Where do LLMs Encode the Knowledge to Assess the Ambiguity?

Hancheol Park **Geonmin Kim**
Nota Inc.
{hancheol.park,geonmin.kim}@nota.ai

Abstract

Recently, large language models (LLMs) have shown remarkable performance across various natural language processing tasks, thanks to their vast amount of knowledge. Nevertheless, they often generate unreliable responses. A common example is providing a single biased answer to an ambiguous question that could have multiple correct answers. To address this issue, in this study, we discuss methods to detect such ambiguous samples. More specifically, we propose a classifier that uses a representation from an intermediate layer of the LLM as input. This is based on observations from previous research that representations of ambiguous samples in intermediate layers are closer to those of relevant label samples in the embedding space, but not necessarily in higher layers. The experimental results demonstrate that using representations from intermediate layers detects ambiguous input prompts more effectively than using representations from the final layer. Furthermore, in this study, we propose a method to train such classifiers without ambiguity labels, as most datasets lack labels regarding the ambiguity of samples, and evaluate its effectiveness.

1 Introduction

Due to the unprecedentedly large scale of data and the enormous size of models that can be trained on it, the recently proposed large language models (LLMs) have been able to retain a significant amount of knowledge. Furthermore, through instruction tuning, LLMs have learned to provide natural language responses to input prompts for various natural language understanding (NLU) tasks, such as sentiment analysis and natural language inference (NLI), structured in the form of natural language instructions. This naturally enables LLMs to perform well on NLU tasks that were not observed during the instruction tuning (Ouyang et al., 2022; Sanh et al., 2022; Lee et al., 2023; Zhao et al., 2023). Nevertheless, LLMs often generate unreliable responses to users' inputs. Especially due to these reliability issues, it may be difficult for service providers to offer their LLMs, which have required significant investment to develop, leading to serious setbacks. Given the recent growth in the market for applications based on LLMs, this problem should be addressed.

Prompt	Premise: The newspaper publishes just one letter a week from a reader. Hypothesis: There are many letters submitted each week, but only one is chosen. Is this hypothesis entailed by the premise? Candidates: {entailment, neutral, contradiction} Answer:
Ambiguity Distribution	Entailment: 50% Neutral: 47% Contradiction: 3%
Generated Response	Entailment

Table 1: An ambiguous sample from ChaosNLI dataset (Nie et al., 2020). In this example, we might naturally assume that one of the numerous letters will be selected and published in the newspaper. However, if the newspaper is not well-known, the newspaper company may only receive one or two letters from readers each week. Therefore, we cannot necessarily conclude that the hypothesis is correct (i.e., neutral). Here, "ambiguity distribution" refers to the label distribution obtained from the evaluations of 100 annotators.

The most well-known cause for generating unreliable responses is hallucination behavior. This refers to the behavior where LLMs respond with a tone of high confidence in incorrect information (Azaria and Mitchell, 2023; Zhao et al., 2023;

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 445–452
January 19–24, 2025. ©2025 Association for Computational Linguistics

Huang et al., 2024). This issue has been extensively addressed by numerous researchers (Azaria and Mitchell, 2023; Huang et al., 2024). Another reason is the response behavior of LLMs, which often provide a single biased answer to an ambiguous question that could have multiple correct answers, as shown in Table 1. Ideal LLMs should indicate whether such questions are ambiguous and encourage alternatives such as using multi-label classification models or judgments from experts to help users make better decisions without bias. Particularly, since it is well-known that numerous ambiguous samples exist in NLU tasks (Uma et al., 2021), it is crucial to determine whether a given sample is ambiguous. Nevertheless, research on determining whether input prompts are ambiguous or not has not been relatively well-explored, and only a few impractical methods have been proposed (Lee et al., 2023; Portillo Wightman et al., 2023).

