End-to-end Dependency Parsing via Auto-regressive Large Language Models

Claudiu Daniel Hromei[1], Danilo Croce[1] and Roberto Basili[1]

[1]University of Rome Tor Vergata, Italy

Abstract

This paper presents a straightforward application of Large Language Models (LLMs) for Dependency Parsing. The parsing process is approached as a sequence-to-sequence task, where a language model takes a sentence as input and generates a bracketed form, allowing for the deterministic derivation of the dependency graph. The experimental evaluation explores the feasibility of utilizing LLMs for this purpose, while also assessing the process's sustainability with modest parameter sizes (training on a single GPU with limited resources) and investigating the impact of incorporating multilingual data during training. The results demonstrate that an end-to-end dependency parsing process can indeed be formulated using a task-agnostic architecture.

Keywords

Dependency Parsing, Large Language Models, End-to-End Parsing, Sequence-to-Sequence, Auto-Regressive Models

1. Introduction

Dependency parsing is a crucial component of natural language processing that plays a significant role in capturing the syntactic intricacies within sentences [1]. The primary objective of dependency parsing is to establish dependency relations among words. This allows humans to understand how words are connected and how they depend on one another in the sentence's structure [2]. Such understanding is instrumental in a wide range of applications, including semantic interpretation, machine translation, relation extraction, and various other linguistic tasks.

One notable parsing technique is the shift-reduce method, as exemplified in [3, 4]. This parser processes sentences from left to right, word by word while maintaining a buffer for words that are yet to be fully processed. Other approaches have been proposed, based on machine learning techniques, such as the biaffine neural networks, as in [5]. These networks, based on Bi-LSTMs, have proven effective in capturing complex dependencies between words. Another intriguing parsing approach is UDPipe [6], focused on parsing with the Universal Dependency Framework [7]. UDPipe stands out because it performs dependency parsing and other essential tasks like tokenization, morphological analysis, part-of-speech tagging, and lemmatization for multiple languages. UDPipe performs all these tasks without relying on external data. It employs a Bi-LSTM architecture fed with end-to-end, character-level, pre-trained, and contextualized embeddings. The model was trained on an extensive dataset of over a million sentences across different languages to capture cross-lingual relations effectively. The system was later extended [8] as UDPipe+ by incorporating multilingual BERT [9] in its token representations. However, while these methods have achieved state-of-the-art results in various languages, they are tailored specifically for the structure prediction problem based on ad hoc methods.

More recently, models based on the Transformer architecture [10] have gained popularity for their ability to perform classification, regression, and rewriting tasks. These models operate on a sequence and they output another sequence. For instance, the work in [11] introduced an end-to-end seq2seq method for dependency parsing, where the model directly predicts the relative position of the head for each word in the sentence. It also utilized a beam search decoder with tree constraints and sub-root decomposition to improve the results. Moreover, in [12] the authors have experimented with a multi-task, multilingual version of BERT [9]. This model was pre-trained on 104 languages and could predict not only dependency parsing trees but also lemmas, part-of-speech tags, and more for each word in an input sentence. One notable Transformer-based architecture is the LLaMA [13] foundational models. LLaMA is a large model with billions of parameters that generates output sequences in an auto-regressive manner based on the input and previously generated output tokens. It has been recently applied in [14] to a variety of linguistic tasks by instruction-tuning a monolithic architecture to solve them all.

In this work, we raise a crucial question about the applicability of models like LLaMA for predicting tree-

CLiC-it 2023: 9th Italian Conference on Computational Linguistics, Nov 30 — Dec 02, 2023, Venice, Italy

hromei@ing.uniroma2.it (C. D. Hromei); croce@info.uniroma2.it (D. Croce); basili@info.uniroma2.it (R. Basili)

0009-0000-8204-5023 (C. D. Hromei); 0000-0001-9111-1950 (D. Croce); 0000-0002-1213-0828 (R. Basili)

like structures. Specifically, we seek to explore whether such models can be used to define an end-to-end parsing process without relying on architecture choices that are task-dependent. We envision a system that, given an input sentence, predicts an output sequence in a parenthetical form as in [15]. This output sequence allows for the reconstruction of the dependency tree of the original sentence. The experimental results on three Italian treebanks demonstrate that such an approach is not only feasible but also capable of achieving results comparable to the state-of-the-art while requiring minimal training resources, such as training on a single GPU with modest memory.

