SFT For Improved Text-to-SQL Translation

Ankit Agrahari * , Puneet Kumar Ojha , Abhishek Gautam , Parikshit Singh

Posted Date: 13 February 2024

doi: 10.20944/preprints202402.0693.v1

Keywords: Text-to-sql

 doi:10.20944/preprints202402.0693.v1

Article

SFT For Improved Text-to-SQL Translation

Puneet Kumar Ojha *, Abhishek Gautam, Ankit Agrahari and Parikshit Singh

Independent Researcher; devxabhishek@gmail.com; ankitagr2312@gmail.com; parikshitcs0072@gmail.com
* Correspondence: puneetkumar.2705@gmail.com

Abstract: Large Language Models (LLMs) have proved significant proficiency when comes to code generation especially in Structured Query Language (SQL) for databases and recent successful Text-to-SQL method involves fine-tuning pre-trained LLMs for SQL generation tasks. Transforming natural language text into SQL queries, has been attempted to solve with various learning techniques including Few-shot learning Wang et al. (2019), fine tuning. In this paper we propose Supervised fine-tuning (SFT) as a better alternative for learning technique for text-to-SQL generation task using Code-Llama that pushes state of art accuracy on spider test suite to 89.6% on dev set which represent first instance of surpassing the earlier best-in-class with 5.5% higher score and 86.8% of exact match accuracy on dev set. Furthermore, we demonstrate that properly prompted LLM along with SFT provides far fewer hallucinations and much more robust LLM that can be used as a general tool for any text-to-SQL generation use case.

Keywords: Text-to-sql

1. Introduction

Automatic SQL generation from natural language has been one of the most crucial needs to enhance database accessibility without the knowledge of data definition or querying methods. With advancement in LLM's conversational chatbots have bloomed and come up with easier ways to access the database and provide better data analytics.

Several training and optimization techniques have been demonstrated for achieving decent performance in text-to-SQL generation. RESDSQL Li et al. (2023) for example utilizing a distinct approach for connecting database schemas and dissecting the structure of queries, employing an improved encoding process with ranking and a decoding framework aware of skeleton structure, this was primarily achieved with the encoder-decoder model T5 by fine tuning the model in two stages cross encoder training followed by seq2seq training. PICARD Scholak et al. (2021) applied an innovative method involving progressive parsing to restrict auto-regressive decoding, while RASAT Qi et al. (2022) merged self-attention mechanisms aware of database schemas with controlled auto-regressive decoders within the model's framework.

The development of massive LLMs such as GPT-3 Brown et al. (2020), PaLM Chowdhery et al. (2022), ChatGPT Cha (2023), GPT-4 OpenAI (OpenAI), and PaLM-2 Google (Google), each with billions of parameters, has led to significant strides in zero-shot and few-shot learning techniques, particularly in-context learning Wei et al. (2022). These approaches, especially few-shot prompting, are advantageous over fine-tuning because they require less computational power, are less likely to overfit training data, and can easily adjust to new datasets. This is especially beneficial for converting text into SQL queries due to the various dialects of SQL. However, a downside is that their performance may not be as high as desired. As an illustration, while CodeX Chen et al. (2021) and ChatGPT Liu et al. (2023) have demonstrated encouraging outcomes in converting text into SQL queries using in-context learning methods, they still fall short compared to fine-tuned models with moderately sized LLMs. SQL-PALM Sun et al. (2023), the prior best-in-class, demonstrated considerable enhancements by employing both few-shot learning and fine-tuning on the PALM-2 Google (Google); Sun et al. (2023) model using the Spider dataset. Meanwhile, DIN-SQL adopts a least-to-most prompting strategy Zhou et al. (2022), dividing the Text-to-SQL task into smaller elements such as connecting schemas, categorizing queries,

 doi:10.20944/preprints202402.0693.v1

and breaking them down. Subsequently, it employs few-shot prompting specifically for each sub-task with customized prompts. Notably, DIN-SQL Pourreza and Rafiei (2023) is the first to surpass the effectiveness of fine-tuned state-of-the-art models in evaluations using a few-shot prompting approach.

