DOI: 10.25968/opus-3450

Generative AI is not Magic

Christian Wartena
Hochschule Hannover, Fakultät III – Medien, Information und Design, Abt. Information und Kommunikation
Data|H – Institute for Applied Data Science Hannover
christian.wartena@hs-hannover.de

***Abstract*—Although the capabilities of large language models are astonishing and exceed the expectations most researchers had about the potential of language modeling, these models are not complete black boxes, and the principles of how these systems work are not too difficult to understand. We show what theories, methods and skills can contribute to a basic understanding of large language models and how these can fit into a curriculum for students with and without a strong background in mathematics and computer science.**

***Index Terms*—Large Language Models, Linguistics, Teaching**

I. Large language models are language models

In recent years, much progress has been made in the field of generative artificial intelligence, and anyone with an internet connection can now try out different tools and see what they can do. In particular, the ability of Large Language Models (LLMs), the main generative AI tool for dealing with language, understanding and generating text, has impressed many people. In what follows, we will focus on LLMs as the most impressive tool available to the public.

To most people LLMs are just black boxes that work some magic to answer any question. Although LLMs are indeed black boxes in the sense that we cannot follow all the steps they take to produce a given output, and we cannot predict the output they will produce by doing a pencil and paper exercise, their architecture and design principles are known and not extremely difficult to understand. In this paper we propose a number of methods and theories that could help to gain a basic understanding of LLMs. Most important for understanding LLMs is to realize that LLMs are not a further development of traditional expert systems using formal knowledge bases and inference rules (see e.g. [1]), but systems designed to model language and linguistic structures. Thus, some basic understanding of the main principles of language and linguistic structures might be a good starting point on a journey to explore generative artificial intelligence. In particular, a good understanding of distributional semantics and statistical language modeling might help to understand LLMs. In addition, students should of course be familiar with artificial neural networks (ANNs) and have a solid knowledge of the principles of machine learning in general.

Of course, it is possible to write excellent prompts (i.e. questions or instructions) without having any idea what a system does to generate a response. In fact, understanding the system may not help you write good prompts at all. However, it might help to decide whether prompting is an appropriate method for a given problem at all, whether one shot or few shot learning might improve the results, or whether training a simple classification model using the raw output of the LLM or even fine-tuning might be a better choice. Training a model on top of an LLM (i.e., training a model using the raw output of the LLM, with or without fine-tuning) does not require a lot of programming skills, since many models and software libraries are freely available. However, using these libraries requires an understanding of their functionality and basic principles of their inner workings.

II. Understanding LLMs

LLMs are basically transformers trained on huge amounts of text. In the now famous picture of the transformer model, as introduced in [2], the model looks quite impressive. Instead of trying to explain this architecture directly, we propose to first explore some of the problems and ideas that led to it. Most of these ideas can be taught at the undergraduate level to students with a basic background in mathematics and computer science.

A. Linguistics

LLMs are designed to model language, not to capture knowledge, and can therefore be better understood if we understand the challenges of describing natural languages. We believe that students should have an overview of linguistic phenomena ranging from word distributions to phonology, morphology, syntax and semantics. For example, byte-pair encoding, the method used by many LLMs as a preprocessing step to encode a sentence, cannot be properly understood unless students are familiar with Zipf's law of word distribution and basic principles of morphological analysis. Similarly, the need for the attention mechanism can be easily explained if students know what types of structures and dependencies can be found in sentences and texts. It is also important to have an overview of different types of language and to understand that different types of language may require different solutions.

B. Language Models

Language models are models that give a probability for the next word that can appear given the previous words. Statistical language models have been developed since the 1980s [3] with a large number of applications. Teaching details of the models might not be relevant nowadays, but studying these models shows what the fundamental problems for next word prediction are and what types of information can be used to predict the next word. Furthermore, these models appealingly illustrate that it is possible to generate naturally sounding texts

Published in: H. Homann, C. Rohbani, and J. C. Will, Eds., KI-Forum 2024: AI 4 Students – AI in Research – AI Showroom. Hannover: HsH Applied Academics, 2024. Pp. 10–13.

with relatively simple methods and small amounts of training material.

C. Distributional Semantics

One of the key concepts for processing language with neural networks is that of word embeddings. Word embeddings have been studied since the early 90's (see e.g. [4] or [5] for an overview) and became popular since the efficient neural implementations by Mikolov et al. [6]. Though transformer models are much more complicated than the simple count based distributional models and the skip-gram and continuous bag of word models from [6], both the concepts of word embeddings and the training methods from [6] are essential for the understanding of the most recent language models. Naive but working implementations of the models can be built easily from scratch (see e.g. [7]).