In the rest of the paper, Section 2 described the proposed approach, Section 3 presents and discusses the experimental evaluations, while Section 4 derives the conclusions.

2. Dependency Parsing via Auto-regressive Language Model

The architecture of Transformers [10] has revolutionized Natural Language Processing (NLP), achieving increasingly higher results. In fact, the Architecture can be divided into three major families: Encoder-only models like BERT [9], RoBERTa [16], and DeBERTa [17] that are responsible for encoding input sequences and generating meaningful representations (embeddings) using the self-attention mechanism; Encoder-Decoder models, such as T5 [18] and BART [19], able to combine the strengths of both the encoder and decoder components and to maintain the integration of the two aforementioned blocks and typically used in tasks like machine translation, summarization, or question-answering, where complex input understanding and transduction are required; Decoder-only models like GPT [20], GPT3 [21], and LLaMA [13], that generate output sequences in an auto-regressive manner based on the input and previously generated output tokens. Recently, approaches based on Large Language Models (LLMs) have shown state-of-the-art performance in countless scenarios and tasks. LLMs excel at understanding language and following instructions, with ChatGPT[1] being a prime example.

However, training and fine-tuning such models require heavy computational resources, i.e. countless GPUs. Recently, a method for efficient training has been introduced, called Low-Rank Adaptation (LoRA [22]). LoRA involves freezing the weights of the pre-trained model and introducing trainable rank decomposition matrices into each layer of the Transformer architecture. This approach significantly limits the number of trainable parameters for downstream tasks while avoiding additional inference latency. Additionally, [23] introduces Quantized-LoRA, an optimization that further reduces memory usage enough to finetune a 65B parameter model on a single 48GB GPU while preserving full 16-bit finetuning task performance. QLoRA backpropagates gradients through a frozen, 4-bit quantized pre-trained language model into LoRA.

One of the challenges in modeling tasks with LLMs (Large Language Models) is that these models take sequences as input and produce sequences as output. For instance, consider this (Italian) sentence[2]:

"*Tutti gli esseri umani hanno capacità non sfruttate, non utilizzate.*" (1)

The Dependency Graph of this sentence is represented in Figure (1). In this graph, each node represents a word, and the arcs define the syntactic relationships among them. Additionally, each arc is labeled to indicate the type of dependency. A special node labeled ROOT is included to mark the root of the sentence, typically the main verb.

By converting the sentence into an arboreal structure, a Dependency Tree (Figure (2)) can be obtained. This tree illustrates the hierarchical structure of the sentence, with the main verb (*hanno*) serving as the ROOT and all other words depending on it. Non-terminal nodes in the tree represent the labels of the dependencies from the Dependency graph in Figure (1), while terminal nodes represent the words from the original sentence. For example, the NSUBJ arc indicates that the word *esseri* is the subject of the sentence, and the OBJ arc shows that the word *capacità* is the object of the verb. Furthermore, the ADVMOD label indicates that the word *sfruttate* is modified by the word *non*, negating its meaning. Both the Dependency Graph and Tree representations are equivalent, and it has been demonstrated in [15] that the Dependency Tree can be transformed into a linguistic representation, e.g. for computational purposes. The linguistic representation of the sentence corresponds to:

[ROOT [NSUBJ [DET:PREDET [Tutti]] [DET [gli]] [esseri] [AMOD [umani]]] [hanno] [OBJ [capacità] [ACL [ADVMOD [non]] [sfruttate] [CONJ [PUNCT [,]] [ADVMOD [non]] [utilizzate]]]] [PUNCT [.]]] (2)

Finally, it is worth noting that the process is reversible, meaning the DP tree can be constructed from the linguistic representation and vice versa. This ability facilitates

[1] https://openai.com/blog/chatgpt

[2] In English: "*All human beings have untapped, unused capacities.*".