In this paper we propose, Supervised fine-tuning as another option to regular fine-tuning for training LLM for better text-to-SQL generational task. We have used open Llama-V2 due to its several architectural advantages including pre-normalization, SwiGLU activation, and Rotary embeddings. The model, when trained, attained top-tier results on the Spider development set boasting a notable execution accuracy of 89.6% alongside a precise match accuracy of 86.8%.

2. SFT for Text-to-SQL

2.1. LLM's training techniques

2.1.1. Few shot prompting

LLM's prompting is a method of constraining a model to give desired outputs. First identified in Brown et al. (2020), in-context learning leverages the capability of few-shot learning and zero-shot through prompting. This method integrates a limited set of examples and instructions inside the prompt, creating a 'context' that enables LLMs to adapt to new tasks and examples without any alterations to the model. As highlighted in Wei et al. (2022), the efficacy of few-shot prompting is particularly more evident in LLMs above a specific size margin. The achievement of in-context learning has led to the innovation of advanced prompting techniques like two chain-of-thought prompting (CoT) Wei et al. (2022), least-to-most prompting Zhou et al. (2022), and self-consistency prompting Wang et al. (2022), which are efficient strategies for large-shot adaptation. For the Llama-7b model we were only able to get an accuracy score of 11.8% out-of-the box from few-shot prompting only. Although the model was able to generate the output but was very poor at understanding how to put joins and multiple clauses for filtering through the data.

2.1.2. Fine-tuning

Fine-tuning is a training method where the model parameters are changed slightly for a downstream task to improve the models performance on that task. LLMs have demonstrated exceptional capabilities across a range of difficult tasks, like those in BIG-bench Srivastava et al. (2022). This is largely attributed to the extensive knowledge gained from large-scale pre-training, which is then enhanced by instruction-based fine-tuning on various tasks, known as FLAN-fine-tuning. Fine-tuning has proven to be very much effective in neural networks, however in LLM's it often induces a lot of hallucination after output is generated in smaller models, resulting in poor model's generation quality and overall poorly generated queries, we measured an accuracy of 45.5% only when trained with fine-tuning (see Table 1).

2.1.3. Supervised fine tuning

SFT, or Supervised Fine Tuning, entails modifying a model for a new downstream task by fine-tuning the LLM with labeled data. In general, the entire context is passed at once but the final loss is computed only over the label (see Figure 1) that the model is required to generate this allows for the model to learn only the syntactic generation of label rather than entire statement, in our case the schema and question were masked and loss was computed only on the generated query. This allowed for much better learning and text-to-sql generations. Our efforts led to an impressive achievement of 89.4% accuracy (Table 1) on the Spider dev-set.

 doi:10.20944/preprints202402.0693.v1

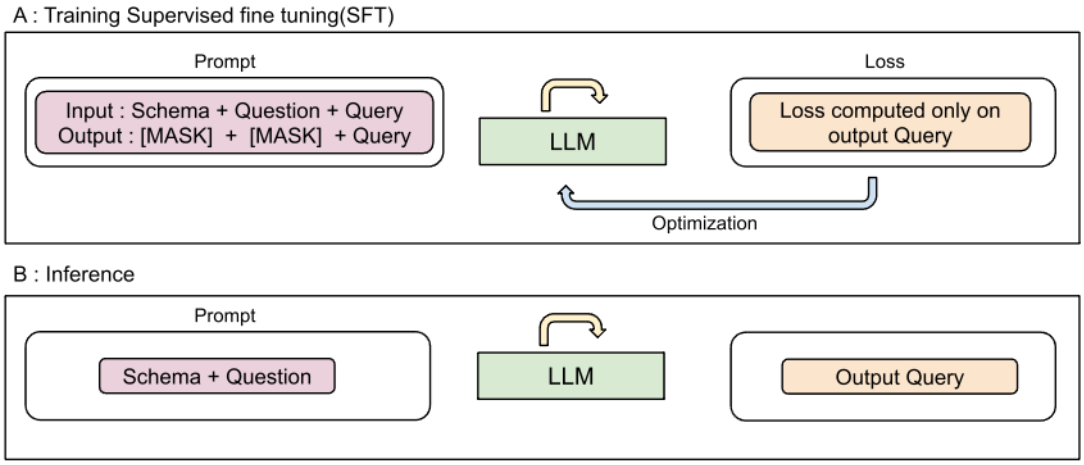

Figure 1. A. Supervised Fine Tuning (SFT) of Llama model on spider **B.** Inference prompting on Llama model.