In this study, we discuss methods for classifying whether input prompts for NLU tasks are ambiguous or not before generating responses. Traditionally, NLU tasks have been addressed as classification problems in encoder-based language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Park and Park (2023) discovered that in the embedding space represented by an intermediate layer of a fine-tuned encoder-based language model (i.e., hidden states corresponding to the [CLS] tokens from each layer), ambiguous samples are located close to samples with related labels. However, these relationships disappear in higher layers, and samples with the same label are close together but far from samples with different labels. If intermediate representations of LLMs (i.e., the hidden states of the last input tokens) exhibit these characteristics, then it would be feasible to classify ambiguous prompts easily using a classifier based on these intermediate representations. However, instruction-following LLMs are trained as autoregressive language models, and they have learned numerous tasks simultaneously rather than a specific task. Therefore, it is uncertain whether the same characteristics observed in the intermediate layers of encoder-based models would similarly manifest.

To address this research question, we first construct datasets to train and evaluate the ambiguity of input prompts using existing datasets from sentiment analysis and NLI tasks. The ambiguity of each sample is determined based on evaluations from multiple annotators for that sample. Using such dataset, we train a classifier that uses a representation from a layer of LLMs as input. The experimental results demonstrate that using representations from intermediate layers classifies ambiguous samples with significantly higher accuracy than those from the final layer. This suggests that LLMs encode knowledge about ambiguity in their intermediate layers.

Furthermore, we find that the accuracy in detecting ambiguous samples significantly decreases when training a classifier with a combination of datasets from various NLU tasks. We also observe that performance significantly decreases when evaluated on datasets from tasks or domains different from those used for training. This suggests that the definition of ambiguity is both task- and domain-dependent in NLU tasks. Consequently, a challenge arises in acquiring training datasets for each task or domain to assess ambiguity. In particular, unlike the datasets used in this study, most datasets do not provide evaluation information from individual annotators. To address this data scarcity issue, we also propose a loss function that uses training dynamics (i.e., the phenomenon where a deep learning model learns easy samples early in training and difficult samples later.) (Arpit et al., 2017; Swayamdipta et al., 2020). We assume that ambiguous samples would be difficult examples because it is challenging for the model to determine which label to learn for them. In this study, we demonstrate that by using this loss function, it is possible to create a classifier capable of assessing ambiguity using readily available datasets annotated with single labels for a given NLU task.

2 Related Work

In encoder-based language models, various calibration methods have been proposed to adjust the probability distribution from the classifier so that the entropy of the probability distribution is high for ambiguous samples (Wang et al., 2022; Park and Park, 2023; Park et al., 2024). Unlike classification tasks in encoder-based models, LLMs do not provide probability distributions for labels. Instead, they offer probabilities for the next token. These probabilities are based on a text generation perspective, making them difficult to interpret as probabilities for labels. To address this issue, various methods have been proposed that repeatedly generate responses and aggregate them to produce probability distributions (Lee et al., 2023; Portillo Wightman

NLI Task
Training and validation sets Premise: [Premise] Hypothesis: [Hypothesis] Does the premise entail the hypothesis? Options: entailment, contradiction, and neutral Answer:
Evaluation set Can we conclude the following hypothesis from the premise? Premise: [Premise] Hypothesis: [Hypothesis] Candidates: entailment, contradiction, and neutral Answer:
Sentiment Analysis
Training and validation sets Text: [Text] What is the sentiment of this text? Options: positive, negative, and neutral Answer:
Evaluation set Input text: [Text] How would this input text be described in terms of sentiment? Options: positive, negative, and neutral Answer:

Table 2: Examples of prompt templates that we used in this study. Different multiple templates were used to train and evaluate the models; however, only one example was described for each stage due to space limitations.

et al., 2023). These approaches are known to help LLMs provide somewhat calibrated distributions. However, running LLMs multiple times—ranging from a few dozen to a few hundred times—to obtain distributions is impractical for real-world applications. Therefore, this study discusses methods for determining ambiguity in a single inference, as opposed to those that repeatedly generate responses to obtain distributions. Furthermore, there has been an attempt to have LLMs generate confidence values for their responses in textual form (Lin et al., 2022).

3 Proposed Method

In this study, we verify whether the representations (i.e., hidden states of the last input tokens) from intermediate layers contain knowledge that can judge the ambiguity of input prompts. To do this, we first automatically construct annotated datasets indicating whether each input prompt is ambiguous or not across various NLU tasks (§3.1). Then, we train classifiers that use representations of input prompts from the instruction-following LLMs as inputs (§3.2).