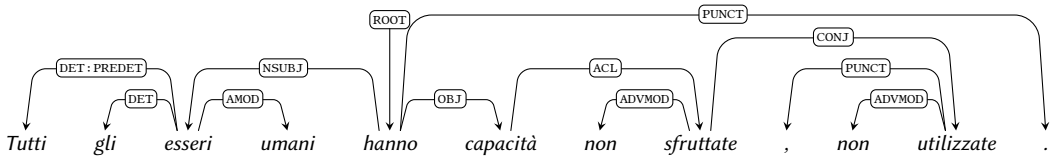

Figure 1: Example of a dependency graph associated to the sentence "*Tutti gli esseri umani hanno capacità non sfruttate, non utilizzate.*"

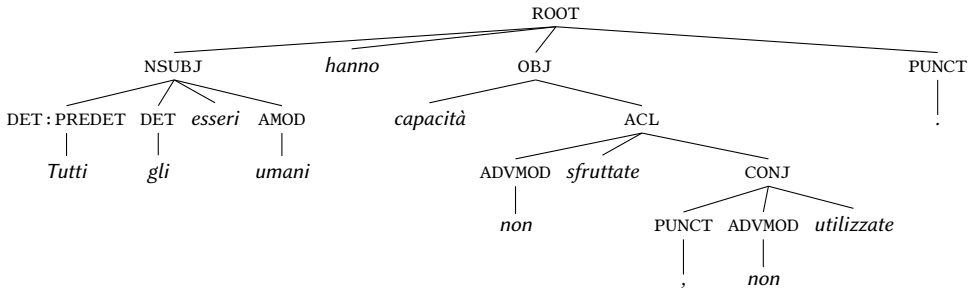

Figure 2: The syntactic parse tree associated with the dependency graph from Figure (1).

various computational tasks involving language modeling and analysis and, more importantly, allows the usage of such LLMs for predicting the Dependency Tree of sentences by training on the linguistic representation

In fact, several studies, such as [21], have highlighted the impressive few-shot learning capabilities of Language Models. These models can generalize information from only a limited number of input examples provided through prompting, producing coherent and accurate output. In this paper, we explore the application of the LLaMA $7B$ and $13B$ foundational models to Italian sentences with the goal of extracting DP Trees in parenthetical form. LLaMA is one of the Large Language Models that operates by taking a sequence of words as input and predicting the next word to generate text recursively. The model is built on the popular Transformers architecture [10], with several key differences. Firstly, to enhance training stability, the RMSNorm function [24] is applied before each layer for normalization. Secondly, the SwiGLU activation function [25] is utilized. Lastly, Rotary Embeddings (RoPE) [26] replace absolute positional embeddings. The combination of these modifications, along with the vast size of parameters and training data (trillions of tokens), makes LLaMA a highly promising model for various natural language processing tasks.

It's important to stress that LLaMA operates as a sequence-to-sequence model, following an autoregressive approach, where text is fed as input, and text is generated as output. This allows the model to capture complex linguistic structures and dependencies in the input sentences and produce corresponding DP Trees in parenthetical form, showing the effectiveness of the approach in parsing Italian sentences. The input/output pairs used for LLaMA consist of sentences (as in Eq. (1)) and the linguistic representation of the DP Trees (as in Eq. (2)). During training and inference, the model is prompted with a simple instruction ("*Parse this sentence.*") to guide it in generating the desired output.

3. Experimental Evaluation

The training of the model utilized PyTorch and the Huggingface library, along with the Peft packages, to implement the Q-LoRA technique. The LLaMA models underwent 3 epochs of training with a learning rate of $3 \cdot 10^{-4}$ and a batch size of 32. To optimize the model's performance, a linear scheduler with warmup was utilized, using a warmup ratio of 0.1. The training process employed Q-LoRA 4-bit to refine the transformer's W_q and W_v modules, as in [23]. The LoRA matrices had a matrix rank R of 8 and a parameter α of 16. The training was performed on a single Tesla T4 GPU with 16GB of memory. This is particularly interesting as we have implied the two smallest available models, i.e. with 7 and 13-billion-parameters, to demonstrate that it can be used even on standard architectures. It doesn't rule out the possibility of evaluating larger models like LLaMA $65B$, but currently, they require such computational power that would limit their applicability in real-world scenarios, due to their extensive training duration and memory requirements.

We used the Universal Dependency Parsing dataset

Table 1
UAS using the Gold Standard tokenization provided.

