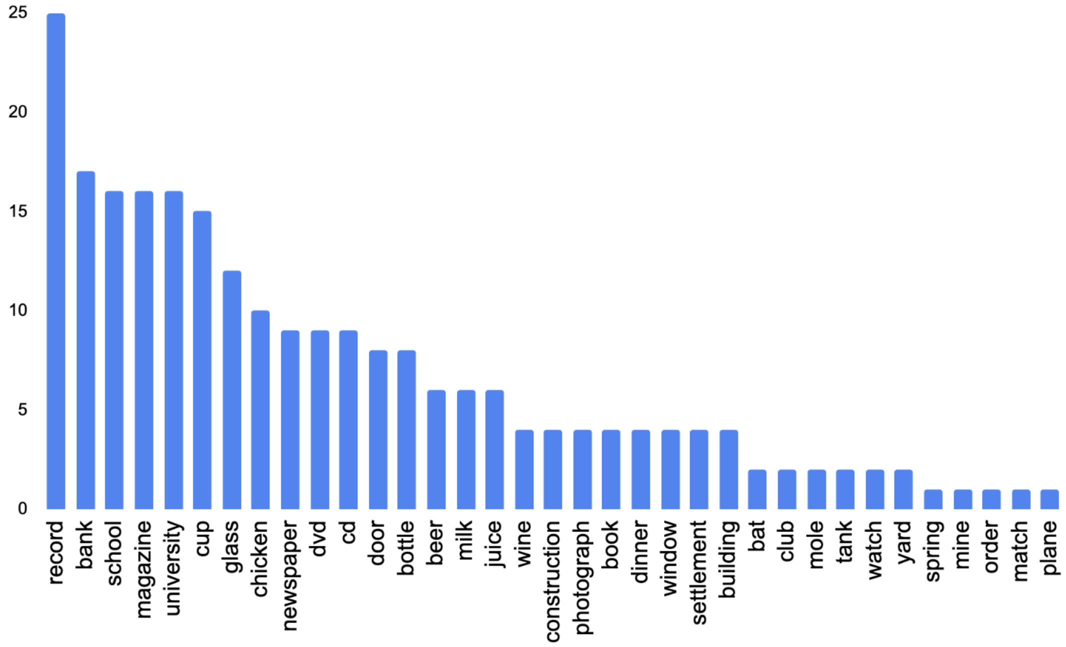


Figure 1: The distribution of the target words in our dataset.

the tokens of the target words contained in the stimulus sentences (repeated by the model at the beginning of the generation). Afterwards, cosine similarity values were calculated between the target word vectors extracted from the last layer and the last four layers.

3.5. Investigating anisotropy in decoder-only models

The so-called *representation degeneration problem* [17] is a well-known phenomenon observed in several Transformer architectures, even in those trained on data other than text [18]. This issue causes most of the model's learned word embeddings to drift to a narrow region of the vector space [2], making them very close to each other in terms of cosine similarity, and consequently limiting their semantic representational power. Since our work primarily focuses on analyzing LLMs' ability to capture subtle semantic properties such as polysemic relations and relies in part on the computation of cosine similarity between token pair embeddings, we decided to further investigate this phenomenon.

We conducted an analysis of the distribution of the models' generated tokens in the vector space to understand the extent of representation degeneration and its implications for the semantic representation of our target tokens. For each model, we sampled 1,000 pairs of random tokens from all generations of the model across the entire dataset. We extracted the representations of these tokens from both the last layer and the average of the last four layers. We then computed the average cosine similarity of the sampled embedding pairs for the last and last four layers separately.

