Machine Learning for Textual and Unstructured Data

Lecture 4: Large Language Models

Stephen Hansen
University College London

FUNDACIÓN RAMÓN ARECES

IESE Business School University of Navarra

Center for International Finance

Introduction

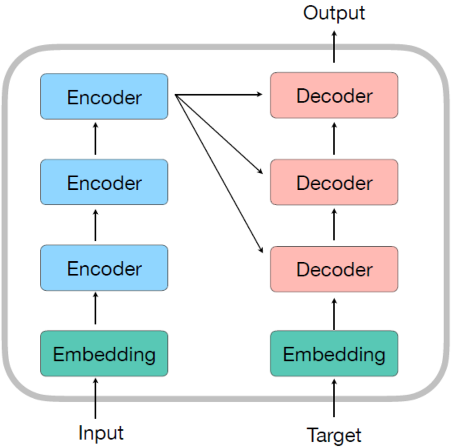
Recall the twin problems from the previous lecture slides: synonymy and polysemy.

Word embeddings help address the problem of synonymy but not that of polysemy: every instance of a word has the same vector representation.

We now move from word embeddings to sequence embeddings.

Doing so is one of the tasks performed by so-called large language models.

Conceptual Diagram of LLM



Revisit Word Prediction

Recall the text from the previous slide:

> *Every morning last summer in Greece, I visited the [MASK] where I would swim, play in the sand, and sunbathe.*

How to build a conditional probability for [MASK] given its context?

Traditional way: RNN, LSTM. But computationally expensive.

Key breakthrough came via attention operation [Vaswani et al., 2017].

Large language models are neural networks that combine attention and feedforward layers to perform language prediction tasks.

Such networks have a Transformer architecture.

Attention

Attention layers take as input a sequence of initial token embeddings and output a sequence of new token embeddings.

Let $(\boldsymbol{\rho}^0_{d,1}, \ldots, \boldsymbol{\rho}^0_{d,N_d})$ be the initial embeddings that make up a document.

The new embedding at each position n is given by

$$\boldsymbol{\rho}^1_{d,n} = \sum_{n'=1}^{N_d} w_{(d,n),n'} \boldsymbol{\rho}^0_{d,n'} \text{ where } \sum_{n'=1}^{N_d} w_{(d,n),n'} = 1.$$

The attention weights allow terms to interact in the formulation of updated embeddings.

Parameterization of Attention Weights

Let $\mathbf{q}_{d,n}$ be a query vector associated $w_{d,n}$.

Let $\mathbf{k}_{d,n}$ be a key vector associated $w_{d,n}$.

Attention weights used to update $\boldsymbol{\rho}^0$ are given by

$$w_{(d,n),n'} = \frac{\exp\left(\mathbf{q}_{d,n}^T \mathbf{k}_{d,n'}\right)}{\sum_{n'=1}^{N_d} \exp\left(\mathbf{q}_{d,n}^T \mathbf{k}_{d,n'}\right)}$$

$\boldsymbol{\rho}^0$ sometimes called a value.

Transformer Architecture

In LLMs the attention operation is preformed repeatedly.

Multi-head attention performs separate attention operations in parallel, and then linearly combines the output to obtain new vector.

The above operation is more precisely called self-attention.

Cross-attention links the encoder and decoder layers by updating one input conditional on the other.

Transformers have multiple attention layers that operate in sequence.

The whole system is trained to perform language prediction tasks.

See [Phuong and Hutter, 2022] for more formal description.

BERT

Encoding Sequences

BERT (Bidirectional Encoder Representations from Transformers) was a breakthrough LLM that vastly outperformed existing methods on benchmark NLP tasks.