Model	IT-ISDT	IT-ParTUT	IT-PoSTWITA
UDPipe	93.49%	92.64%	86.03%
UDPipe+	94.97%	95.36%	87.25%
7B_ita_b1	90.52%	93.00%	83.27%
7B_ita_b4	92.04%	93.18%	83.96%
13B_ita_b1	91.51%	93.76%	84.34%
13B_ita_b4	92.38%	94.01%	85.53%
7B_multi_b1	93.06%	94.22%	84.89%
7B_multi_b4	93.30%	94.55%	85.45%

and, to align with [6], we utilized version 2.3 of the dataset and focused on the same subsets of examples in the Italian language, i.e., three Treebanks: IT-ISDT obtained by conversion from ISDT (Italian Stanford Dependency Treebank), IT-ParTUT a conversion of a multilingual parallel treebank and consisting of a variety of text genres, and IT-PoSTWITA a collection of Italian tweets. The neural architecture was trained on the union of these three datasets, comprising $20,270$ training examples, $1,391$ development examples, and $1,309$ test sentences. This initial set of experiments, referred to as ita, involved training and evaluating the neural architecture using examples from the same language.

Objectives. In this experimentation, our primary objective is to address three crucial experimental questions related to natural language processing using the LLaMA model. First and foremost, we seek to determine if this process effectively works and if LLaMA is capable of achieving state-of-the-art performance. Secondly, we aim to explore the potential advantages of employing larger architectures (from $7B$ to $13B$ parameters): traditional large models have been criticized for their considerable computational and environmental costs. By investigating the use of bigger architectures in the LLaMA model, we strive to determine if advancements in performance can be achieved without compromising sustainability. Furthermore, we want to investigate the significance of multilingual data in enhancing the LLaMA model's performance. We draw inspiration from previous works such as UDPipe [6], which have demonstrated the positive impact of multilingual training on various natural language processing tasks. As the LLaMA model supports multiple languages, we aim to analyze whether incorporating multilingual data leads to improved overall performance on the Dependency Parsing task.

In a second set of experiments (referred to as $multi$), we trained the system by incorporating data from English, French, and Spanish datasets. Specifically, we added training examples from the English-EWT, English-GUM, English-LinES, English-ParTUT, French-GSD, French-ParTUT, French-Sequoia, French-Spoken, French-Old, Spanish-AnCora, and Spanish-GSD datasets to the training material. In this case, while the test data remained unchanged, being in Italian, the training dataset consisted of 101,284 examples. The development dataset was also kept in Italian for comparison purposes. To evaluate the performance of the LLaMA model, we have selected two key metrics: *UAS* (Unlabeled Attachment Score), and *LAS* (Labeled Attachment Score). *UAS* assesses the accuracy of the model's dependency tree structure by verifying if the correct head and dependency arcs are generated. On the other hand, *LAS* provides a more comprehensive evaluation by measuring the accuracy of the dependency labels assigned to each arc in the dependency tree.

Table 2
LAS using the Gold Standard tokenization provided.

Model	IT-ISDT	IT-ParTUT	IT-PoSTWITA
UDPipe	91.50%	90.50%	81.80%
UDPipe+	93.40%	93.40%	83.10%
7B_ita_b1	87.40%	89.50%	77.80%
7B_ita_b4	89.00%	90.00%	78.60%
13B_ita_b1	88.87%	90.61%	79.09%
13B_ita_b4	89.81%	90.87%	80.46%
7B_multi_b1	90.42%	91.19%	79.63%
7B_multi_b4	90.80%	91.61%	80.37%

Results Discussion. The experimental results are reported in Table 1, and 2 for the *UAS* and *LAS* metrics, respectively. Here we compare our approach with UDPipe [6] and the subsequent extension UDPipe+ [8], as these are the state-of-the-art systems for the Italian Treebanks. Notice that our LLaMA-based models fail in 0.5-1% of the times to correctly rewrite the whole sentence, i.e. they sometimes skip a word and do not produce any label, differently from UDPipe which covers 100% of the words in a sentence. Please note that, for the purpose of comparison, we have applied gold-standard tokenization in these initial experiments, as done in [8].

Our models are divided into two categories: *ita*, which is trained exclusively on Italian data, and *multi*, trained using material from other languages. From Tables 1 and 2, it is evident that the $7B$ models fall short of achieving state-of-the-art performance; however, they only slightly lag behind UDPipe. Advancing to the $13B$ models shows a modest performance increase, but considering their larger size, their practicality may be limited. Furthermore, we observe a performance boost when incorporating multilingual data during fine-tuning, as the LLaMA models support multiple languages. By enriching the Italian training set with data from other languages, we effectively leverage valuable relations and structures from diverse linguistic sources.