3. Experiments

3.1. Dataset

Analyzed the extensive, cross-domain Text-to-SQL benchmark known as Spider Yu et al. (2018), consisting of 7000 training examples across 166 databases and 1034 evaluation samples ('Dev split') spanning 20 databases. Spider-SYN Gan et al. (2021), an intricate iteration of the Spider Dev dataset, is generated by manually substituting synonyms within the natural language queries. Spider-realistic Deng et al. (2021) selects 508 text-SQL pairings from the Spider Dev Split, omitting direct references to column names in the natural language questions. Additionally, Spider-DK Gan et al. (2021) draws 535 question-SQL pairs from 10 databases in the Spider Dev split, adding domain knowledge to these pairings.

3.2. Model

Open Llama-V2 Touvron et al. (2023) is an open-source replication of the Llama model. Llama has shown very promising results across several benchmarks despite its smaller size compared to GPT-4, GPT-3 and chat-GPT models. For our task we chose Llama V2 with 7 billion parameters as roughly being the sweet spot for decent size to performance tradeoffs.

Code Llama Rozière et al. (2023) is fine tuned on coding data representing a constellation of large language models , for large contexts ,code infilling, and zero-shot instruction following for programming tasks.

3.3. Baselines

For fine-tuning approaches, SQL-PALM Sun et al. (2023) leverages the transformer-based PALM-2 Google (Google) model, applying both fine-tuning and few-shot techniques for the text-to-SQL task. PICARD Scholak et al. (2021) employs incremental parsing to limit auto-regressive decoding, and RASAT Qi et al. (2022) is a transformer model that fuses relation-aware self-attention with controlled auto-regressive decoders. Additionally, RESDSQL Li et al. (2023) innovatively separates schema linking from skeleton parsing, employing a decoding framework that is aware of the query structure and an encoding framework enhanced with ranking.

In the domain of in-context learning, a detailed evaluation of CodeX and GPT-3's text-to-SQL capabilities is presented in Rajkumar et al. (2022), while an in-depth analysis of ChatGPT's Cha (2023) performance is offered in Liu et al. (2023). DIN-SQL Pourreza and Rafiei (2023) methodically decomposes Text-to-SQL into subtasks such as employing few-shot prompting with GPT-4 OpenAI

 doi:10.20944/preprints202402.0693.v1

(OpenAI) in tasks such as query classification, schema linking, self-correction , SQL generation, and decomposition. The Self-debugging methodology Chen et al. (2023) incorporates error messages into prompts and executes successive iterations of few-shot prompting for error rectification. According to the data in Table 2, ChatGPT Liu et al. (2023) utilizes the prompting techniques suggested by OpenAI. It's noteworthy that Self-debugging Chen et al. (2023) focuses exclusively on Execution accuracy (EX).

3.4. Evaluation

We have utilized two primary evaluation metrics on the Spider test-suite: execution accuracy (EX) and exact match (EM). Execution accuracy (EX) evaluates if the predicted SQL query aligns precisely with the gold SQL query through their conversion into a specialized data structure. In contrast, exact match (EM) juxtaposes the outcomes of executing the predicted SQL query against the gold SQL query. It's worth highlighting that, in contrast, the EX metric is influenced by the values generated within the query, whereas the EM metric remains unaffected by this factor.

4. Results

We demonstrate execution accuracy of various learning methods on the Llama-7B model in Table 1. We can clearly see from the results in the table that Supervised fine tuning far outperforms regular fine-tuned model. In our testing, fine-tuning smaller models resulted in much more hallucinations and as such resulted in poor performance as compared to the SFT counterpart.