3.6. Evaluation

We compared the Model Rating Scores (MRSs), the Cosine Similarity Scores (CSSs), and the Human Rating Scores (HRSs) collected by Haber and Poesio [1] by means of Spearman Correlation. The correlation between MRSs and CSSs should shed light on the internal coherence of each model and aims at answering the following question: **Is the metalinguistic knowledge of the model consistent with its internal representations?** By comparing HRSs with MRSs and HRSs with CSSs, we aim to explore a different issue: **Do the human ratings have a more similar distribution to what a model generates rather than its internal representation or vice-versa?** Before computing the correlation, we rescaled the CSSs in the range $0.01 - 1.00$. We also rescaled the MRSs from the range $1 - 100$, to the range $0.01 - 1.00$. As for the HRSs, we used the average of the

Table 2
Spearman correlation measures between Model Rating Scores (MRS), Human Rating Scores (HRS), and Cosine Similarity Score (CSS). The results with CSS are computed both with the last hidden state vectors (*Last*) and with vectors averaged from the last four hidden states (*Last4*). The model's result with the correlation score farther from zero for each comparison is in boldface. P-values < 0.05 are marked with *.

Model	MRS vs. HRS	CSS vs. HRS		MRS vs. CSS	
		Last4	*Last*	*Last4*	*Last*
Mistral	$0.404*$	$\mathbf{-0.020}$	$\mathbf{-0.020}$	0.047	0.042
Gemma	$0.446*$	-0.002	0.001	0.066	0.056
LLaMa	$\mathbf{0.616}^*$	$\mathbf{0.016}$	0.110	-0.002	$\mathbf{0.118}$

collected ratings for each sentence pair in the correlation.

4. Results and analyses

Table 2 reports the correlations among human ratings, model ratings, and cosine similarities. First, we consider the correlation between cosine similarities and human ratings. The three models exhibit a near-zero correlation between CSS and HRS, which is always negative for Mistral (-0.020) and positive for LLaMa (0.016, 0.110). Second, we compare model ratings to human ones. We observe that there is a moderate-to-high correlation for LLaMa (0.616), and a low-to-moderate correlation for Mistral (0.404) and Gemma (0.446). Thus, despite being more correlated than cosine similarities, the models' ratings often differ from human ones. We observed some recurrent patterns in the score assignments by each model[5]. LLaMA frequently assigns similarity ratings of 20, 60, and 80. Gemma shows a preference for very low or very high scores, leaving the middle range sparsely populated. Mistral appears the most balanced in its evaluations, yet it still favors round values (100, 90, 80, etc.) and shows a strong preference for values close to 1. However, these rating preferences do not seem to correspond to lexical preferences. Although MRS appears to correlate better with HRS than CSS, the unstable nature of prompt results and their sensitivity to biases from the data or prior training make them less suitable for inspecting the model's competence regarding complex semantic features like polysemy.

In addition to this, we observe that in the comparison between CSS and HRS, the cosine similarity distributions of Mistral and LLaMA appear similar, while Gemma's distribution is shifted towards higher values. We can surmise that this may be attributed to a greater anisotropy in the embedding space characterizing the Gemma model (see Section 4.1 for a thorough analysis). Overall, the CSS reflects the similarity distribution indicated by the human subjects far less accurately than the MRS.

Table 3
Average cosine similarities between 1000 random pairs of tokens for each model.

Model	**Avg Cosine Similarity**	
	Last4	*Last*
Mistral	0.138	0.137
Gemma	0.672	0.746
LLaMA	0.24	0.228

Finally, to evaluate the internal coherence of the models in terms of the agreement between the generated similarity scores and hidden representations, we also compared the cosine similarities and model ratings of each model. In this case, the highest correlation is obtained by LLaMa, which nonetheless exhibits a very weak correlation (0.118 on the last layer), meaning that one can not reliably predict MSR based on the CSS. We speculate that a complex phenomenon like polysemy is only sub-optimally represented at the token embedding level.

