Are Decoder-Only Language Models Better than Encoder-Only Language Models in Understanding Word Meaning?

Muhammad Reza Qorib, Geonsik Moon, and **Hwee Tou Ng**
Department of Computer Science, National University of Singapore
mrqorib@u.nus.edu, gsmoon97@u.nus.edu, nght@comp.nus.edu.sg

Abstract

The natural language processing field has been evolving around language models for the past few years, from the usage of n-gram language models for re-ranking, to transfer learning with encoder-only (BERT-like) language models, and finally to large language models (LLMs) as general solvers. LLMs are dominated by the decoder-only type, and they are popular for their efficacy in numerous tasks. LLMs are regarded as having strong comprehension abilities and strong capabilities to solve new unseen tasks. As such, people may quickly assume that decoder-only LLMs always perform better than the encoder-only ones, especially for understanding word meaning. In this paper, we demonstrate that decoder-only LLMs perform worse on word meaning comprehension than an encoder-only language model that has vastly fewer parameters.

1 Introduction

Large language models (LLMs) are highly effective tools for solving different kinds of problems in natural language processing (Qorib and Ng, 2023; Zhou et al., 2023), computer vision (Liu et al., 2023a), robotics (Zeng et al., 2023), and more. Due to their fascinating abilities to solve a myriad of tasks, large language models, which are dominated by the decoder-only type, are often considered general problem solvers (Mirchandani et al., 2023; Yao et al., 2023). Moreover, large language models are able to perform tasks outside of what they were trained on with few to no examples, a phenomenon referred to as emergent abilities (Wei et al., 2022).

Large language models have been shown to have strong comprehension abilities (Liu et al., 2023b). As such, it is natural to think that they are the best for lexical semantics. However, Zhu et al. (2024) reported that decoder-only language models struggle with understanding more nuanced contextual features. This motivates us to investigate the semantic understanding of decoder-only language models compared to the encoder-only ones.

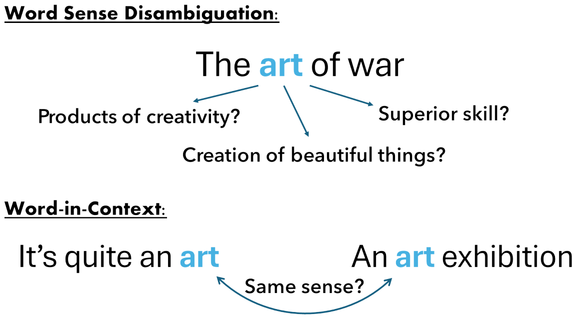


Figure 1: Illustration of the word sense disambiguation and word-in-context tasks.

We evaluate the semantic understanding capability of language models through the word sense disambiguation (WSD) and word-in-context (WiC) tasks. Word sense disambiguation is the task to determine which sense of a word is meant in a particular context, while the word-in-context task is to identify whether the word senses of the same word in two sentences are the same or not (Figure 1). We investigate both open-source decoder-only models, Mistral (Jiang et al., 2023) and Llama 2 (Touvron et al., 2023), and a closed-source model, GPT-4[1] (OpenAI, 2023). We compare the results against a strong encoder-only model, DeBERTa-V3-Large (He et al., 2023).

2 Methods

2.1 Encoder-Only Models

We use encoder-only language models as a binary classifier for the word sense disambiguation and word-in-context tasks. For the WSD task, we follow the ESR (Song et al., 2021) method. The input to the model is the concatenation of the sentence s with the enhanced sense representation ($\mathtt{ESR}$) of

[1] The architecture of GPT-4 is not transparent, but the model is trained with the next token prediction task. In addition, previous GPT versions are decoder-only models.