Moreover, since the dependency parsing process of a sentence is a global property of the entire sentence, we have also investigated more complex decoding processes, such as adopting deterministic beam search [27] during

the decoding process. In a nutshell, beam search involves exploring up to b possible sequences during decoding until the completion of individual generations. This way, we believe that the generated sequence is not penalized by locally optimal choices for the decoding process but rather optimized at the sentence level. This enables us to enhance decoding strategies by adopting a larger beam search size ($b4$[3]) instead of relying solely on greedy search ($b1$). As a result, the adoption of beam search systematically improves performances for both the $7B$ and $13B$ parameter models. The results indicate that the model excels in generating the syntactic structure of sentences, with comparable performance to UDPipe. Remarkably, the neural model remains task-agnostic. Additionally, the study suggests that incorporating data from other languages, if possible, is more beneficial than merely scaling up to larger architectures.

Notice that these evaluations used the Gold Standard tokenization of the sentences available from the Treebanks, both during training and inference. For this reason, we trained and evaluated the same models using the "raw" sentences, without any tokenization and requiring the model to produce the resulting DP tree. For instance, the sentence from Eq (1) will be given to the model without any additional spaces for the punctuation, but the resulting output should remain the same, i.e. the one from Eq (2). The models, thus, are required to learn the tokenization during the training phase. The results are in Tables 3 and 4, where the performance for the UDPipe models is not available as they rely completely on the Gold Standard tokenization. For both *UAS* and *LAS* metrics, there is a loss in performance: every model drops around 1% of accuracy with respect to the GS tokenization, with $7B\_ita\_bs1$ losing almost 2% on the *IT-PoSTWITA* treebank. Intuitively, this drop is due to the fact that the model is required to learn the tokenization and the majority of the errors are because of missing punctuation and so on. This result shows the robustness of the LLaMA model even on "un-tokenized" data.

Table 3
UAS computed on the end-to-end process, where the tokenization is performed by the model.

Model	IT-ISDT	IT-ParTUT	IT-PoSTWITA
7B_ita_b1	90.13%	91.19%	81.36%
7B_ita_b4	91.27%	91.96%	82.18%
13B_ita_b1	92.31%	93.80%	82.90%
13B_ita_b4	92.57%	94.34%	83.59%
7B_multi_b1	92.65%	94.12%	83.31%
7B_multi_b4	93.22%	94.31%	83.99%

[3]We experimented with different values for the beam search parameter 'b', but none of them yielded significant performance improvements except for when 'b' was set to 4.

Table 4
LAS computed on the end-to-end process, where the tokenization is performed by the model.

Model	IT-ISDT	IT-ParTUT	IT-PoSTWITA
7B_ita_b1	87.27%	87.77%	76.26%
7B_ita_b4	88.51%	88.45%	77.13%
13B_ita_b1	89.81%	90.52%	77.90%
13B_ita_b4	90.10%	91.04%	78.79%
7B_multi_b1	89.93%	91.32%	78.04%
7B_multi_b4	90.57%	91.62%	78.78%

4. Conclusions

In this paper, we investigate the application of recent popular LLMs, specifically the LLaMA foundational models, to address the Dependency Parsing problem. Our exploration aimed to answer three key questions: **Can we utilize LLaMA in a sequence-to-sequence scenario to rewrite Dependency Parsing Trees from input sentences?** The answer is affirmative. Although LLaMA did not achieve a new state-of-the-art performance, our results demonstrate that the adopted model and approach are competitive with UDPipe, the current leading model. **Can we scale up LLaMA by increasing the number of parameters while ensuring sustainability?** Our evaluation reveals that almost doubling the model's parameters leads to little or no significant gain in performance. However, we found a notable performance increment by leveraging the beam search technique instead of the greedy search. This aspect could be explored further in the future. **Does the inclusion of multilingual data improve the LLaMA model?** Our findings in this paper support the initial hypothesis that using multilingual data enhances the LLaMA model's performance. Every model trained with multilingual data consistently outperforms those trained solely on Italian data.