Key features:

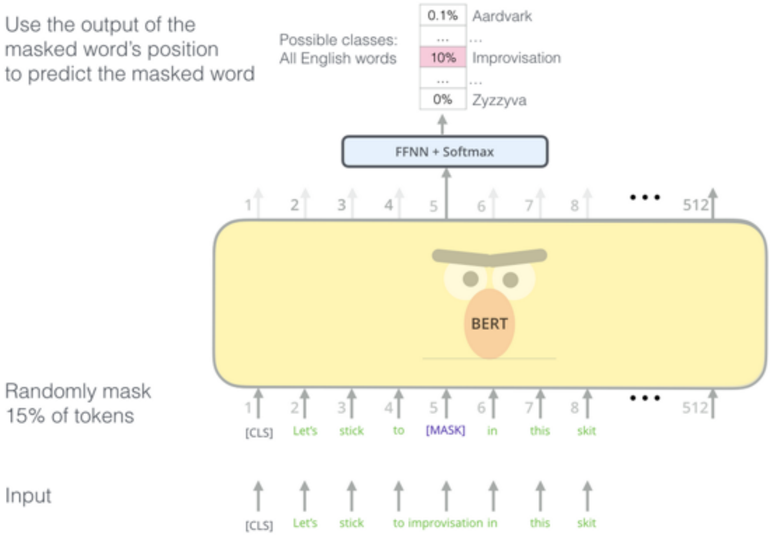
1. Custom tokenization include special tokens [MASK], [CLS], [SEP].
2. Maximum document length is 512.
3. Base model has 110 million parameters and twelve multi-head self-attention layers.
4. Multiple variants: cased/uncased, large, non-English

Training Objective

The neural network parameters are adjusted to minimize a loss that depends on two prediction tasks:

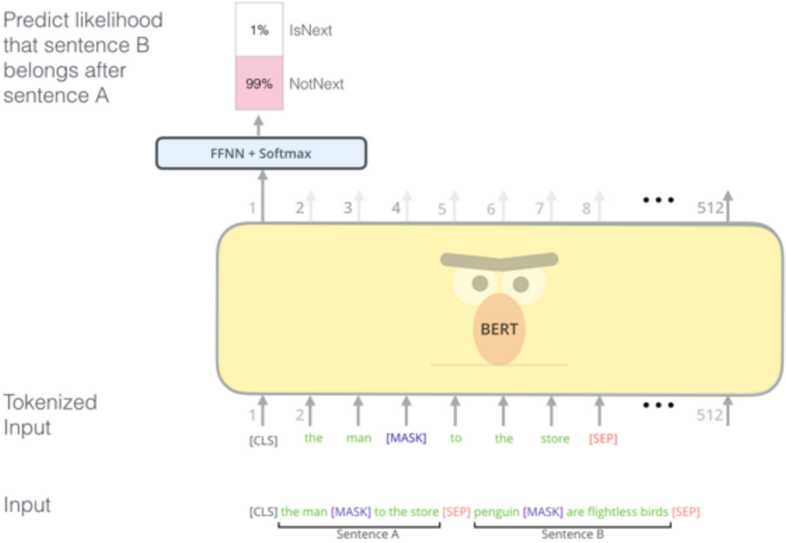
1. Masked-word prediction. Randomly replace 15% of tokens with [MASK]. Form embeddings for [MASK] tokens that successfully predict hidden work.
2. Next-sentence prediction. Documents begin with [CLS] tokens. Form embeddings for [CLS] that successfully predict next segment defined by [SEP].

BERT I



Use the output of the masked word's position to predict the masked word
Possible classes: All English words
0.1% Aardvark
... ...
10% Improvisation
... ...
0% Zyzzyva
FFNN + Softmax
1 2 3 4 5 6 7 8 ... 512
BERT
Randomly mask 15% of tokens
1 2 3 4 5 6 7 8 ... 512
[CLS] Let's stick to [MASK] in this skit
Input
[CLS] Let's stick to improvisation in this skit

BERT II



Predict likelihood that sentence B belongs after sentence A
1%
IsNext
99%
NotNext
FFNN + Softmax
1
2
3
4
5
6
7
8
512
BERT
Tokenized Input
1
2
512
[CLS]
the
man
[MASK]
to
the
store
[SEP]
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Training Data

BERT is trained on a corpus of books and Wikipedia.

