
Prompt-Time Symbolic Knowledge Capture with Large Language Models

Tolga Çöplü, Arto Bendiken, Andrii Skomorokhov, Eduard Bateiko,

Stephen Cobb, Joshua J. Bouw

Haltia, Inc.

{tolga, arto, andriy, eduard, steve, joshua}@haltia.ai

Abstract

Augmenting large language models (LLMs) with user-specific knowledge is crucial for real-world applications, such as personal AI assistants. However, LLMs inherently lack mechanisms for prompt-driven knowledge capture. This paper investigates utilizing the existing LLM capabilities to enable prompt-driven knowledge capture, with a particular emphasis on knowledge graphs. We address this challenge by focusing on prompt-to-triple (P2T) generation. We explore three methods: zero-shot prompting, few-shot prompting, and fine-tuning, and then assess their performance via a specialized synthetic dataset. Our code and datasets are publicly available at <https://github.com/HaltiaAI/paper-PTSKC>

1 Introduction

Large language models (LLMs) are transforming human-machine interaction with their adeptness in carrying out conversations. However, despite their proficiency in responding to queries, LLMs cannot be considered good listeners. Their limitation lies in their inability to learn from user-provided information. To utilize any data received from users beyond their context window, LLMs require support from an external system, a significant gap in their interactive capabilities.

Primarily, LLMs capture knowledge during their training phase. This phase, which enables the implicit encoding of knowledge into the model’s parameters, is deemed efficient in terms of data compression. However, the substantial computational power, time, and cost required for training, particularly in the pre-training stage, render it impractical for prompt-driven continuous learning. LLMs are unable to capture knowledge obtained from users or through external integrations/plugins, which presents significant challenges for many AI applications. For instance, the ability of AI assistants to capture and utilize personal information in future interactions is crucial. This limitation is currently being addressed through various Retrieval-Augmented Generation (RAG) approaches. Within this realm, knowledge graphs are distinguished by their clear structures, symbolic representations, and capacity for factual reasoning, making their integration with LLMs a vibrant area of ongoing research [1, 2].

In this paper, we focus on the building blocks of prompt-driven symbolic knowledge capture. We investigate the extraction of prompts in subject-predicate-object triples¹ for a predefined context (relation) through in-context learning and fine-tuning approaches. Utilizing a specially designed dataset, we aim to assess the efficacy of these methods, highlighting their strong points and identifying areas for enhancement.

¹<https://www.w3.org/TR/rdf12-concepts/>

The structure of this paper is as follows: Section 2 introduces the proposed in-context learning and fine-tuning approaches by providing examples. Section 3 describes the experimental setup by presenting the development framework, the language model selection, and the dataset creation process. Section 4 outlines our test results and their interpretations. Finally, Section 5 concludes the paper and suggests future directions.

2 Prompt-to-triple generation

Triples, composed of (*'subject'*, *'predicate'*, *'object'*), are considered a universal data model thanks to their inherent simplicity and versatility. This format reflects the fundamental structure of human language and cognition, capturing the essence of any asserted statement or fact. Each triple represents a distinct atom of knowledge, with the subject and object identifying entities and the predicate describing their relationship. In our study, we have chosen triples as our data model for these characteristics. Furthermore, for ease of presentation, we employ an informal free-form triple format, which allows for greater flexibility in our discussions and examples.

Generating triples based on a predefined context from user prompts can be viewed as a specific aspect of the broader text-to-graph (T2G) generation problem [3, 4]. This perspective led us to define our research problem as prompt-to-triple (P2T) generation. P2T generation entails extracting 'subject' and 'object' terms from user prompts that correspond with a 'predicate' drawn from a restricted vocabulary. This vocabulary consists of predefined, user-specific relations such as birthdays, anniversaries, locations, and events. A key aspect is ensuring that the 'predicate' term of the generated triple accurately reflects the relevant predefined relation. For example, from the prompt 'I was born in 1979', our goal is to generate the triple (*'I'*, *'birthday'*, *'1979'*), aligning with the 'birthday' relation.

In our research, we began by pinpointing some relevant predefined relations (aka a restricted vocabulary for the 'predicate' term). Following on this, we developed the requisite training and test datasets essential for addressing the problem. Building on this groundwork, we have formulated the following three methodologies to effectively tackle the P2T generation challenge.