4.1. Anisotropy

As shown in Table 3, the degree of anisotropy varies quite significantly among the three decoder-only models, especially between Gemma and the other two models, Mistral and LLaMA. Gemma exhibited the highest cosine similarity scores, approximately 0.67 for the last four layers and slightly higher for the last layer (0.75), corroborating the findings of [2] regarding anisotropy in decoder models such as GPT-2, which peaks in the last layer. Conversely, Mistral showed the lowest scores (0.137 for both the last and last four layers), followed by LLaMA (0.24 for the last four layers and 0.228 for the last layer), indicating a much more isotropic space than one would expect for models with similar architecture and comparable size. This suggests that anisotropy might not be the same in all Transformer-based models. Rather, it appears to be a property that is present at varying degrees in models, with some exhibiting greater anisotropy than others. This may be due to specific differences in how models were trained, both in terms of data used, and pre-training, fine-tuning, and post-training techniques. We aim to further investigate this aspect in the future.

Due to these differences, we decided not to apply any post-processing method [12, 10] to mitigate the anisotropy of our target vectors. However, looking in detail at the relationship between the models' anisotropy and their respective cosine similarities, it seems that the relatively low degree of anisotropy in both Mistral and LLaMa does not result in a better correlation between their CSS and HRS. On the contrary, despite a generally

[5] Figure 3 in the appendix enables a detailed examination of the ratings generated by the models. An interactive version of these plots will be available on GitHub.

moderate level of anisotropy found in these decoder-only models, the CSS of the target tokens correlate less with the HRS than the MRS. This finding suggests that the low correlations of cosine similarities can not be (entirely) due to the embedding anisotropy and that conversely the latter does not affect the model generation abilities significantly. This appears to confirm recent trends suggesting that cosine similarity is a suboptimal measure to explore Transformers' geometries [19].

5. Conclusion and future work

Our study investigates how LLMs handle LA, using two distinct methodologies: Eliciting rating scores from the model and analyzing the cosine similarity between pairs of polysemous words. We calculated the Spearman correlation between HRS vs. MRS, HRS vs. CSS, and MRS vs. CSS. The aim was to determine whether the model's metalinguistic knowledge aligns with its internal representations and to assess if human ratings more closely match the outputs generated by the model than its internal representations.

The lack of correlation between CSS and MRS provides intriguing insights into the relationship between the internal representations of LLMs and the responses they generate in metalinguistics tasks, like explicitly assigning similarity ratings. Specifically, the argument presented by Hu and Levy [9] appears to be validated: Generated responses do not always reflect the model's internal processing. Hu and Levy [9] compared model generations with their probability distributions and found the latter method to be more accurate. In contrast, in our study, using the internal representations of the model (i.e., the contextual embeddings, as motivated in Section 2) proved to be a less reliable method. The most straightforward conclusion is that generative LLMs might be suboptimal for estimating word sense similarity. The superior performance of probability estimation reported by Hu and Levy [9] might be due to its direct link to the prediction training objectives of LLMs. To further investigate the relationship between CSS and MRS, we inspected the anisotropy of the embeddings. The average cosine similarity among a sample of generated tokens was relatively low, indicating that anisotropy did not affect our cosine similarity measures and is not characteristic of all decoder-only models under investigation. The lack of anisotropy observed in some of the analyzed decoder-only models is at odds with the conclusions of Ethayarajh [2], who reported a higher anisotropic space for GPT-2.

Only MRS yielded a moderate correlation with HRS, indicating that LA is not fully captured by the analyzed models, in text generation and vector representations. In conclusion, the relationship between human judgments, model generations, and internal representations appears unclear and calls for further research. Despite the low anisotropy of the examined models, cosine similarity did not reveal a correlation between the generations and the internal representations of the models, indicating a need for deeper investigation. We plan to repeat the experiments by leveraging recent results with sparse autoencoders [20] to decompose the meanings of lexically ambiguous words. This could provide a deeper understanding of the models' ability to handle and represent polysemy.

