Report on Card 4 - Vídeo: Introdução ao LLM (II)

Marcus Vinicius Oliveira Nunes

1. Activity Description

1.1 How Large Language Models Work

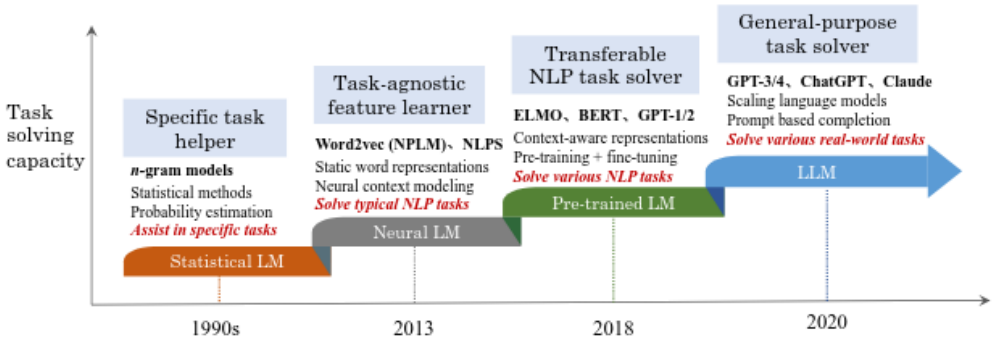
The video was about LLM(Large language models), what they are, how they work and what they are used for. They have this name because they are trained on a big quantity of data(words). I'll take the video Let's build GPT: from scratch, in code, spelled out, for instance, in order to train the transformer more than 1 million tokens were used, and even with this quantity of tokens it can be considered a small one. A larger model can hold around 1 petabyte.

LLM models like the Transformers, which are known to have the best results when it comes to NLP, they work with sentences and in each sentence they learn the relation between words and the context of each one through self-attention. This approach helps the model make right predictions of what words come after other words. Today it's possible to see these generative models everywhere such as GPT, Deep Seek, Gemini etc.

1.2 A Survey of Large Language Models

Although LLMs are a state of the art algorithm today when it comes to NLP, this approach within the NLP field took some years to evolve to what it is today. They are capable of producing human-like text, engaging in dialogues and doing some tasks like summarization, filling holes in texts or even enhancing texts to better writing.

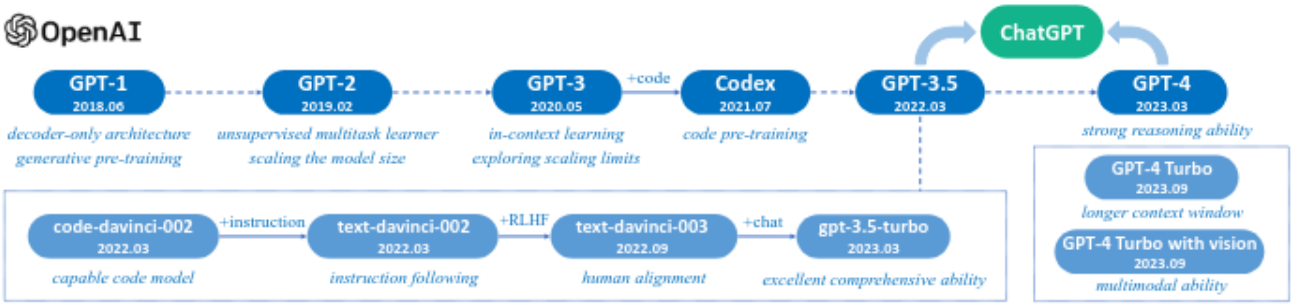
The article's first topics describes the evolution of the models throughout the de decades and describes each approach(models and structures) for the NLP field, the first ones were the **Statistical Models**, which were focused on predicting words sequences for tasks like information retrieval and speech recognition, after that **neural models** came with techniques like word2vec that introduced efficient word representations, it's important mention that it improved performance across NLP tasks. The next state of the art approach were the **pre-trained models** like BERT and ELMo that use large datasets , they use context-aware training. The large language models came after, which are an enhancement of pre-trained models that can work with larger datasets, improving even more the quality of the models.



The evolution of these models throughout the years

The second topic provides information on the much more recent evolution of large language models, it points out some of the things that were already said like that the LLMs use mainly transformers, which are neural networks that excel in the NLP field and can be trained with large datasets. This topic also mentions the evolution of GPT models. Throughout the years the generative pre-trained transformer has been improving its capacity of data, knowledge, optimization for dialogue, reasoning, context tracking etc.

Not only GPT has evolved but other LLMs too, mainly through scaling, RLHF and tool manipulation.



The evolution of GPT throughout the years

The next topic is related to the applications of LLM models, many of these applications have been applied throughout the fastcamp until now and those that are mentioned in the article were also mentioned throughout the NLP fastcamp like tagging, text generation, question answering etc. These are just a few things that it's used for. LLMs can work in recommendation systems, educational establishments like schools to help with writing assistance, personalized learning, enhance texts and much more.

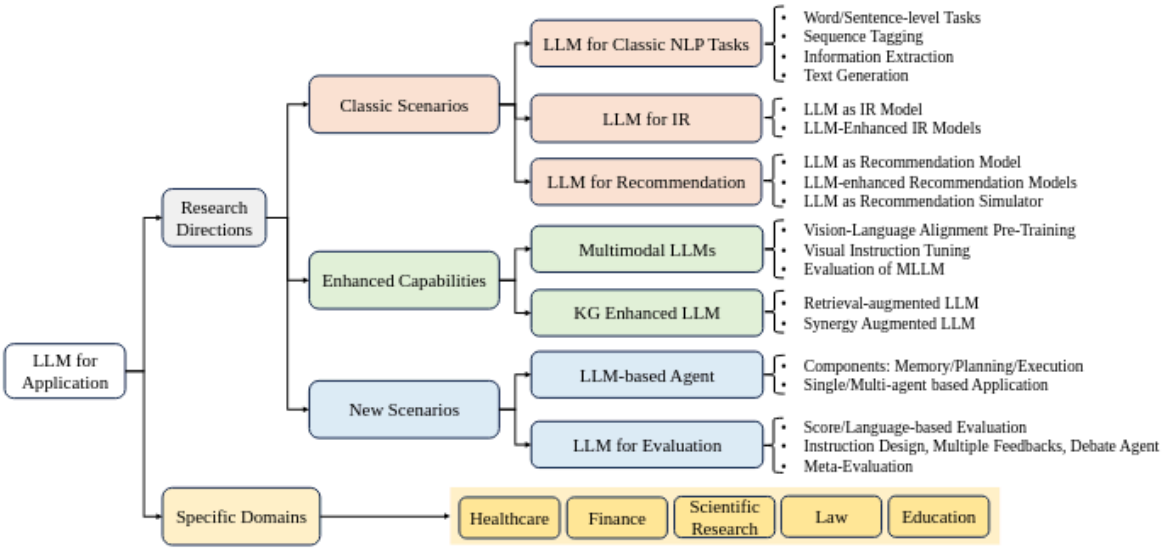


Fig. 18: The applications of LLMs in representative research directions and downstream domains.

Conclusion

LLM models are very powerful and are considered the state of the art of NLP models, they can work with very large datasets and have better results in the NLP field than other AI models, accomplishing this by the self-attention method that the transformers use.

These models went through a lot of changes and enhancements before becoming what they are today such as the enhancement of the quantity of data that the models can use and the quality of the outputs.

References

Let's build GPT: from scratch, in code, spelled out.

https://arxiv.org/pdf/2402.06196