Table 1. Comparison of Llama-V2 7B performance on few-shot learning, fine-tuning and supervised finetuning on test suite accuracy spider dev set.

Methods	Easy	Medium	Hard	Extra hard	All
Few shot (out of box)	29.4	9.0	4.0	1.8	11.8
Fine Tuning	66.1	42.6	38.7	29.5	45.5
Supervised fine tuning	94.8	91.0	86.2	80.1	89.4

We delve into how our proposed method fares across different levels of difficulty in SQL query generation. These levels are determined by various factors, including: SQL keywords used, the incorporation of attributes aggregations or selections and the utilization of nested sub-queries. Table 2 illustrates comparative performance of proposed method against a standard few-shot prompting approach using CodeX-davinci and GPT-4, as well as against DIN-SQL Pourreza and Rafiei (2023) and the prior SOTA, SQL-PALM, on the Spider development set. Our method consistently outshines the alternatives at all levels of difficulty, showing significant improvements. This indicates that our method does not exhibit a bias towards any specific category of difficulty. Our model specifically improved in generation of hard and extra hard SQL's resulting in significant performance improvements over the alternatives, and previous SOTA by almost 11% and being almost 50 times smaller.

Table 2. Accuracy on the Spider dev split test-suite: SQL results are classified into different levels. The first two rows represent the conventional few-shot prompting approach. Beginning six rows are from Sun et al. (2023)

Methods	Easy	Medium	Hard	Extra hard	All
Few-shot CodeX-davinci	84.7	67.3	47.1	26.5	61.5
Few-shot GPT-4	86.7	73.1	59.2	31.9	67.4
DIN-SQL Li et al. (2023) CodeX-davinci	89.1	75.6	58.0	38.6	69.9
DIN-SQL Li et al. (2023) GPT-4	91.1	79.8	64.9	43.4	74.2
Few-shot SQL-PaLM2	93.5	84.8	62.6	48.2	77.3
Fine-tuned SQL-PaLM2	93.5	85.2	68.4	47.0	78.2
SFT Llama 7b V2(Ours)	93.5	89.9	85.6	80.1	88.5
SFT Code Llama7b(Ours)	96.0	90.8	90.2	75.9	89.6

 doi:10.20944/preprints202402.0693.v1

Table 3. Comparison of various models performance on spider dev-set for text-to-SQL, non-sequence evaluation metrics include Exact Match (EM) and Execution Accuracy (EX) and seq2seq methods performance from Li et al. (2023)

Approach	EM(dev set)	EX(dev se
Non-seq2seq methods		
GRAPPA + RAT-SQL Yu et al. (2020)	73.4	-
NatSQL + RAT-SQL + GAP Gan et al. (2021)	73.7	75.0
GRAPPA + SMBOP Deng et al. (2021)	74.7	75.0
RoBERTa + DT-Fixup SQL-SP Xu, Kumar, Yang, Zi, Tang, Huang, Chi, Cheung, Prince, and Cao (Xu et al.)	75.0	-
ELECTRA + LGESQL Cao et al. (2021)	75.1	-
S2SQL + ELECTRA Hui et al. (2022)	76.4	-
Seq2seq methods		
T5-3B Scholak et al. (2021)	71.5	74.4
PICARD + Scholak et al. (2021)	75.5	79.3
PICARD + RASAT Qi et al. (2022)	75.3	80.5
RESDSQL-3B	78.0	81.8
RESDSQL-3B + NatSQL	80.5	84.1
Our proposed method		
Llama-7B v2 (SFT)	86.7	88.5
Code Llama	86.8	89.6

5. Conclusion

We present a LLM based model SFT Code Llama-7B and SFT Open Llama 7B v2 for text-to-SQL task which leverages Llama transformer supervised fine tuning. We demonstrate significant performance improvements by simply changing the learning method to adopt the model to new data. Our model being even 50 times smaller compared to PALM-2 outperforms the competition setting a newer SOTA score on the spider test suite of 89.6% in execution accuracy and 86.8% in exact match. More importantly SFT Code-Llama-7B was able to produce very decent results, when prompted in the exact same way demonstrating the efficacy and understanding of the model towards text-to-SQL generation task.