Enormous computational resources required, not feasible for most academic teams.

Transfer learning becomes essential.

Even when the base model is not of innate interest, it is the starting point for further training.

Further Pre-Training

Prior to using BERT in downstream applications, it is common to further adjust the embeddings to predict masked words in specific corpora.

Examples from corpus of Lightcast job postings:

As a leading firm in the [MASK] sector, we hire highly skilled software engineers.

As a leading firm in the [MASK] sector, we hire highly skilled petroleum engineers.

Reconstructed Word Probabilities

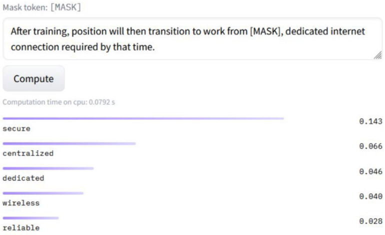
'software engineers' Sentence		'petroleum engineers' Sentence	
Word	Prob.	Word	Prob.
it	0.08	energy	0.279
automotive	0.079	oil	0.27
technology	0.072	petroleum	0.088
healthcare	0.058	mining	0.035
insurance	0.053	defence	0.021
software	0.041	automotive	0.02
engineering	0.031	construction	0.017
public	0.03	gas	0.017
infrastructure	0.028	engineering	0.016
financial	0.028	water	0.012

Table 1: Predictions for Masked Words in Example Sentences

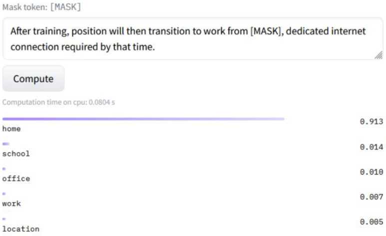
This tables displays masked word prediction probabilities for the two example sentences above. The training corpus for estimating these probabilities is English-language online job postings provided by Lightcast (formerly Emsi Burning Glass). The Transfomer model estimated for the task is DistilBERT (Sanh et al. 2020). See Hansen et al. (2023) for more details.

Does Further Pre-Training Make a Difference?

Out-of-the-box model



Model with additional pre-training



Fine-Tuning

Beginning from baseline BERT, the [CLS] token can be adjusted to predict any label associated with a document.

The NLP community has defined various sequence-level labels relevant for natural language tasks.

Sequence embedding models show outstanding performance at predicting these.

To make such models relevant for economics, we need to define labels with economic content.

NLP Tasks

Task	Example	Dataset	Metric
Grammatical	"This toast is than that one." **= Ungrammatical**	CoLA	Matthews
Sentiment Analysis	"Toy Story 2 was okay." **= .543291 (neutral)**	SST-2	Accuracy
Similarity	a.) A pride of lions surrounded a monkey. b.) Lions encompassed a monkey. **= 4.7 (Very Similar)**	STS-B	Person / Spearman
Paraphrase	A. Last week, Seattle reported 12 new earthquakes. B. Seattle reported another 12 earthquakes yesterday. **= A Paraphrase**	MRPC	Accuracy / F1
Question Similarity	a.) How can I cook noodles over a campfire? b.) How do you make Mac & Cheese? **= Not Similar**	QQP	Accuracy / F1
Contradiction	a.) Glossier products are the best! b.) Glossier products are overpriced. **= Contradiction**	MNLI-mm	Accuracy
Answerable	a.) How does the Dyson Airwrap work? b.) The Airwarp uses the Coanda effect to create a vortex pulling the hair towards the attachments. **= Answerable**	QNLI	Accuracy
Entail	a.) In 2006, Paul David bought a Microprocessing center to create 30,000 jobs in Northern Minnesota. b.) Paul David created 30,000 jobs in MN. **= Entail**	RTE	Accuracy
Ambiguous pronouns	a.) Federico spoke to Marie, breaking her focus. b.) Federico spoke to Marie, breaking Federico's focus. **= Incorrect Referent**	WNLI	Accuracy