We could not extract embeddings from commercial models, such as those provided by OpenAI, which are accessible only through APIs. However, it would be valuable in future research, if and when this functionality becomes available, to analyze and compare the internal representations and the generated outputs of these state-of-the-art models.

Another promising avenue for future research is to examine the differences between vector representations and generated tokens with respect to linguistic phenomena beyond polysemy and lexical ambiguity. For instance, incorporating out-of-vocabulary words could allow for an exploration of semantic shifts caused by the addition of prefixes or suffixes (e.g., “order” vs. “dis-order”), offering valuable insights. This analysis would benefit from using a tokenization strategy that treats morphemes as subtokens, alongside an investigation into the degree of anisotropy in these models.

Acknowledgments

We acknowledge financial support under the PRIN 2022 Project Title "Computational and linguistic benchmarks for the study of verb argument structure" – CUP I53D23004050006 - Grant Assignment Decree No. 1016 adopted on 07/07/2023 by the Italian Ministry of University and Research (MUR). This work was also supported under the PNRR—M4C2—Investimento 1.3, Partenariato Esteso PE00000013—“FAIR—Future Artificial Intelligence Research”—Spoke 1 “Human-centered AI,” funded by the European Commission under the NextGeneration EU programme, and partially supported by the Italian Ministry of University and Research (MUR) in the framework of the PON 2014-2021 “Research and Innovation" resources – Innovation Action - DM MUR 1062/2021 - Title of the Research: “Modelli semantici multimodali per l'industria 4.0 e le digital humanities.”

References

[1] J. Haber, M. Poesio, Patterns of polysemy and homonymy in contextualised language models, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 2663–2676.

[2] K. Ethayarajh, How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings, arXiv preprint arXiv:1909.00512 (2019).

[3] D. Yenicelik, F. Schmidt, Y. Kilcher, How does bert capture semantics? a closer look at polysemous words, in: Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, 2020, pp. 156–162.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[5] T. Mickus, D. Paperno, M. Constant, K. van Deemter, What do you mean, bert?, in: Proceedings of the Society for Computation in Linguistics 2020, 2020, pp. 279–290.

[6] S. Trott, B. Bergen, Raw-c: Relatedness of ambiguous words–in context (a new lexical resource for english), arXiv preprint arXiv:2105.13266 (2021).

[7] S. Nair, M. Srinivasan, S. Meylan, Contextualized word embeddings encode aspects of human-like word sense knowledge, arXiv preprint arXiv:2010.13057 (2020).

[8] S. Trott, Can large language models help augment english psycholinguistic datasets?, Behavior Research Methods (2024) 1–19.

[9] J. Hu, R. Levy, Prompting is not a substitute for probability measurements in large language models, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 5040–5060.

[10] J. Mu, S. Bhat, P. Viswanath, All-but-the-top: Simple and effective postprocessing for word representations, arXiv preprint arXiv:1702.01417 (2017).

[11] V. Zhelezniak, A. Savkov, A. Shen, N. Y. Hammerla, Correlation coefficients and semantic textual similarity, arXiv preprint arXiv:1905.07790 (2019).

[12] W. Timkey, M. Van Schijndel, All bark and no bite: Rogue dimensions in transformer language models obscure representational quality, arXiv preprint arXiv:2109.04404 (2021).

[13] AI@Meta, Llama 3 model card (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[14] J. Phoenix, M. Taylor, Prompt Engineering for Generative AI, O'Reilly Media, Inc., 2024.

[15] 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 (2022) 24824–24837.

[16] D. Loureiro, A. Jorge, Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation, arXiv preprint arXiv:1906.10007 (2019).

[17] J. Gao, D. He, X. Tan, T. Qin, L. Wang, T.-Y. Liu, Representation degeneration problem in training natural language generation models, 2019. arXiv:1907.12009.

[18] N. Godey, É. de la Clergerie, B. Sagot, Anisotropy is inherent to self-attention in transformers, arXiv preprint arXiv:2401.12143 (2024).