Findings of the Association for Computational Linguistics: ACL 2024, pages 16339–16347
August 11-16, 2024

one candidate sense $c \in C_w$ for the target word w. The model then calculates the probability $p(c)$ of the candidate sense c to be the correct sense, by projecting the hidden representation of the target word w to the binary classes. Lastly, we choose the sense c with the highest probability. With LM representing an encoder LLM, σ representing the softmax function, and the square bracket representing concatenation, the method can be written as follows:

$$\boldsymbol{p}(c, \neg c) = \sigma(\boldsymbol{W} \times \mathtt{LM}([s; \mathtt{ESR}(c)])_w + \boldsymbol{b}) \quad (1)$$

$$\hat{y} = \arg\max_{c \in C_w} p(c) \quad (2)$$

$\boldsymbol{p}(c, \neg c)$ is a 2-dimensional vector denoting the probabilities that w belongs to sense c or not. The predicted sense $\hat{y}$ is the sense with the highest probability among all senses of w.

For the WiC task, the input to the model is the concatenation of the two sentences s_1 and s_2, prepended with a [CLS] pseudo-token. The answer is computed from the projection of the [CLS] pseudo-token to the binary classes.

$$\boldsymbol{p}(yes, no) = \sigma(\boldsymbol{W} \times \mathtt{LM}([s_1; s_2])_{\mathtt{CLS}} + \boldsymbol{b}) \quad (3)$$

$$\hat{y} = \arg\max_{c \in \{yes, no\}} p(c) \quad (4)$$

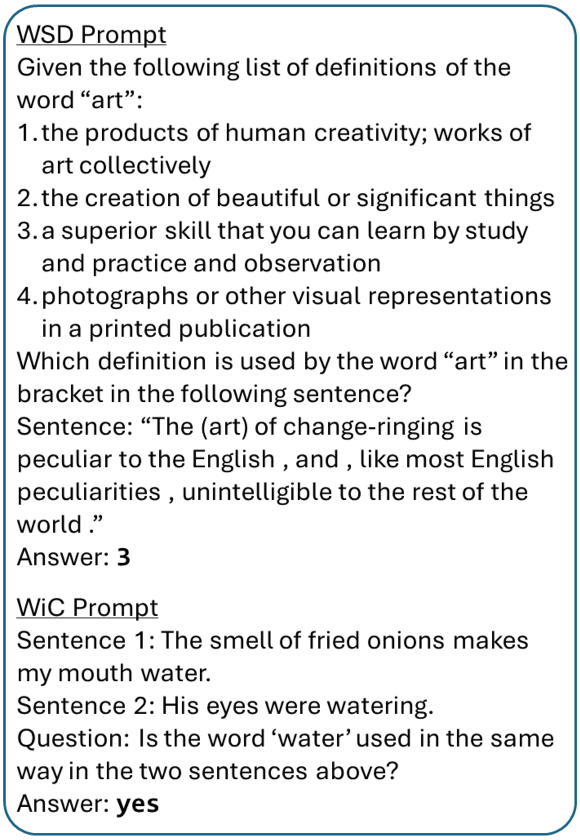
2.2 Decoder-Only Models

We perform WSD and WiC prediction by providing a natural prompt x as input to the decoder-only model and choosing the option $y \in Y$ which is the most probable continuation of the prompt as the selected answer. For WSD, the prompt is in a multiple-choice question style with the option number $Y = \{1, 2, ..., k\}$ as the expected answer. For the WiC task, the prompt is a question about whether the target word in the two sentences has the same sense and the answer options are $Y = \{\mathtt{yes}, \mathtt{no}\}$. The prompt examples are given in Figure 2.

$$\hat{y} = \arg\max_{y \in Y} p(y|x) \quad (5)$$

3 Experiments

We evaluate the models' performance in the zero-shot, few-shot (in-context learning), and fine-tuned settings. All experiments of decoder-only models use natural language prompts. For the few-shot setting, we prepend the prompt with four examples from the training set.



WSD Prompt
Given the following list of definitions of the word "art":
1. the products of human creativity; works of art collectively
2. the creation of beautiful or significant things
3. a superior skill that you can learn by study and practice and observation
4. photographs or other visual representations in a printed publication
Which definition is used by the word "art" in the bracket in the following sentence?
Sentence: "The (art) of change-ringing is peculiar to the English , and , like most English peculiarities , unintelligible to the rest of the world ."
Answer: **3**

WiC Prompt
Sentence 1: The smell of fried onions makes my mouth water.
Sentence 2: His eyes were watering.
Question: Is the word 'water' used in the same way in the two sentences above?
Answer: **yes**

Figure 2: Example prompts for the decoder-only models. The bolded text is the model's expected output.