Economics Applications

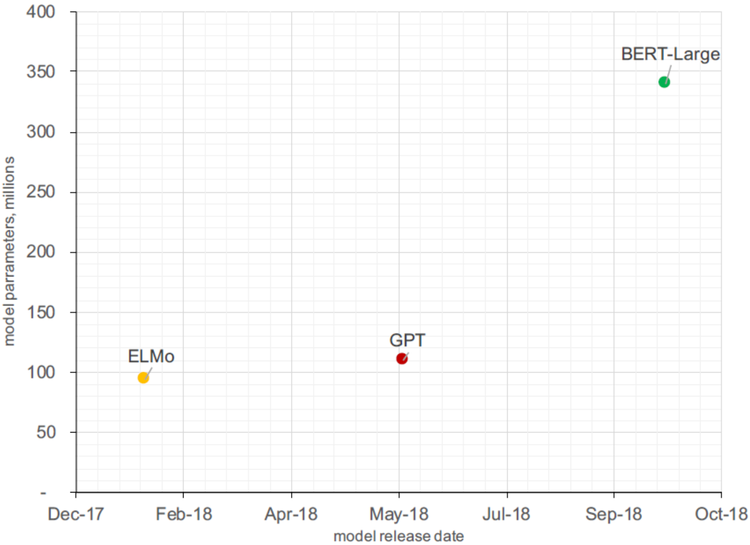
[Bajari et al., 2021] fine-tunes BERT to predict prices of Amazon products from product description text.

[Bana, 2022] predicts posted wages from job posting text.

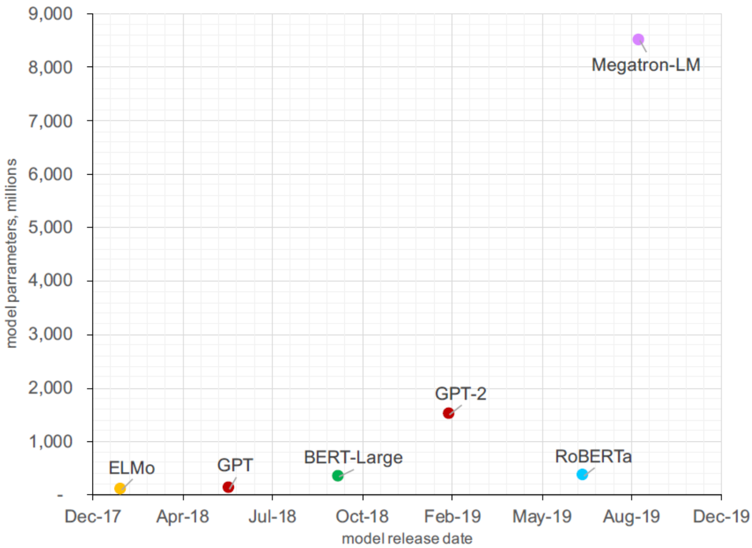
Applications show extremely high out-of-sample R2.

Interpretation remains a challenge.

