Pre-Trained Language Models

Natural Language Processing Seminar

Mohammad ali Ali panah - January 2025

Abstract

Pre-trained language models have achieved striking success in natural language processing (NLP), leading to a paradigm shift from supervised learning to pre-training followed by fine-tuning. We first give a brief introduction of pre-trained models, followed by characteristic methods and frameworks. We then introduce and analyze the impact and challenges of pre-trained models and their downstream applications. Finally, we briefly conclude and address future research directions in this field.

Table of contents

Terminology

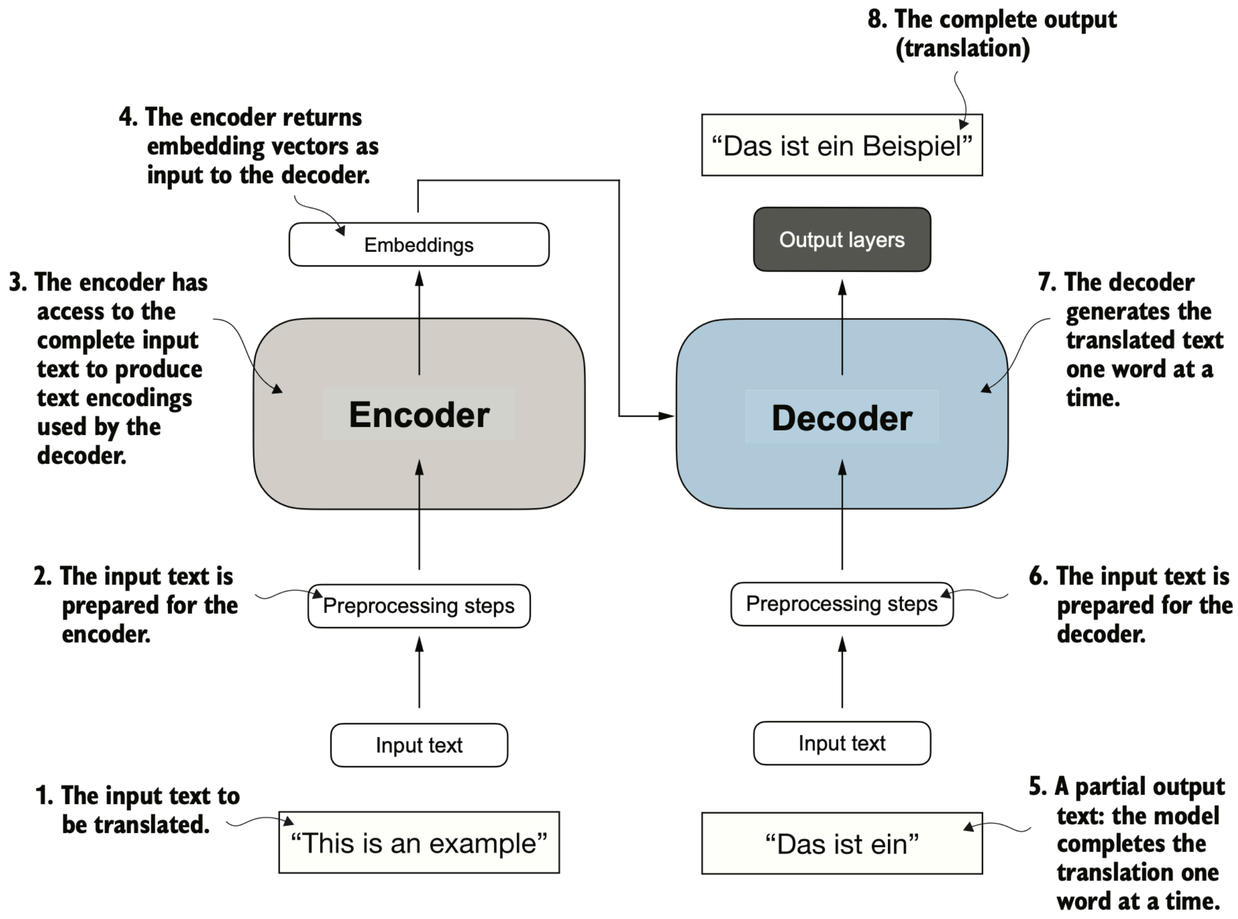
Terminology and Ubiquitous Language

- **Language model**: a system that assigns probabilities to sequences of words and predicts the next word based on previous words. -Jurafsky
- **Fine-tune**: a process where the model is specifically trained on a narrower dataset that is more specific to particular tasks or domains.
- **PTMs**: A pre-trained language model is a machine learning model that has been trained on a large dataset of text before being fine-tuned for a specific task.
- **Attention mechanism**: allows the model to weigh the importance of different words or tokens in a sequence relative to each other. This mechanism enables the model to capture long-range dependencies and contextual relationships within the input data.
- **Zero-shot learning**: the ability to generalize to completely unseen tasks without any prior specific examples.
- **few-shot learning**: learning from a minimal number of examples the user provides as input