2.1 P2T zero-shot prompting

Zero-shot prompting is an in-context learning technique that enables LLMs to apply their inherent knowledge to complete tasks without additional training. This approach is relevant in the T2G generation domain, especially in the multi-turn question answering form, as noted in [5]. However, the P2T generation task requires tailored zero-shot prompts due to specific predefined relations. Our approach has evolved through a series of iterative developments and tests. The critical aspects of zero-shot prompting, as explored in our research, include:

- For the prompt context to match the predefined relations, the set of predefined relations must be present in the system prompt. This leads to scalability issues, as both the prompt size and processing time vary with the size of the relation set.
- Due to pre-training biases, LLMs often default to assigning the sentence's verb to the 'predicate' term. This can result in incorrect triples when the verb doesn't match the predefined relation, despite explicit instructions for relation matching. To improve accuracy, we introduced an extra term for relation matching, which is then incorporated into the 'predicate' term via post-processing. Relevant cases are presented in Figure.1. Case 1 demonstrates that, in the absence of explicit instructions, the LLM generates the 'predicate' by utilizing the verb from the provided sentence, aligning with expectations. In Case 2, despite clear instructions for relation matching, the selection of either 'anniversary' or 'birthday' as the predicate is suppressed. Case 3 resolves this by using a new 'relation' term specifically for matching 'birthday' or 'anniversary'.
- LLMs can recognize that a sentence context falls outside the predefined relation set, even without explicit instruction in the zero-shot prompt. The LLM even adds a justification in the response. An instance of this scenario is depicted in Figure.2.

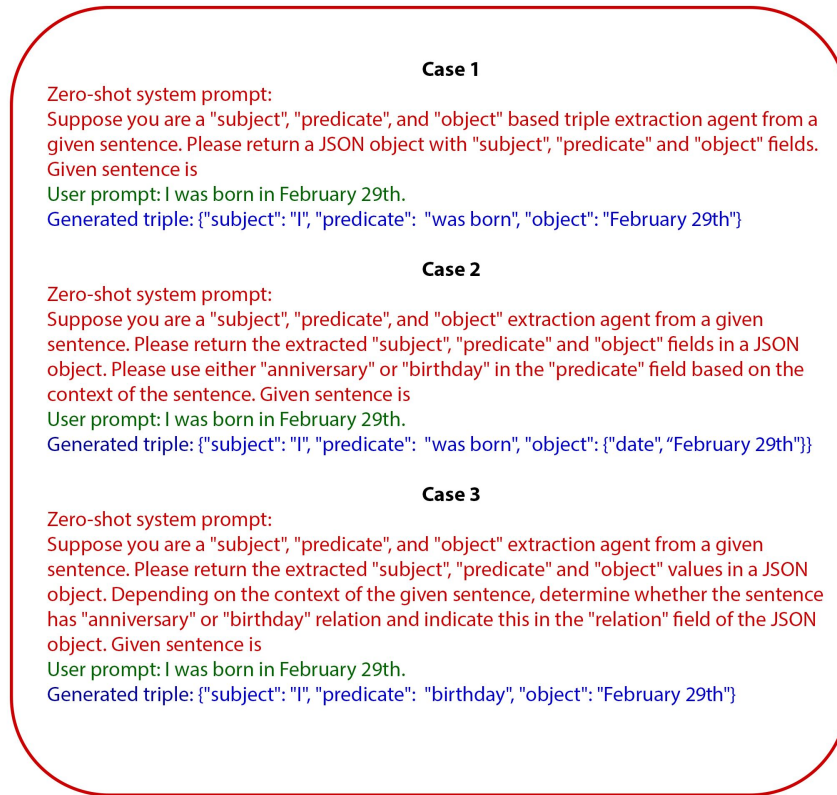


Figure 1: Cases of zero-shot prompting to demonstrate the evaluation.