3.1 Dataset

For WSD, we utilize SemCor (Miller et al., 1994) as the training dataset. SemCor was manually annotated with WordNet senses and predominantly used in the literature for training supervised WSD systems (Zhong and Ng, 2010; Hadiwinoto et al., 2019; Song et al., 2021). The evaluation framework proposed by Raganato et al. (2017) incorporates five evaluation datasets from the Senseval/SemEval series: Senseval-2 (SE2) (Edmonds and Cotton, 2001), Senseval-3 task 1 (SE3) (Snyder and Palmer, 2004), SemEval-07 task 17 (SE07) (Pradhan et al., 2007), SemEval-13 task 12 (SE13) (Navigli et al., 2013), and SemEval-15 task 13 (SE15) (Moro and Navigli, 2015). We evaluate the models by calculating the F1 score on the concatenation of the five test sets (ALL) standardized to the same format and sense inventory of WordNet 3.0 (Table 1).

WiC (Pilehvar and Camacho-Collados, 2019) is a dataset to evaluate a model's capability in differentiating word senses in a binary fashion. The input is two sentences that contain the same target word, and the label is yes if the word sense in the two sentences is the same, and no otherwise. We evaluate the accuracy of the models' prediction on

Dataset	#Doc	#Sent	#Tok	#Ann
SemCor	352	37,176	802,443	226,036
Senseval-2	3	242	5,766	2,282
Senseval-3	3	352	5,541	1,850
SemEval-07	3	135	3,201	455
SemEval-13	13	306	8,391	1,644
SemEval-15	4	138	2,604	1,022
ALL	26	1,173	25,503	7,253

Table 1: Statistics of the WSD datasets (after standardization). #Doc, #Sent, #Tok, and #Ann refer to the number of documents, sentences, tokens, and annotations respectively.

Split	Instances	Nouns	Verbs	#Word
Training	5,428	49%	51%	1,256
Dev	638	62%	38%	599
Test	1,400	59%	41%	1,184

Table 2: Statistics of different splits of the WiC dataset. #Word denotes the number of unique words.

the standard test split (Table 2).

3.2 Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) (Liu et al., 2022) is a set of techniques where only a small subset of added or selected parameters are trained to allow for efficient fine-tuning of LLMs. For all the fine-tuning experiments, we use LoRA (Hu et al., 2022), one of the most widely used PEFT methods. We use the HuggingFace (Wolf et al., 2020) framework to fine-tune the models with LoRA. Additionally, we utilize 4-bit quantization to optimize memory usage and speed up inference.

3.3 Evaluation

With regard to our in-context learning experiments, we randomly select four examples from the training dataset. This set of four examples is subsequently fixed to ensure the consistency of the experimental settings, mitigating any possible influence on the scores that may arise from variations in the quality of in-context learning examples.

Since decoder-based models are predominantly utilized to generate text (Fu et al., 2023), it is imperative to constrain their output in order to render them suitable for classification tasks. In this regard, we employ the Language Model Evaluation Harness (Gao et al., 2023) framework to develop a methodology for calculating the generation probability to predict each class as the subsequent token based on the given prompt. Consequently, the class with the highest generation probability is chosen as the predicted response.

4 Results

On the WiC task, the zero-shot scores of Mistral and Llama are close to random chance, probably because the models do not understand the expected output without any example (Table 3). On the WSD task, Llama zero-shot performance is relatively better even though the task seems harder (multiple possible answers instead of binary). Hu and Levy (2023) raised concerns that the zero-shot performance of language models on metalinguistic prompts, such as in our tasks, may not be representative of the model's capability. Therefore, it is more important to focus on the few-shot and fine-tuned results.