3.1 Datasets for Detecting Ambiguity

We first create datasets where each sample is annotated to indicate whether it is ambiguous or not. To automatically construct these datasets, we use existing datasets that are used for multi-label classification or those that contain multiple annotations per sample. Specifically, we use three datasets for sentiment analysis and NLI tasks. For sentiment analysis, we employ the GoEmotions dataset (Demszky et al., 2020), which is a multi-label emotion and sentiment analysis dataset. For the NLI tasks, we use the SNLI (Bowman et al., 2015) and MNLI development and test datasets (Williams et al., 2018), which contain multiple annotations (5 or 100) per sample.

For multi-label datasets, samples annotated with multiple labels are considered ambiguous. For the NLI datasets, we follow criteria from previous research (Jiang and de Marneffe, 2022) to classify samples as ambiguous or non-ambiguous. If all five annotators provide the same label, the sample is considered non-ambiguous. If two labels receive at least two votes each (e.g., 3/2/0 or 2/2/1), the sample is considered ambiguous. Additionally, a subset of samples from SNLI and MNLI is annotated by 100 annotators in the ChaosNLI dataset (Nie et al., 2020). We use this information to annotate each sample: samples where the majority label receives more than 80 votes out of 100 are considered unambiguous, while samples where the majority label receives less than 60 votes are considered ambiguous. As in the previous study (Jiang and de Marneffe, 2022), samples receiving between 60 and 80 votes are excluded because it is difficult to determine whether they are ambiguous or not.

Finally, the texts in the entire dataset are modified into the format of input prompts for LLMs. To simulate scenarios where actual users employ instruction-following LLMs, the prompts used during training are constructed differently from those used during the evaluation stage. Examples of the prompt templates we used are illustrated in Table 2. The statistics of the constructed dataset

	SNLI		MNLI		GoEmotions	
	Unamb.	**Amb.**	**Unamb.**	**Amb.**	**Unamb.**	**Amb.**
Train	1,935	1,935	2,160	2,160	1,683	1,683
Validation	215	215	240	240	186	187
Test	536	536	602	602	468	467

Table 3: Statistics of our datasets. "Unamb." and "Amb." stand for unambiguous sample and ambiguous sample, respectively.

are described in Table 3[1].

3.2 Classifier for Detecting Ambiguous Samples

We train a classifier that uses a representation from a layer of an LLM as input to determine whether input samples are ambiguous or not. This representation corresponds to the hidden state of the last token in the input prompt. If our hypothesis hold true, representations from intermediate layers should effectively distinguish between ambiguous and unambiguous samples, leading to high classification accuracy. In this work, we employ a three-layer multi-layer perceptron (MLP) as the classifier, with ReLU activation functions applied to each layer.

4 Experiments

In this section, we quantitatively evaluate how helpful representations from intermediate layers are in judging ambiguity.

4.1 Experimental Settings

As instruction-following LLMs, we use instruction-tuned OPT-IML-1.3B (Iyer et al., 2023), LLaMA 2-7B and 13B (Touvron et al., 2023). These models have 24, 32, and 40 layers and 2,048, 4,096, and 5,120 hidden units, respectively. We use three-layer MLP classifiers to detect ambiguous samples. For OPT-IML-1.3B, the configuration is 2,048-512-128-2 for the hidden units. For LLaMA 2-7B, the configuration is 4,096-1,024-256-2, and for LLaMA 2-13B, it is 5,120-1,024-256-2 hidden units.

The classifiers mentioned earlier were all trained with a batch size of 64, and the learning rate was set to 5e-3 with a linear decay. The AdamW optimizer (Loshchilov and Hutter, 2019) was used to update the parameters of all classifiers, with the weight decay set to 0.01. The optimal numbers of epochs for classifiers using representations from LLaMA 2-7B and 13B were selected between 20 and 25 epochs based on accuracy on the validation sets, while for classifiers using representations from OPT, the optimal numbers of epochs were chosen between 35 and 40. In this study, we use accuracy as the main evaluation metric.