D. Artificial Neural Networks

Of course, students should understand the basic principles of Artificial Neural Networks (ANNs). The hard part of understanding ANNs is obviously related to loss functions and the calculation of gradients. However, the basic principles of an ANN can be explained without the mathematical details of back-propagation. In addition, using popular libraries such as Keras or PyTorch, students can easily gain some experience in building and using ANNs.

Once students have mastered the concepts of embeddings and neural networks, LSTMs for sequence-to-sequence learning can be introduced and the encoder-decoder architecture follows naturally. Observing the shortcomings of these models at least the idea of cross-attention should be understandable and brings us close to self-attention and the transformer models.

E. Machine Learning

Both for understanding LLMs and for using LLMs effectively students should have a solid understanding of machine learning. The astonishing results we all know from popular online models are not only based on the trained transformer models but are the result of instruction tuning, a special case of fine-tuning these models. This in fact is a case of traditional supervised learning using an instruction dataset [8], [9].

Though models publicly available through a user interface are easy to use for ad hoc tasks, many tasks can be solved most efficiently by locally deploying a (parameter reduced) LLM or by fine-tuning an LLM. Given the availability of LLMs on the platform Hugging Face (https://huggingface.co/) this can be done quite easily but requires a good knowledge of the methodology and principles of machine learning. Especially, students should have a good knowledge of evaluation methods and evaluation strategies, since developing models without proper evaluation never makes any sense.

III. Using LLMs

Given the popularity of LLMs a basic understanding of LLMs should be obligatory for all students of computer or information science programs. Building models on top of pre-trained models and fine-tuning existing models is a useful exercise for interested students as well.

A. Prompt Writing

Although we argue throughout this paper that we should not be too fixated on prompting, it is also a useful method. Not only for small ad hoc tasks, but in many cases, using an instructional model with prompting can yield excellent results. In other cases, testing how well a task can be done with prompting can provide useful insights into the problem and the capabilities of the model before we start collecting data and writing code to fine-tune a model.

A side effect of designing prompts is that students learn the importance of being precise and meticulous in formulating the question or instruction. This is a general writing skill not usually emphasized in computer science courses, but one that is extremely important for college graduates.

B. Fine-Tuning

LLMs can be used as a kind of pre-processing step for many Natural Language Processing (NLP) tasks. After running an LLM on a sentence or short text, representations for the whole text and for each individual word can be extracted. These representations can be used in any classifier, regression or clustering algorithm, e.g. for named entity recognition, sentiment analysis, text classification, keyword extraction, topic detection etc., if training data is available. If the "pre-processing" is integrated in an ANN with the application on top, fine-tuning is also easily possible. Since different layers of the LLM may represent different properties of a text, and since there are usually different ways to get a sentence (or text) representation, it is important that students have an idea of the architecture and training strategy of the language model, and the type of information needed for the task.

C. Evaluation

A skill that is extremely important in the age of machine learning is that of designing and carrying out an evaluation. Collecting and preparing data for training and evaluation can be combined ideally in an exercise for a term paper with selecting a model, selecting an algorithm and setting hyper-parameters.

D. When not to use LLMs

LLMs have achieved amazing results in many application areas. Nevertheless, it is not necessarily the case that LLMs outperform traditional approaches in every tasks and in every application scenario. Moreover, in some tasks results that can be achieved with LLMs are only slightly better than those from more traditional approaches but at the cost of high computational resources and dependencies on third parties.

For many tasks involving some form of text classification, simple bag-of-word models can achieve competitive results that should be used at least as a baseline for evaluating LLM-based approaches. Similarly, for tasks involving sequence classification, such as named entity recognition, the use of sequence-optimizing models, linear Hidden Markov Models or Conditional Random Fields are still very useful. In addition, LLMs are still designed for relatively short texts (e.g. at most 512 sub word tokens for BERT and 4096 for Llama-2 and GPT-3) and it is not yet clear how they can be used for longer texts [10]. It might even be useful to teach some rule based methods since rule based methods in some cases are very efficient for simple tasks.

Furthermore, a number of methods that produce good but maybe not the best results anymore, give useful insights in the nature of linguistic phenomena or introduce interesting programming concepts, like Dynamic Programming or Hidden Markov Models, that are important for students to achieve a higher level of programming skills.