Mistral 7B was reported to outperform Llama 7B and 13B across a wide range of benchmarks (Jiang et al., 2023), but we find that Llama performs better in almost all experimental settings. On the other hand, GPT-4 in the zero-shot setting outperforms both Mistral 7B and Llama 13B in their few-shot settings. GPT-4 performs exceptionally well in the zero-shot setting of the WSD task but only gains very slight improvement with in-context learning. As GPT is a closed-source model, information on its model training procedure and data is not accessible. We cannot rule out the possibility that GPT may have been trained on the WSD task or with WSD training and test data.

On the WiC task, fine-tuned Llama 2 performs close to DeBERTa-V3-Large. However, the score difference is still large when compared to the bigger DeBERTa 1.5B, which achieves an accuracy of 76.4 on the WiC task (He et al., 2021).

5 Analysis

5.1 Importance of Encoders

We hypothesize that an encoder can help language models understand word meanings better. As such, we also run the WSD experiment on an encoder-decoder LLM, Flan-T5-XL (Chung et al., 2022). The encoder-decoder model is evaluated using the same method as the decoder-only models (Section 2.2). Since the Flan-T5 model was instruction-tuned, we compare it with the instruction-tuned versions of Mistral (`Mistral 7B Instruct 0.1`) and Llama 2 (`Llama 2 13B Chat`).

With much fewer parameters (3B), Flan-T5-XL outperforms Mistral 7B and even Llama 2 13B by

Model	# param	Setting	SE07	SE2	SE3	SE13	SE15	ALL	WIC
Mistral 7B	7B	Zero-shot	39.1	54.2	46.9	54.2	56.7	51.7	50.1
		Few-shot	55.4	66.7	64.3	67.5	72.3	66.3	53.1
		Fine-tuned	69.9	78.2	77.9	78.4	80.3	78.0	70.6
Llama 2 13B	13B	Zero-shot	51.6	62.9	64.1	64.4	69.3	63.7	50.0
		Few-shot	51.9	66.8	65.4	64.4	67.9	65.1	54.9
		Fine-tuned	75.2	79.8	77.7	79.1	81.7	79.1	74.1
GPT-4	N/A	Zero-shot	65.7	77.1	76.4	79.9	83.9	77.8	59.4
		Few-shot	66.2	77.6	75.3	80.9	83.5	77.9	71.5
DeBERTa-V3-Large	0.4B	Fine-tuned	76.9	81.0	79.8	81.0	84.9	81.0	74.4

Table 3: Experimental results on the WSD and WiC tasks, measured in F1 score and accuracy respectively. #param denotes the number of model parameters. GPT-4 is a closed-source model with an undisclosed number of parameters. ALL refers to the concatenation of SE07, SE2, SE3, SE13, and SE15.

substantial margins (Figure 3). Flan-T5-XL also scores higher than the non-instruction-tuned versions of Mistral and Llama (Table 4). Raffel et al. (2020) also previously reported that an encoder-decoder model performs better than a comparable decoder-only model on the WiC and other natural language understanding tasks. This may suggest that having an encoder architecture helps in understanding word meaning.

Another interesting observation is that the instruction-tuned versions of Mistral and Llama perform worse than their base versions without instruction tuning (Table 3). After being fine-tuned on the WSD training data, the instruction-tuned Mistral achieves an F1 score of 65.2 (versus 78.0 without instruction tuning), while the instruction-tuned Llama achieves an F1 score of 78.1 (versus 79.1 without instruction tuning). We believe this phenomenon can be explained by the "alignment tax" law (Ouyang et al., 2022), which states that reinforcement learning with human feedback (RLHF) comes at the cost of lower performance on certain downstream tasks.

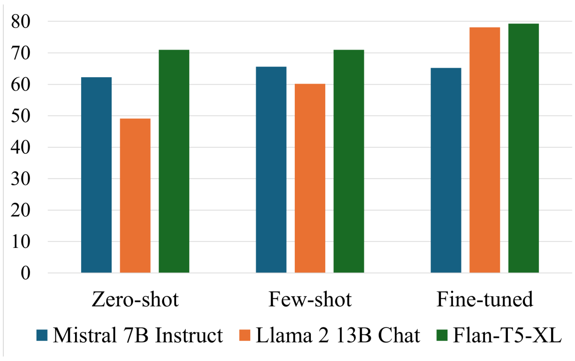


Figure 3: An encoder-decoder language model (Flan-T5-XL) with fewer parameters performs better than decoder-only language models.