References

2023. Chatgpt.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020, 5. Language models are few-shot learners. *Advances in Neural Information Processing Systems 2020-December*.

Cao, Ruisheng, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021, 6. Lgesql: Line graph enhanced text-to-sql model with mixed local and non-local relations. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2541–2555. doi:10.18653/v1/2021.acl-long.198.

Chen, Mark, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew,

Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021, 7. Evaluating large language models trained on code.

Chen, Xinyun, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023, 4. Teaching large language models to self-debug.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022, 4. Palm: Scaling language modeling with pathways.

Deng, Xiang, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-sql. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 1337–1350. doi:10.18653/v1/2021.naacl-main.105.

Gan, Yujian, Xinyun Chen, Jinxia Xie, Matthew Purver, John R. Woodward, John Drake, and Qiaofu Zhang. 2021, 9. Natural sql: Making sql easier to infer from natural language specifications. *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, 2030–2042. doi:10.18653/v1/2021.findings-emnlp.174.

Google. Palm 2 technical report.

Hui, Binyuan, Ruiying Geng, Lihan Wang, Bowen Qin, Yanyang Li, Bowen Li, Jian Sun, and Yongbin Li. 2022, 3. S^2sql: Injecting syntax to question-schema interaction graph encoder for text-to-sql parsers. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1254–1262. doi:10.18653/v1/2022.findings-acl.99.

Li, Haoyang, Jing Zhang, Cuiping Li, and Hong Chen. 2023, 2. Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023 37*, 13067–13075. doi:10.1609/aaai.v37i11.26535.

Liu, Aiwei, Xuming Hu, Lijie Wen, and Philip S. Yu. 2023, 3. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability.

OpenAI. Gpt-4 technical report.

Pourreza, Mohammadreza and Davood Rafiei. 2023, 4. Din-sql: Decomposed in-context learning of text-to-sql with self-correction.

Qi, Jiexing, Jingyao Tang, Ziwei He, Xiangpeng Wan, Yu Cheng, Chenghu Zhou, Xinbing Wang, Quanshi Zhang, and Zhouhan Lin. 2022, 5. Rasat: Integrating relational structures into pretrained seq2seq model for text-to-sql. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, 3215–3229. doi:10.18653/v1/2022.emnlp-main.211.

Rajkumar, Nitarshan, Raymond Li, and Dzmitry Bahdanau. 2022, 3. Evaluating the text-to-sql capabilities of large language models.

Rozière, Baptiste, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023, 8. Code llama: Open foundation models for code.

Scholak, Torsten, Nathan Schucher, and Dzmitry Bahdanau. 2021, 9. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 9895–9901. doi:10.18653/v1/2021.emnlp-main.779.

Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice

 doi:10.20944/preprints202402.0693.v1

Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian

Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2022, 6. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Sun, Ruoxi, Sercan Ö. Arik, Rajarishi Sinha, Hootan Nakhost, Hanjun Dai, Pengcheng Yin, and Tomas Pfister. 2023, 11. Sqlprompt: In-context text-to-sql with minimal labeled data. pp. 542–550. doi:10.18653/v1/2023.findings-emnlp.39.

Touvron, Hugo, 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, 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, 7. Llama 2: Open foundation and fine-tuned chat models.

Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022, 3. Self-consistency improves chain of thought reasoning in language models.

Wang, Yaqing, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2019, 4. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys 53*. doi:10.1145/3386252.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022, 1. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems 35*.

Xu, Peng, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi, Kit Cheung, Simon J D Prince, and Yanshuai Cao. Optimizing deeper transformers on small datasets.

Yu, Tao, Chien Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020, 9. Grappa: Grammar-augmented pre-training for table semantic parsing. *ICLR 2021 - 9th International Conference on Learning Representations*.

Yu, Tao, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018, 9. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 3911–3921. doi:10.18653/v1/d18-1425.

Zhou, Chunting, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022, 4. Prompt consistency for zero-shot task generalization. *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2613–2626. doi:10.18653/v1/2022.findings-emnlp.192.