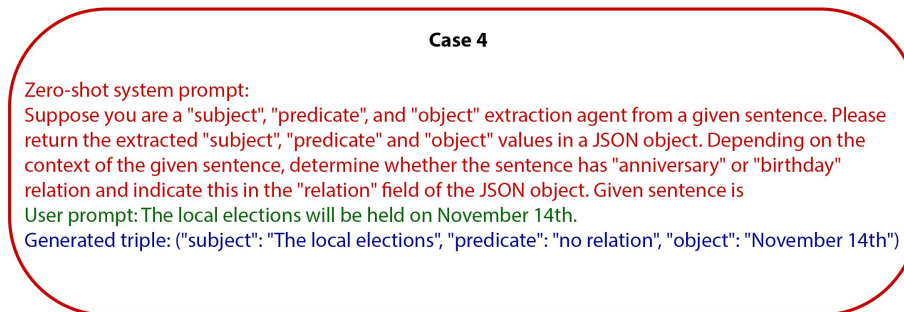


Figure 2: Zero-shot prompting case for out-of-context input

2.2 P2T few-shot prompting

Few-shot prompting, an in-context learning technique, provides an LLM with a small set of examples to enhance its task understanding before generating a response [6]. This method stands apart from zero-shot prompting, which requires no examples, and fine-tuning, which demands extensive training data. Few-shot prompting seeks a balance by offering sufficient context for efficient model operation. Although there are criticisms in the literature [7, 8] regarding the efficiency of few-shot prompting, in our study, we aimed to evaluate this method using our own dataset. Mirroring the approach used in zero-shot prompting, we adopted an iterative development and testing cycle for few-shot prompting. The significant aspects of few-shot prompting are outlined below:

- In a few-shot prompting, providing an example for every predefined relation is necessary. This requirement, similar to zero-shot prompting, leads to scalability challenges.
- The variety of examples has a direct impact on performance. To effectively match the 'birthday' relation, examples must cover various sentence structures, such as "I was born in 1979" and "My birthday is in November". Relevant cases are presented in Figure.3.

Case 5

Few-shot system prompt:
Exclusively use the following examples to generate a single triple.

Example1:
Sentence: Today is my birthday.
{ "subject": "I", "predicate": "birthday", "object": "today" }

Example2:
Sentence: My uncle's birthday falls on November 14th.
{ "subject": "My uncle", "predicate": "birthday", "object": "November 14th" }

Create a single triple from the following sentence, please do not provide any alternatives, and return the answer as JSON object.

User prompt: I was born in February 29th.
Generated triple: { "subject": "I", "predicate": "born", "object": "February 29th" }

Case 6

Few-shot system prompt:
Exclusively use the following examples to generate a single triple.

Example1:
Sentence: Today is my birthday.
{ "subject": "I", "predicate": "birthday", "object": "today" }

Example2:
Sentence: My uncle was born in 1960.
{ "subject": "My uncle", "predicate": "birthday", "object": "1960" }

Create a single triple from the following sentence, please do not provide any alternatives, and return the answer as JSON object.

User prompt: I was born in February 29th.
Generated triple: { "subject": "I", "predicate": "birthday", "object": "February 29th" }

Figure 3: The impact of example diversity is presented with two different few-shot prompts

- When the LLM encounters a sentence context outside the predefined relation set, it relies on its implicit knowledge to perform triple extraction, due to the absence of a corresponding example in the few-shot prompt. Figure.4 represents a case of this particular scenario.

Case 7

Few-shot system prompt:
Exclusively use the following examples to generate a single triple.

Example1:
Sentence: Today is my birthday.
{ "subject": "I", "predicate": "birthday", "object": "today" }

Example2:
Sentence: My uncle was born in 1960.
{ "subject": "My uncle", "predicate": "birthday", "object": "1960" }

Create a single triple from the following sentence, please do not provide any alternatives, and return the answer as JSON object.

User prompt: The local elections will be held on June 1st.
Generated triple: { "subject": "The local elections", "predicate": "will be held on", "object": "June 1st" }

Figure 4: Few-shot prompting example for out of context input

2.3 P2T generation using a fine-tuned LLM

Fine-tuning is a process where a pre-trained LLM is further trained on a specific dataset. This technique adjusts the model's parameters to make it more adept at a particular task, such as P2T generation in our case. The following points highlight the key aspects of P2T fine-tuning:

- The training dataset is critical to the success of the fine-tuning process. It provides a targeted environment with specific examples for relation matching and extracting subjects and objects, requiring a diverse array of examples to address all potential issues. Details regarding the dataset and its generation are presented in Section 3.3.
- For each predefined relation, the training dataset must contain varied examples. This requirement does not lead to scalability issues as in zero-shot or few-shot prompting, but an enlarged training set might increase the risk of performance degradation in other tasks for the LLM.
- When the LLM encounters a sentence context not present in the predefined relation set, it resorts to its implicit knowledge for triple extraction, similar to the few-shot prompting scenario. Figure.5 presents cases for relations both included and excluded in the training set.