[19] H. Steck, C. Ekanadham, N. Kallus, Is cosine-similarity of embeddings really about similarity?, in: Companion Proceedings of the ACM on Web Conference 2024, 2024, pp. 887–890.

[20] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, et al., Towards monosemanticity: Decomposing language models with dictionary learning, Transformer Circuits Thread 2 (2023).

A. The prompt

The following text box shows the prompt used to test LLMs in our lexical ambiguity experiment. The underlined text was replaced by sentences and word targets from the dataset shared by Haber and Poesio [1].

> You will receive two sentences. Your task is to rate how similar is the use of the word 'word' in the two sentences.
>
> - Sentence 1: s1
> - Sentence 2: s2
>
> You must follow the following principles:
>
> - Assign a rating on a scale of 1-100, where 1 means that the word is used with completely different senses in the two sentences and 100 means that the word is used in the same sense across the two sentences.
> - Return your answer in this way:
>   - Rewrite the two sentences following this template:
>   - Sentence1: <text>
>   - Sentence2: <text>
>   - Motivation: <a concise motivation for your rating>
>   - Rating score: <only a float number on a scale of 1-100 and nothing else>.
> - Interrupt generation after the rating score.
>
> Question: how similar is the use of the word word in the following two sentences?
> s1
> s2
> Answer:

B. More on human-rated pairs

Table 4 shows the interquartile ranges of the human ratings collected by Haber and Poesio [1] and related only to the sentence pairs filtered as described in Section 3.1. The ranges are plotted in Figure 2.

In Table 5, the Spearman correlation measures between Model Rating Scores (MRS), Human Rating Scores (HRS), and Cosine Similarity Score (CSS). Sentence pairs from the similarity class 'Homonym' and 'Same sense & context', for which Haber and Poesio [1] did not provide crowdsourced data, were not included in the computation.

Table 4
The interquartile ranges of the human ratings related to the sentence pairs selected for our experiments.

Quartile	Range
First	$0 - 0.556$
Second	$0.556 - 0.845$
Third	$0.845 - 0.934$
Fourth	$0.934 - 1.00$