Setting	SE07	SE2	SE3	SE13	SE15	ALL
Zero-shot	58.5	71.1	67.1	75.0	77.1	71.0
Few-shot	58.9	71.0	66.9	74.8	77.6	71.0
Fine-tuned	75.2	80.4	77.7	79.0	82.0	79.3

Table 4: F1 scores of Flan-T5-XL on the WSD task.

5.2 Prompt Styles

For the same goal of finding the correct sense of a word in a given context, there are different ways of prompting a language model. We investigate various types of prompts to utilize decoder-based language models for the WSD task.

Multiple-choice questions (MCQ) – Inspired by the question answering task, we formulate the WSD task as choosing the correct sense among the possible senses for the target word w. The prompt starts by specifying w and a list of sense definitions of w, followed by the sentence containing w. One difference from the usual prompt for the multiple-choice question answering task is that we list the options in numbers (e.g., 1, 2, ...) instead of letters since the number of possible senses can exceed 26. This is the prompt style used for our main results.

Sentence completion (COMP) – Decoder-based LLMs are pre-trained with the sentence completion task, so it is imperative to try formulating the downstream task as a sentence completion task too. The COMP prompt asks for the sense definition of w given the sentence, and expects the model to give the appropriate sense definition following the definition from WordNet.

Binary classification (BIN) – Encoder-only language models solve WSD by formulating it as a binary classification, predicting whether each sense definition of w is appropriate in the given context. The encoder models are effective in solving the WSD task with this formulation, so we are intrigued to investigate whether using this formulation can make the decoder-based models perform better.

The prompt examples are given in Appendix A. We run experiments by fine-tuning Mistral and Llama 2 on a subset of the training data that contains 10,000 target words. Despite the intuitive motivation, we find that the other prompt styles do not increase the performance of Mistral or Llama 2 (Figure 4).

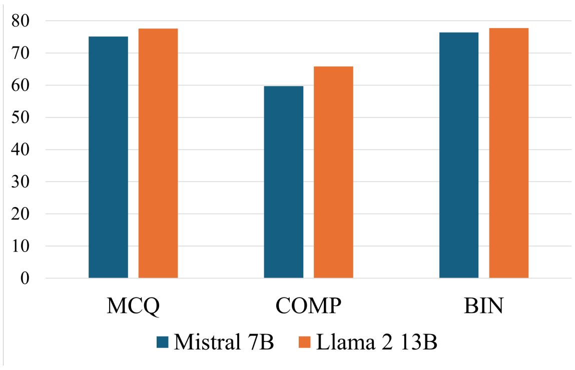


Figure 4: Performance of decoder-based models for different prompt styles.

5.3 Low-Resource Settings

LLMs are known to be good at adapting to a new task with few to no examples. In this experimental setting, we investigate whether decoder-based language models can outperform encoder-only ones when the amount of training data is limited. We fine-tune all models in the WSD task with subsets of the training data containing 10,000 and 20,000 target words. We found that even with less data, the encoder-only model still outperforms the decoder-only models (Figure 5).

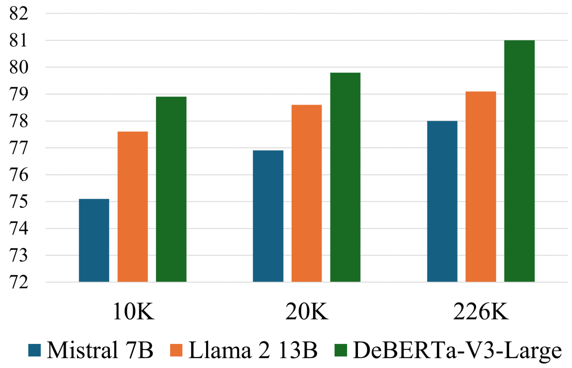


Figure 5: Performance of the language models on the WSD task with less training data.