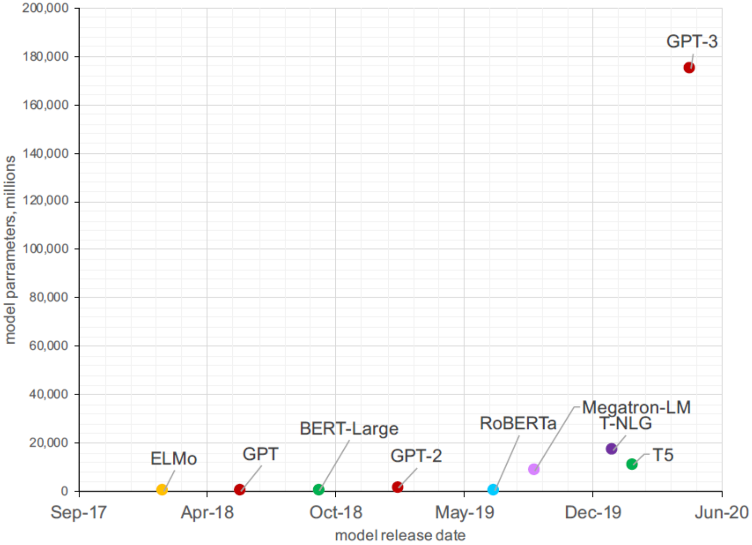
Recent Developments



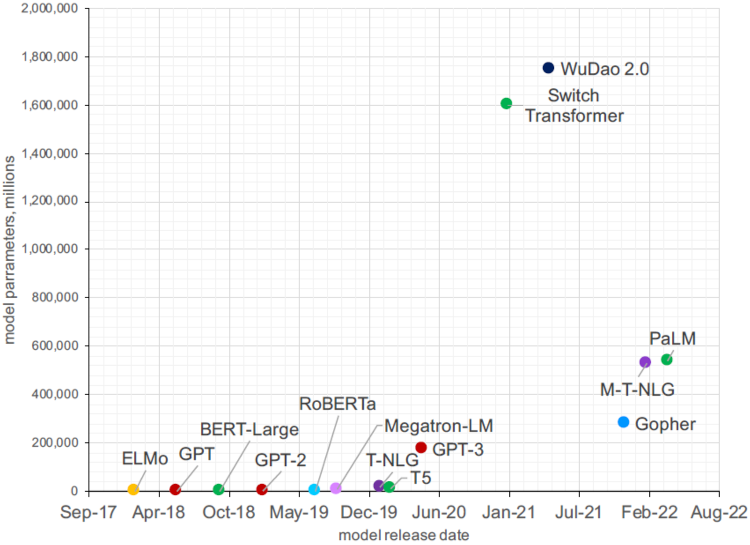
Thanks to Max Ahrens



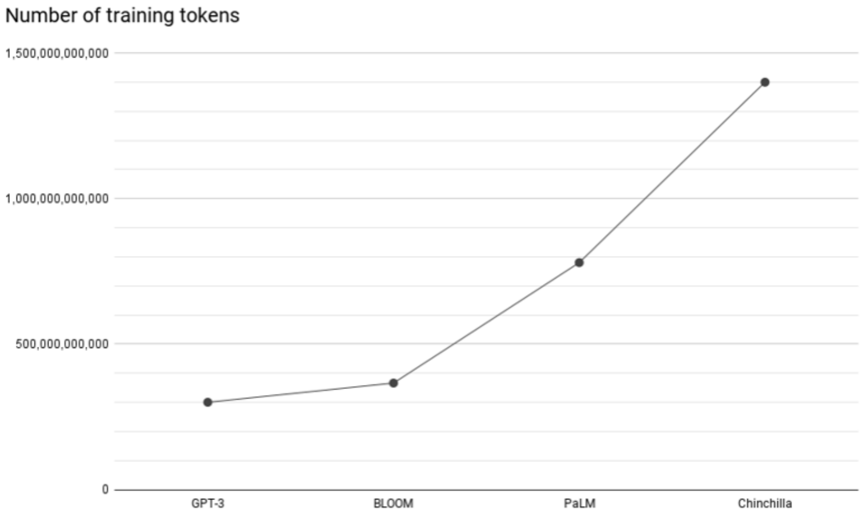
Thanks to Max Ahrens



Thanks to Max Ahrens



Thanks to Max Ahrens



Thanks to Magnus Sahlgren

Next-Token Prediction

GPT and related models have a different prediction task than BERT: predict the next element in a sequence of data.

This allows them to generate text in response to an input.

Related to the decoder block of the full Transformer model of [Vaswani et al., 2017].

Basic architectural elements remain the same: multi-head attention + feedforward layers.