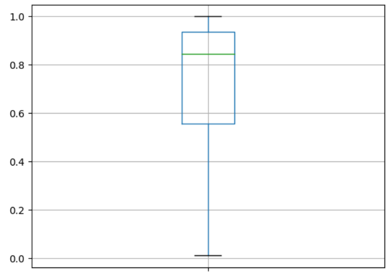
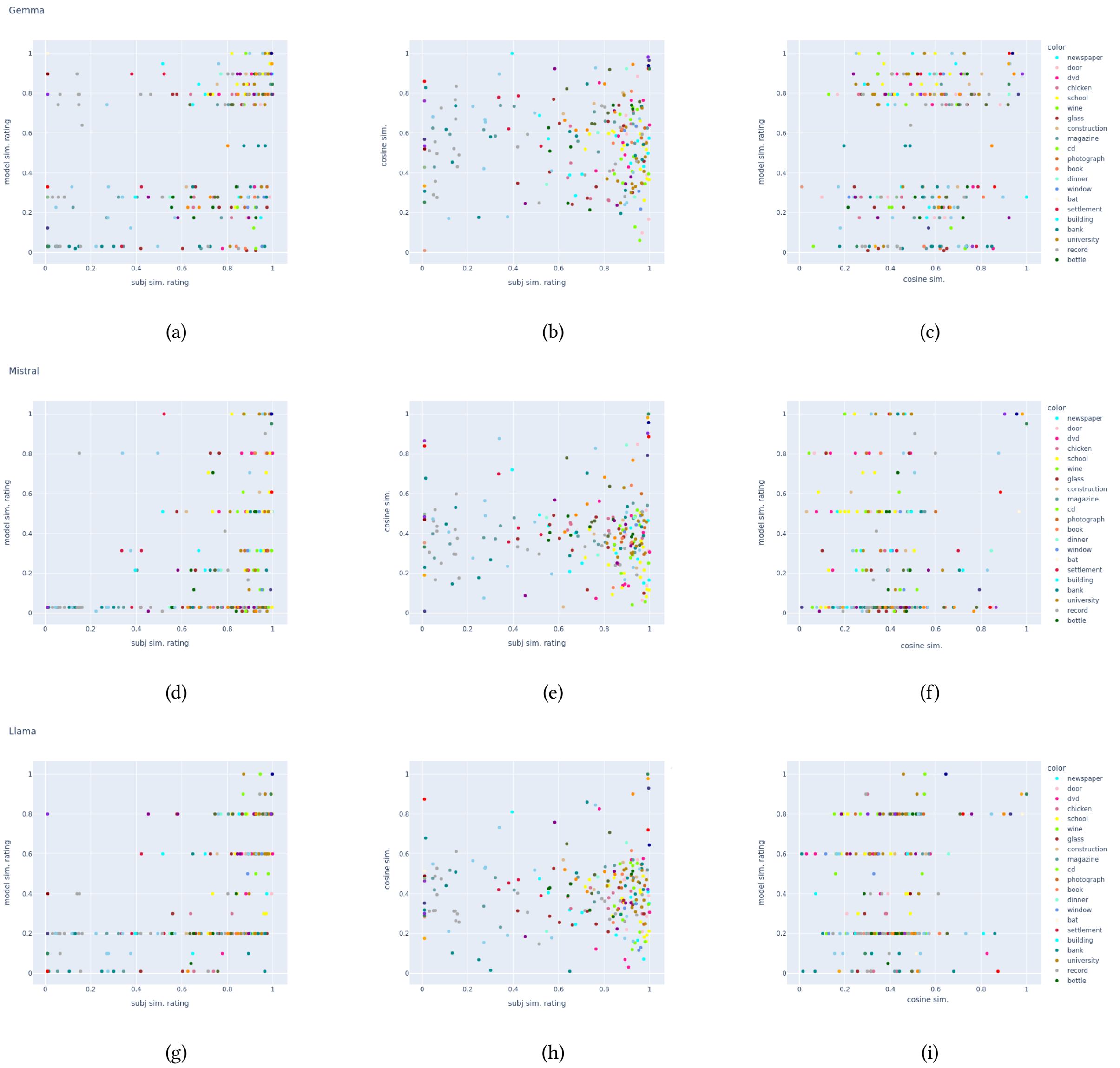


Figure 2: The distribution of the human ratings given to sentence pairs filtered as described in Section 3.1.

Table 5
Spearman correlation measures between MRS, HRS, and CSS. The CSS are computed both with last hidden state vectors (*Last*) and the average of the last four (*Last4*). In bold is the model's result with the correlation score further from zero for each comparison. 'Homonym' and 'Same sense & context' pairs were not included in the computation. P-values < 0.05 are marked with *.

Model	MRS vs HRS	CSS vs HRS		MRS vs CSS	
		Last4	*Last*	*Last4*	*Last*
Mistral	0.333*	−0.010	−0.100	0.018	0.026
Gemma	0.420*	**−0.130**	**0.126**	**0.18**	0.028
LLaMa	**0.583***	−0.067	0.098	0.052	**0.053**

C. Additional Figures



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3: In this image, the scatterplots of the results are reported for the three models. In the first row, the results related to Gemma (a, b, c); in the second row, Mistral's results (d, e, f); in the third row LLaMa's results (g, h, i). In the first column (a, d, g), we plotted the comparison between HRSs (on the x-axis) and MRSs (on the y-axis); in the second column (b, e, h), the comparison between CSSs (on the x-axis) and HRSs (on the y-axis); in the third column c, f, i), we compared CSSs (on the x-axis) and MRSs (on the y-axis). In the plots, each color refers to a different target word.