6 Related Work

Prior work has reported performance comparisons of decoder-only models with encoder-only models on different tasks. Li et al. (2023) investigate the performance of decoder-based LLMs on financial text analytics tasks and report that GPT-4 with in-context learning has comparable performance to fine-tuned encoder-only models on financial sentiment analysis and lower performance on financial headline classification and relation extraction.

Yu et al. (2023) investigate Llama 2 and GPT-4 on named entity recognition, political ideology prediction, and misinformation detection. They also find that GPT-4 can achieve comparable performance on some of the tasks and perform worse on others. These works support our findings that LLMs still have limited capabilities on some downstream tasks, but they have not investigated LLMs' capabilities in understanding word meaning.

7 Conclusion and Future Work

In this paper, we demonstrate that decoder-only LLMs perform worse on word meaning comprehension than an encoder-only language model with vastly fewer parameters. We report the performance of Mistral 7B, Llama 2 13B, and GPT-4 on the WSD and WiC tasks and show that DeBERTa-V3-Large with vastly fewer parameters outperforms all aforementioned decoder-only models on both tasks. We discuss the importance of the encoder in understanding word meaning by running the same experiment on an encoder-decoder model, Flan-T5-XL, and find that it also outperforms the decoder-only models.

This work provides a concrete task on which encoder-only language models can still outperform decoder-only language models, but it is not meant as a study on neural network architecture. Future research can explore the optimal architecture and the roles of the encoder and decoder in natural language understanding. Additionally, investigating methods to enhance the capabilities of decoder-only models in comprehending word meaning is also a valuable research direction.

Limitations

Our research and analysis are focused on the English language. We only evaluated recent language models that fitted into our compute budget. Our experiments were carried out using the current state-of-the-art LLMs in both encoder-only and decoder-only architectures with their respective model sizes and training data as is. As previously stated, the decoder-only models were trained with quantization and parameter-efficient fine-tuning, following the common practice for decoder-only LLMs in the literature. Our finding is based on the optimal prompting method currently known for classification tasks for decoder-only models. Given these limitations, it is possible that a different conclusion can be reached in the future with improved language models, training methods, or prompting techniques.

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv*, 2210.11416.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL*, pages 1–5.

Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. 2023. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *arXiv*, 2304.04052.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. In *Proceedings of EMNLP*, pages 5297–5306.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *Proceedings of ICLR*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of ICLR*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of EMNLP*, pages 5040–5060.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv*, 2310.06825.

Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are ChatGPT and GPT-4 general-purpose solvers for financial text analytics? A study on several typical tasks. In *Proceedings of EMNLP*, pages 408–422.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Proceedings of NeurIPS*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *Proceedings of NeurIPS*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023b. Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of HLT*.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. In *7th Annual Conference on Robot Learning*.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of SemEval*, pages 288–297.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of SemEval*, pages 222–231.

OpenAI. 2023. GPT-4 Technical Report. *ArXiv*, 2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*, pages 27730–27744.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL*, pages 1267–1273.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*, pages 87–92.

Muhammad Reza Qorib and Hwee Tou Ng. 2023. System combination via quality estimation for grammatical error correction. In *Proceedings of EMNLP*, pages 12746–12759.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1).

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of EACL*, pages 99–110.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL*, pages 41–43.

Yang Song, Xin Cai Ong, Hwee Tou Ng, and Qian Lin. 2021. Improved word sense disambiguation with enhanced sense representations. In *Findings of EMNLP*, pages 4311–4320.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2307.09288.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Proceedings of NeurIPS*.

Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. 2023. Open, closed, or small language models for text classification? *arXiv*, 2308.10092.

Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S. Yu. 2023. Large language models for robotics: A survey. *arXiv*, 2311.07226.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of ACL: System Demonstrations*, pages 78–83.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. 2023. A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT. *arXiv*, 2302.09419.

Yilun Zhu, Joel Ruben Antony Moniz, Shruti Bhargava, Jiarui Lu, Dhivya Piraviperumal, Site Li, Yuan Zhang, Hong Yu, and Bo-Hsiang Tseng. 2024. Can large language models understand context? In *Findings of EACL*.