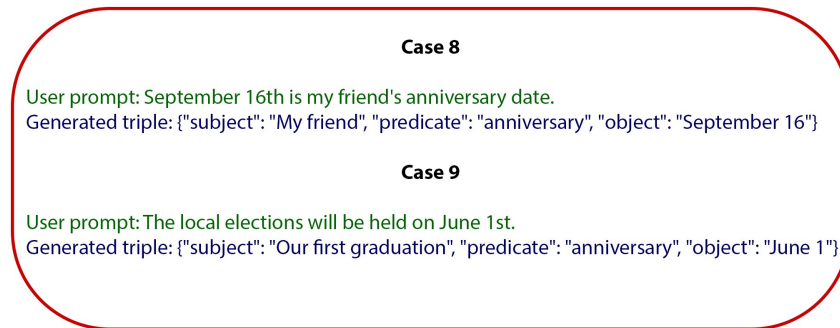


Figure 5: Fine-tuning examples for in-context and out-of-context prompts

3 Experimental setup

This section explores the components of our experimental setup.

3.1 Development framework

The methods suggested in this paper have been implemented using the Apple MLX framework [9]. MLX is a specialized array framework designed for machine learning applications, akin to NumPy, PyTorch, or Jax, with the distinction of being exclusive to Apple silicon.

P2T fine-tuning has been conducted using the parameter-efficient QLoRA approach [10] on our custom dataset, comprising randomly selected, non-overlapping sets of 1,000 training, 200 validation, and 200 test samples. The fundamental QLoRA parameters used are as follows:

- Optimizer: Adam
- Learning rate: 1×10^{-5}
- Number of layers to fine-tune: 16
- Minibatch size: 4
- Iterations: 1,000

3.2 LLM

The methods we have developed here do not have a structural dependency on a particular underlying foundation model. The key factors guiding our LLM selection were its proven effectiveness across diverse domains in community benchmarks and its prevalence in the field. Owing to its performance in the Hugging Face Leadership benchmark [11] and its robust ecosystem, the Mistral-7B-Instruct-v0.2 [12], based on the Llama 2 [13] architecture, was selected for our research. We ran all examples, tests, and benchmarks on a 4-bit quantized version of this model.

3.3 Dataset

In the field of natural language processing (NLP), the creation of robust and diverse datasets is crucial for training and evaluating LLMs, especially for tasks such as knowledge extraction, which involves identifying structured information from unstructured text. Aligning with these needs, we have created a synthetic dataset focused on 'birthday' and 'anniversary' content for P2T generation study, adhering to the detailed stages described below.

Our synthetic dataset creation process consists of three distinct stages:

1. We engaged native speakers to write templates for 86 user prompts and model responses. This initial step ensures the dataset's foundational accuracy and contextual relevance.
2. We leveraged Python's capabilities, particularly a random date generator, to expand this dataset from 86 to 860 prompt-response pairs.
3. The final stage of our dataset development involved using an LLM, specifically Llama-2-7b-chat-hf, to paraphrase each prompt from the previous dataset five times. This resulted in a new dataset of 4300 prompt-response pairs. Paraphrasing is a critical step as it introduces linguistic variations and nuances, thereby enriching the dataset and making it more representative of natural language variability [14, 15]. This approach is supported by recent studies which highlight the importance of paraphrasing in dataset creation for NLP tasks, as it significantly contributes to model generalizability and understanding of diverse linguistic expressions.

4 Performance evaluation

We have evaluated the proposed methods using a non-overlapping test dataset, comprising 128 'birthday' and 72 'anniversary' relations. The outputs generated during the evaluation were compared with manually crafted ground-truths. The comparisons were conducted in two distinct manners: 'relation-based' and 'triple-based'.

- **Relation-based comparison:** This approach focuses solely on the 'predicate' term. In this scenario, equality between the test result and the ground-truth value is reported as a True Positive (TP). 'Predicate' values falling outside the predefined relation set are reported as False Negatives (FN), while those within the set but differing from the ground-truth are reported as False Positives (FP).
- **Triple-based comparison:** This method involves comparing all terms of the generated triple. The comparison of the 'predicate' term follows the same approach as the relation-based method. However, the 'subject' and 'object' values are compared based on a relationship of inclusion rather than direct equality. For example, a generated triple ('I', 'birthday', 'November 14th') compared with the ground truth ('I me', 'birthday', 'November 14') is classified as TP.

The macro precision, recall, and f1-score calculated for both the relation-based and entity-based approaches are presented in Table 1.