IV. LLMs for Information Science and Computer Science Students

In the Bachelor program *Information Management* at the University of Applied Sciences and Arts Hannover (Hochschule Hannover, HsH) generative AI is integrated in a number of courses. On the one hand side we have courses that address ethical and legal aspects of LLMs and many courses address the issue how prompt based AI can be used effectively, e.g. for information search. On the other hand side the topics related to the understanding of LLMs as sketched abovev are covered in a number of lectures. The course *Grundlagen der Computerlinguistik* (Introduction to Computational Linguistics) in the third semester gives a high level overview of phenomena in natural languages, methods to model these phenomena and solve specific tasks and includes a introduction to LLMs as well. This mandatory course provides a high-level overview of all topics without technical details or exercises. In the fifth semester the elective module *Information Retrieval* covers evaluation of retrieval and classification results extensively, both theoretically and with exercises. Finally, in the sixth semester, the elective module *Text and Data Analytics* deals with machine learning, neural networks, text classification and information extraction. In this module, the focus is on hands-on work with various methods and techniques to develop both practical skills and a better understanding of the methods. Many exercises are done using Python, and a term paper is required that involves the design, implementation, and documentation of a typical machine learning experiment. Typical topics for the term paper are hate speech detection, sentiment analysis, text classification, requirements extraction, etc.

The same ideas are implemented in the course *Natural Language Processing* in the International Master's Program in Data Analytics at the University of Hildesheim. Here, we assume that all students are already familiar with machine learning, neural networks and evaluation techniques and we focus on linguistic phenomena and specific methods used in computational linguistics.

V. Conclusion

In this paper we have argued that LLMs are not completely magical black boxes, and that a basic understanding of their architecture and operating principles need not be limited to a few people working in leading large technology companies. The design and capabilities can be well understood by students with a basic computer science background. We argue that to understand LLMs, it is helpful to think of them as *language* models, designed primarily to solve problems of language modeling. We have shown which topics from linguistics and computational linguistics can contribute to the understanding of LLMs, and which knowledge from the field of machine learning is minimally required. We have outlined how students can gain practical experience in working with LLMs. Finally, we have shown how the elements for understanding LLMs can be taught in a bachelor's curriculum in library and information science and in a master's curriculum in data science.

NLP is a rapidly developing area of research. The methods used to solve many problems have changed completely in recent years, and at the same time more and more students without a linguistic background are becoming interested in the field. Thus, the question of how to design an NLP curriculum, and which methods should be taught and which are outdated, has been the subject of some debate (see e.g. [11]). In this paper, we have argued that it is more important for understanding LLMs to have an idea of linguistic phenomena and properties of language than to know the exact details of activation functions, backpropagation, regularization, and so on. Similarly, for a full computational linguistics program, our position is that linguistic phenomena should be the guiding principle, not the techniques that are currently best at modeling them. Language and linguistics are what distinguish NLP from machine learning in general, and a proper understanding of these topics is essential to the successful use and design of NLP applications.

References

[1] J. C. Giarratano and G. Riley, *Expert Systems*, 3rd ed. USA: PWS Publishing Co., 1998.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[3] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[4] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behavior research methods*, vol. 39, pp. 510–526, 2007.

[5] D. Kiela and S. Clark, "A systematic study of semantic vector space model parameters," in *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, A. Allauzen, R. Bernardi, E. Grefenstette, H. Larochelle, C. Manning, and S. W.-t. Yih, Eds. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 21–30. [Online]. Available: https://aclanthology.org/W14-1503

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[7] "Distributionelle semantik," https://textmining.wp.hs-hannover.de/DistributionelleSemantik.html, accessed: 2024-07-12.

[8] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," 2022. [Online]. Available: https://arxiv.org/abs/2109.01652

[9] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 730–27 744. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[10] J. Ainslie, S. Ontanon, C. Alberti, V. Cvicek, Z. Fisher, P. Pham, A. Ravula, S. Sanghai, Q. Wang, and L. Yang, "ETC: Encoding long and structured inputs in transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 268–284. [Online]. Available: https://aclanthology.org/2020.emnlp-main.19

[11] A. Friedrich, S. Grünewald, M. Mieskes, J. Strötgen, and C. Wartena, Eds., *Proceedings of the 1st Workshop on Teaching for NLP*. Ingolstadt, Germany: Association for Computational Lingustics, Sep. 2023. [Online]. Available: https://aclanthology.org/2023.teach4nlp-1.0