A Prompt Examples

A.1 MCQ Style for WSD

- **Prompt** :
  Given the following list of definitions of the

Hyper-parameter	Mistral 7B	Llama 2 13B	DeBERTa-V3-Large	FLAN-T5-XL
Batch Size	32	32	16	32
Learning Rate	1×10^{-5}	2×10^{-4}	8.5×10^{-6}	5×10^{-5}
Learning Rate Scheduler	cosine	cosine	linear	cosine
Optimizer	Paged AdamW 32bit	Paged AdamW 32bit	AdamW	Paged AdamW 32bit
Max Epoch	3	3	3	3
LoRA Alpha	128	128	-	128
LoRA Rank	64	64	-	64
LoRA Dropout	0.1	0.1	-	0.1
LoRA Target Modules	[q_proj, v_proj]	[q_proj, v_proj]	-	[q, v]

Table 5: Hyper-parameter values for fine-tuning the language models on the WSD task.

word “art”:
1. the products of human creativity; works of art collectively
2. the creation of beautiful or significant things
3. a superior skill that you can learn by study and practice and observation
4. photographs or other visual representations in a printed publication
Which definition is used by the word “art” in the bracket in the following sentence?
Sentence: “The (art) of change-ringing is peculiar to the English , and , like most English peculiarities , unintelligible to the rest of the world .”
Answer:

- **Expected Answer** :
  3

A.2 COMP Style for WSD

- **Prompt** :
  The definition of [art] in “The [art] of change-ringing is peculiar to the English , and , like most English peculiarities , unintelligible to the rest of the world .” is

- **Expected Answer** :
  “a superior skill that you can learn by study and practice and observation.”

A.3 BIN Style for WSD

- **Prompt** :
  Is it correct that the definition of the word “art” in “The (art) of change-ringing is peculiar to the English , and , like most English peculiarities , unintelligible to the rest of the world .” is “the products of human creativity; works of art collectively”?

- **Expected Answer** :
  no

B Compute Budget

All zero-shot and few-shot computations are done on a single NVIDIA A100 40GB GPU, while all fine-tuning computations are done on a single NVIDIA A100 80GB GPU.

Model	Time (HH:MM:SS)
Zero-shot	
Mistral 7B	00:13:42
Llama 2 13B	00:25:06
Flan-T5-XL	00:28:38
Few-shot	
Mistral 7B	00:59:07
Llama 2 13B	01:47:15
Flan-T5-XL	01:09:34
Fine-tune	
Mistral 7B	18:20:31
Llama 2 13B	48:18:06
Flan-T5-XL	06:02:48
DeBERTa	19:34:27

Table 6: Running times.

C Experimental Details

We describe our hyper-parameters’ search intervals for the decoder-only and encoder-decoder models

Hyper-parameter	Search Interval
Batch Size	{32, 64}
Learning Rate	$\{1, 5, 10, 20\} \times 10^{-5}$
Max Epoch	{3, 10}
LoRA Alpha	{16, 32, 64, 128}
LoRA Rank	{8, 16, 32, 64}
LoRA Target Modules	{[q_proj, v_proj], 'all-linear'}

Table 7: Hyper-parameter search intervals.

in Table 7. For the encoder-only model, we follow the final hyper-parameters of ESR (Song et al., 2021). The hyper-parameter search was done manually, so not all possible hyper-parameter values were investigated for all models. We describe the final hyper-parameters of the models in Table 5. All our results are from single-run experiments.

D Resources

We utilize open source code for our experiments. For the decoder-only and encoder-decoder models, our fine-tuning code is based on the Alpaca-Lora code base[2]. For the encoder-only model, we use the ESR code base[3]. For the evaluation, including the zero-shot and few-shot settings, we use the Language Model Evaluation Harness (LM-Eval) framework[4].

[2] https://github.com/tloen/alpaca-lora

[3] https://github.com/nusnlp/esr

[4] https://github.com/EleutherAI/lm-evaluation-harness/