As indicated in Table 1, the recall for relation-based evaluation is impeccable across all methods. We believe this outcome is associated with the tests being conducted on only two relations. When considering precision and f1-score, it is observed that both zero-shot prompting and fine-tuning methods emerge as prominent. We assess that the clear guidance provided by the zero-shot prompt, instructing the LLM to select one of the predefined relations, plays a significant role in its superior

performance compared to few-shot prompting. The fact that the fine-tuning method yields significantly good results clearly demonstrates its success in learning straightforward tasks.

Upon examining the triple segment in Table 1, we observe that despite differing precision and recall performance, zero-shot prompting and few-shot prompting methods exhibit similar f1-scores. In this segment, fine-tuning has demonstrated superior performance compared to other methods. These outcomes motivate us to focus on fine-tuning and conduct more comprehensive studies on it.

Table 1: Relation and triple generation performance based on macro precision, recall and f1-score.

	Relation			Triple		
	Precision	Recall	F1-score	Precision	Recall	F1-score
zero-shot prompting	0.815	1.0	0.8981	0.6636	0.4479	0.5348
few-shot prompting	0.49	1.0	0.6577	0.3855	0.6531	0.4848
fine-tuning	1.0	1.0	1.0	1.0	0.96	0.9796

5 Conclusion

In this paper, we initially discussed prompt-driven symbolic knowledge capture and its significance in the LLM domain. We then projected the prompt-driven symbolic knowledge capture problem into prompt-to-triple (P2T) generation, which involves generating triples based on predefined relations from user prompts. To address P2T, we developed new approaches using fundamental LLM techniques, including in-context learning and fine-tuning. We concluded our work with performance evaluations of these proposed methods.

Our findings indicate that fine-tuning is particularly sensitive in addressing P2T. In our future work, we aim to refine the fine-tuning approach and comprehensively examine its impact on the overall performance of the model across various scenarios.

Please see the corresponding GitHub repository at <https://github.com/HaltiaAI/paper-PTSKC>

References

- [1] Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. Large Language Models and Knowledge Graphs: Opportunities and Challenges, August 2023. URL <http://arxiv.org/abs/2308.06374>. arXiv:2308.06374 [cs].
- [2] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying Large Language Models and Knowledge Graphs: A Roadmap, June 2023. URL <http://arxiv.org/abs/2306.08302>. arXiv:2306.08302 [cs].
- [3] Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. CycleGT: Unsupervised Graph-to-Text and Text-to-Graph Generation via Cycle Training, December 2020. URL <http://arxiv.org/abs/2006.04702>. arXiv:2006.04702 [cs].
- [4] Yi Xu, Luoyi Fu, Zhouhan Lin, Jiexing Qi, and Xinbing Wang. INFINITY: A Simple Yet Effective Unsupervised Framework for Graph-Text Mutual Conversion, September 2022. URL <http://arxiv.org/abs/2209.10754>. arXiv:2209.10754 [cs].
- [5] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-Relation Extraction as Multi-Turn Question Answering, September 2019. URL <http://arxiv.org/abs/1905.05529>. arXiv:1905.05529 [cs].
- [6] Tom B. Brown, 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. Language Models are Few-Shot Learners, July 2020. URL <http://arxiv.org/abs/2005.14165>. arXiv:2005.14165 [cs].

- [7] Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!, October 2023. URL <http://arxiv.org/abs/2303.08559>. arXiv:2303.08559 [cs].
- [8] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities, May 2023. URL <http://arxiv.org/abs/2305.13168>. arXiv:2305.13168 [cs].
- [9] Awni Hannun, Jagrit Digani, Angelos Katharopoulos, and Ronan Collobert. MLX: Efficient and flexible machine learning on apple silicon. <https://github.com/ml-explore>, 2023.
- [10] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs, May 2023. URL <http://arxiv.org/abs/2305.14314>. arXiv:2305.14314 [cs].
- [11] Edward Beeching, Clementine Fourier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.
- [12] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. Mistral 7B, October 2023. URL <http://arxiv.org/abs/2310.06825>. arXiv:2310.06825 [cs].
- [13] 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. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [14] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A Survey of Data Augmentation Approaches for NLP, December 2021. URL <http://arxiv.org/abs/2105.03075>. arXiv:2105.03075 [cs].
- [15] Bohan Li, Yutai Hou, and Wanxiang Che. Data Augmentation Approaches in Natural Language Processing: A Survey. *AI Open*, 3:71–90, 2022. ISSN 26666510. doi: 10.1016/j.aiopen.2022.03.001. URL <http://arxiv.org/abs/2110.01852>. arXiv:2110.01852 [cs].