	SNLI	MNLI	GoEmo.
OPT (1.3B)			
24th layer	78.48	86.93	54.33
20th layer	**79.01**	88.82	**57.97**
16th layer	77.12	85.34	50.05
12th layer	71.30	**88.87**	54.26
LLaMA 2 (7B)			
32th layer	69.77	86.90	55.72
28th layer	72.76	83.17	55.90
24th layer	75.10	**88.01**	55.61
20th layer	75.87	85.20	**57.29**
16th layer	**77.12**	87.57	49.98
LLaMA 2 (13B)			
40th layer	77.74	86.74	56.86
36th layer	78.64	86.63	**57.90**
32th layer	78.51	86.99	56.68
28th layer	78.61	86.88	55.62
24th layer	**78.70**	**88.73**	54.83
20th layer	74.84	86.85	55.86

Table 4: Evaluation results for classifying ambiguous samples across three LLMs of different sizes. Each classifier was trained and evaluated on the samples from each dataset without combining other datasets.

4.2 Results

Since LLMs have a large number of layers, experiments are conducted on some intermediate layers, including the final layer, similar to a previous study that analyzed the characteristics of intermediate layers in LLMs (Azaria and Mitchell, 2023). As shown in Table 4, we can observe that using representations from the intermediate layers is more effective in determining the ambiguity of samples than using representations from the final layers.

[1] These datasets are available at `https://github.com/hancheolp/ambiguity_detection`.

	SNLI	MNLI	GoEmo.
32th layer	66.79	56.98	48.87
28th layer	65.95	56.89	48.24
24th layer	70.52	55.73	50.16
20th layer	**71.27**	56.73	49.63
16th layer	70.62	**58.22**	**52.30**

Table 5: Evaluation results when the classifiers are trained only on NLI datasets. The sentiment analysis dataset was not used for training. In this case, we use representations from LLaMA 2-7B.

	SNLI	MNLI	GoEmo.
32th layer	50.00	50.00	50.05
28th layer	50.00	50.00	49.95
24th layer	70.80	55.65	54.97
20th layer	**71.83**	54.82	**55.72**
16th layer	71.55	**58.14**	53.69

Table 6: Evaluation results when the classifier are trained on all datasets combined. In this case, we use representations from LLaMA 2-7B.

	SNLI	MNLI
32th layer	68.66	60.88
28th layer	65.76	54.57
24th layer	72.67	61.71
20th layer	**74.16**	69.27
16th layer	68.28	**70.68**

Table 7: Evaluation results when the proposed loss function described in Equation 1 are used.

It has been also confirmed that detecting ambiguous samples in more subjective tasks such as sentiment analysis is more challenging than in NLI tasks. Furthermore, since the optimal intermediate layer varies across tasks and models, identifying such layer for each task appears to be a new challenge for the future.

5 Discussion

In this section, we discuss two research questions. First, whether learning ambiguity in one task enables judgment of ambiguity in another task. To address this, we combined two NLI datasets and trained classifiers with representations from various layers, then evaluated using samples from sentiment analysis tasks. As shown in Table 5, we found that the classifiers are unable to judge ambiguity at all for the tasks that they were not trained on (i.e., sentiment analysis). Notably, when combining datasets from different domains of the same task for training, the performance degraded compared to when this was not done (see Tables 4 and 5).

Furthermore, we found that performance degraded across all tasks and domains when training on the combined datasets. (see Table 4 and Table 6). These results suggest that ambiguity is task and domain-specific. Therefore, the challenge arises of creating a dataset for each task and domain to address this issue. To tackle this, we propose a loss function based on well-known training dynamics:

$$L(x) = \lambda(-logp_{gt}) + (1-\lambda)(1-p_{gt})(-logp_{amb}) \tag{1}$$

where x is the input prompt, p_{gt} is the predicted probability for the ground truth label of the original task (e.g., for an NLI task, the probability for one of the labels: entailment, neutral, or contradiction) and p_{amb} is the probability that a given sample is ambiguous. Both p_{gt} and p_{amb} are calculated by passing the output logits of a classifier that uses representations from an LLM as input through a softmax layer. To achieve this, the number of output neurons in the final layer of the classifier is adjusted to be the number of labels for each task plus one (for the label indicating that a given sample is ambiguous). It is known that deep learning models start by learning easy samples in the early stages of training and progress to harder samples later on (Arpit et al., 2017). Therefore, we assume that if the p_{gt} value is low in the early stages of training, the sample is ambiguous and difficult to judge with a specific label. The hyperparameter λ is tuned using a small set of labeled validation samples that indicate whether a sample is ambiguous or not. As shown in Table 7, it can be observed that by training with the proposed loss function, it is possible to train classifiers to determine ambiguity even without labels for ambiguity.