As a future work, it would be interesting to exploit data from all available languages and evaluate the model's capabilities across a broader linguistic spectrum. This approach could lead to the development of a Universal Dependency Parsing Model as a unified architecture, which holds significant promise in advancing the field of dependency parsing.

Acknowledgments

We would like to thank the "Istituto di Analisi dei Sistemi ed Informatica - Antonio Ruberti" (IASI) for supporting the experimentations through access to dedicated computing resources. Claudiu Daniel Hromei is a Ph.D. student enrolled in the National Ph.D. in Artificial Intelligence, XXXVII cycle, course on *Health and life sciences*, organized by the Università Campus Bio-Medico di Roma. We acknowledge financial support from the PNRR MUR project PE0000013-FAIR.

References

[1] S. Kübler, R. McDonald, J. Nivre, Dependency Parsing, Springer Cham, 2009. URL: https://doi.org/10.1007/978-3-031-02131-2. doi:10.1007/978-3-031-02131-2.

[2] L. Tesnière, Éléments de syntaxe structurale, Klincksieck, Paris, 1959.

[3] J. Nivre, Algorithms for deterministic incremental dependency parsing, Computational Linguistics 34 (2008) 513–553. URL: https://aclanthology.org/J08-4003. doi:10.1162/coli.07-056-R1-07-027.

[4] D. Chen, C. Manning, A fast and accurate dependency parser using neural networks, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 740–750. URL: https://aclanthology.org/D14-1082. doi:10.3115/v1/D14-1082.

[5] T. Dozat, C. D. Manning, Deep biaffine attention for neural dependency parsing, CoRR abs/1611.01734 (2016). URL: http://arxiv.org/abs/1611.01734. arXiv:1611.01734.

[6] M. Straka, J. Straková, Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe, in: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 88–99. URL: https://aclanthology.org/K17-3009. doi:10.18653/v1/K17-3009.

[7] M.-C. de Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, C. D. Manning, Universal Stanford dependencies: A cross-linguistic typology, in: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland, 2014.

[8] M. Straka, J. Straková, J. Hajic, Evaluating contextualized embeddings on 54 languages in POS tagging, lemmatization and dependency parsing, CoRR abs/1908.07448 (2019). URL: http://arxiv.org/abs/1908.07448. arXiv:1908.07448.

[9] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the NAACL 2019, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[11] Z. Li, J. Cai, S. He, H. Zhao, Seq2seq dependency parsing, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 3203–3214. URL: https://aclanthology.org/C18-1271.

[12] D. Kondratyuk, 75 languages, 1 model: Parsing universal dependencies universally, CoRR abs/1904.02099 (2019). URL: http://arxiv.org/abs/1904.02099. arXiv:1904.02099.

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.

[14] C. D. Hromei, D. Croce, V. Basile, R. Basili, ExtremITA at EVALITA 2023: Multi-task sustainable scaling to large language models at its extreme, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[15] D. Croce, A. Moschitti, R. Basili, Structured lexical similarity via convolution kernels on dependency trees, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011, pp. 1034–1046. URL: https://aclanthology.org/D11-1096.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[17] P. He, X. Liu, J. Gao, W. Chen, Deberta: decoding-enhanced bert with disentangled attention, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67. URL: http://jmlr.org/papers/v21/20-074.html.

[19] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, CoRR abs/1910.13461 (2019). URL: http://arxiv.org/abs/1910.13461. arXiv:1910.13461.

[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training, 2018.

[21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah,

J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[22] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, W. Chen, Lora: Low-rank adaptation of large language models, CoRR abs/2106.09685 (2021). URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[23] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, 2023. arXiv:2305.14314.

[24] B. Zhang, R. Sennrich, Root mean square layer normalization, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

[25] N. Shazeer, GLU variants improve transformer, CoRR abs/2002.05202 (2020). URL: https://arxiv.org/abs/2002.05202. arXiv:2002.05202.

[26] J. Su, Y. Lu, S. Pan, B. Wen, Y. Liu, Roformer: Enhanced transformer with rotary position embedding, CoRR abs/2104.09864 (2021). URL: https://arxiv.org/abs/2104.09864. arXiv:2104.09864.

[27] Y. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, R. Kurzweil, Generating high-quality and informative conversation responses with sequence-to-sequence models, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2210–2219. URL: https://www.aclweb.org/anthology/D17-1235. doi:10.18653/v1/D17-1235.