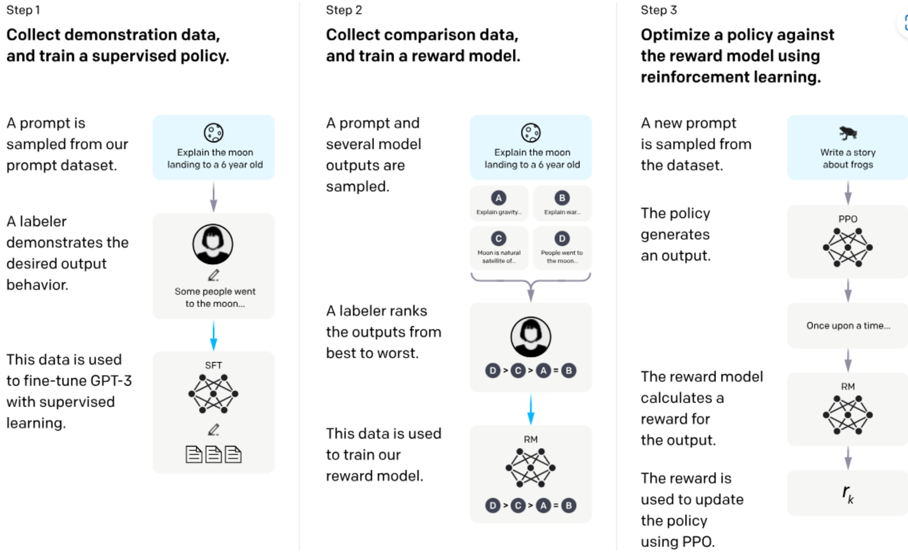
Zero-Shot Learning

Starting with GPT-3, LLMs began to feature a capacity for zero-shot learning for certain NLP tasks.

Such LLMs can be productively used “out-of-the-box” rather than forming the base model for further pre-training.

ChatGPT added to the basic Transformer architecture a reinforcement-learning-based objective guided by human input.

InstructGPT [Ouyang et al., 2022]



Annotator Demographics

Table 12: Labeler demographic data

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%

What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%

How to Fine Tune an LLM?

Updating a model with hundreds of billions of parameters for fine tuning is extremely costly.

[Hu et al., 2022] propose an approach called LoRA (Low-Rank Adaptation) for reducing computational complexity.

Idea: add low-rank, conformable matrices into the neural network. Freeze the pre-trained model parameters and only update the injected matrices.

Allows finetuning of large models with only modest hardware requirements.

LLMs are not Perfect

Example ChatGPT prompt:

> *Jack and Jill are sitting side by side. The person next to Jack is angry. The person next to Jill is happy. Who is happy, Jack or Jill?*

See [Altabaa et al., 2023] for Transformer model of Relational Reasoning.

Open Source Models

The latest OpenAI models remain rather expensive to use at scale.

Increasing tendency to hide training data and model architectures.

LlaMa is an open-source LLM whose weights are widely available.

Alpaca finetunes LlaMa using the output of ChatGPT as a training objective.

Conclusion

The pace of development of LLMs is rapid with new models emerging every month.

They are not magical: attention layers + FFNN + finetuning against human input.

They are clearly useful although exact impact on research (and broader economy) is somewhat unclear.

References I

Altabaa, A., Webb, T., Cohen, J., and Lafferty, J. (2023).
Abstractors: Transformer Modules for Symbolic Message Passing and Relational Reasoning.

Bajari, P., Cen, Z., Chernozhukov, V., Manukonda, M., Wang, J., Huerta, R., Li, J., Leng, L., Monokroussos, G., Vijaykunar, S., and Wan, S. (2021).
Hedonic prices and quality adjusted price indices powered by AI.
Working Paper CWP04/21, Cemmap.

Bana, S. H. (2022).
Work2vec: Using Language Models to Understand Wage Premia.
Unpublished Manuscript.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022).
LoRA: Low-rank adaptation of large language models.
In ICLR 2022.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022).
Training language models to follow instructions with human feedback.

References II

Phuong, M. and Hutter, M. (2022).
Formal Algorithms for Transformers.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017).
Attention is All you Need.
In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.