The second research question is whether, similar to encoder-based models, the embedding spaces of the intermediate layers better represents the ambiguity of samples compared to the final layer. To address this, we investigated how the representations of samples that are annotated as "entailment" in the original dataset but deemed ambiguous through this study are distributed in the embedding space. As shown in Figure 1, ambiguous samples at the lower layers are positioned between two different labels (i.e., entailment and contradiction), while in the higher layers, these samples are largely distanced from those that correspond to "contradic-

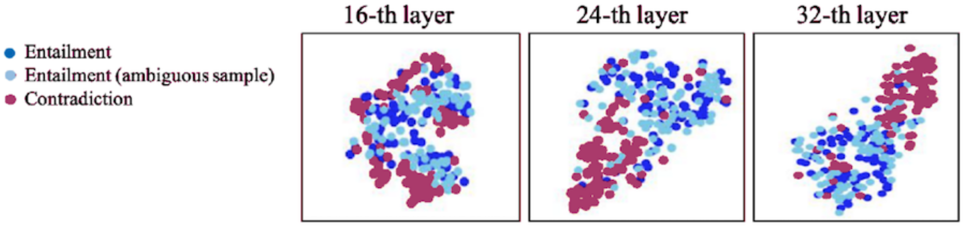

Figure 1: Visualization of feature representations from SNLI samples using t-SNE. The representations are extracted from layers in LLaMA 2-7B.

tion". Therefore, similar to encoder-based language models, we find that ambiguous samples are better represented in the lower layers of decoder-based LLMs as well.

6 Conclusion

In this study, we found that using representations from intermediate layers allows for a more accurate assessment of the ambiguity in input prompts. This enables LLMs to evaluate the ambiguity of inputs before generating responses for tasks that require such judgment. In future work, we will explore methods for automatically annotating the ambiguity of samples in NLU datasets, particularly when evaluation results from multiple annotators per sample are unavailable. Furthermore, we will investigate techniques for automatically selecting the optimal intermediate layer that most effectively supports the assessment of input prompt ambiguity.

Limitations

We have verified that using representations from the intermediate layers of LLMs are more helpful to capture ambiguous samples than the knowledge from the final layer. However, the method for selecting the optimal layer was not addressed in this study. Additionally, since the definition of ambiguity varies across tasks and domains, there is a need to construct datasets that assess ambiguity of samples for each task and domain. We discussed a method to address these issue, but there is a need for improvement as the performance is lower than using datasets designed specifically for judging ambiguity. In this study, we explored relatively small-sized LLMs with fewer than 13 billion parameters, but future research may need to investigate larger-scale models.

Ethics Statement

In this study, ethical concerns are considered minimal because we used well-established datasets that have been widely used by numerous researchers without any issues to date. However, some samples, such as those used in sentiment analysis, may contain examples that could evoke negative emotions in readers.

Acknowledgments

This work was supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

References

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of 34th International Conference on Machine Learning*, pages 233–242.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2023. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374.

Noah Lee, Na Min An, and James Thorne. 2023. Can large language models capture dissenting human voices? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4585, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 27730–27744.

Hancheol Park, Soyeong Jeong, Sukmin Cho, and Jong C. Park. 2024. Self-knowledge distillation for learning ambiguity. *arXiv preprint arXiv:2406.09719*.

Hancheol Park and Jong Park. 2023. Deep model compression also helps models capture ambiguity. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6893–6905, Toronto, Canada. Association for Computational Linguistics.

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362, Toronto, Canada. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

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 preprint arXiv:2307.09288*.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Yuxia Wang, Minghan Wang, Yimeng Chen, Shimin Tao, Jiaxin Guo, Chang Su, Min Zhang, and Hao Yang. 2022. Capture human disagreement distributions by calibrated networks for natural language inference. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1524–1535, Dublin, Ireland. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.