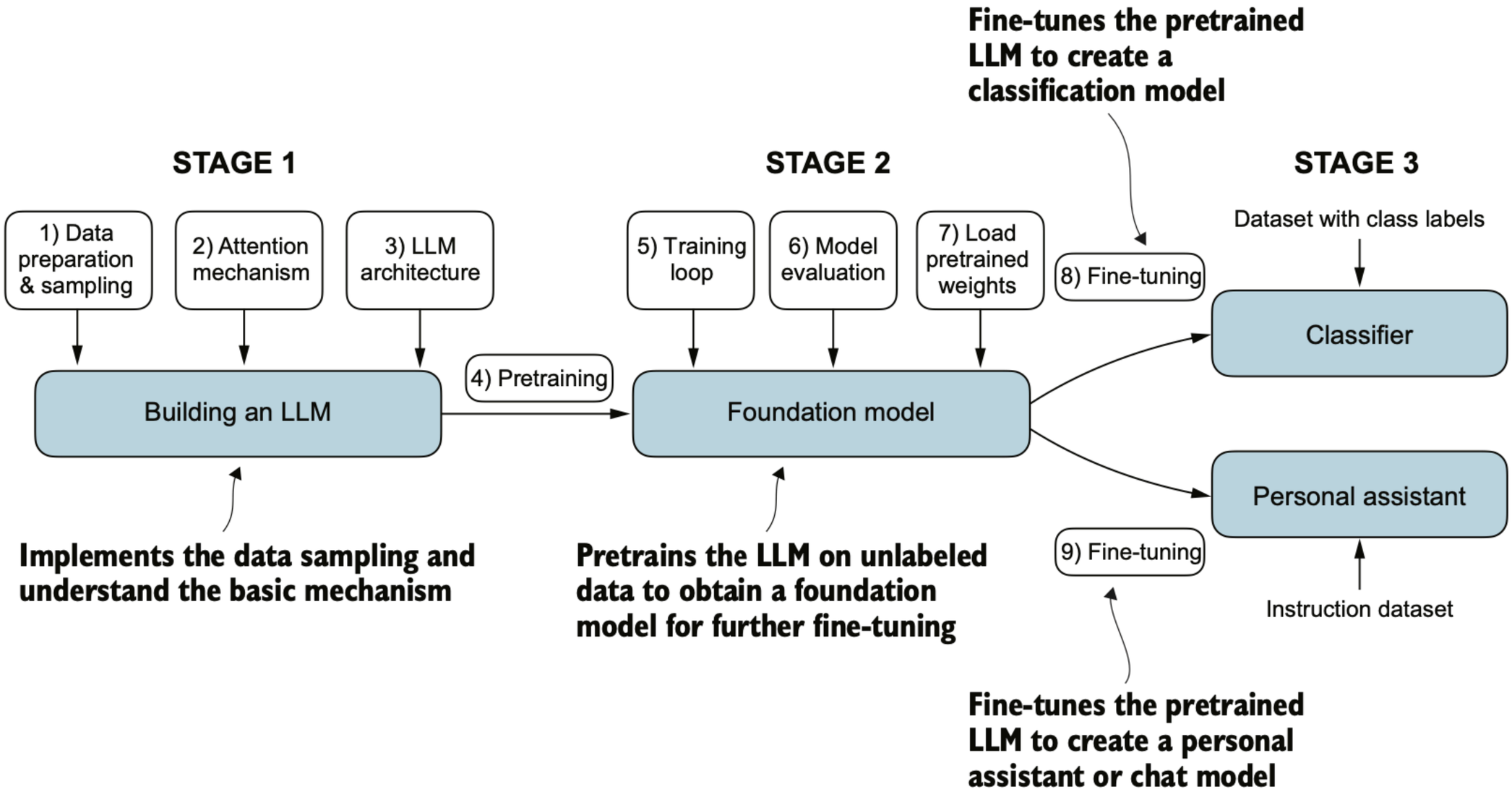
Transformer architecture

The transformer architecture consists of two submodules: an encoder and a decoder.

- **Encoder**: processes the input text and encodes it into a series of numerical representations or vectors that capture the contextual information of the input.
- **Decoder**: takes encoded vectors and generates the output text.

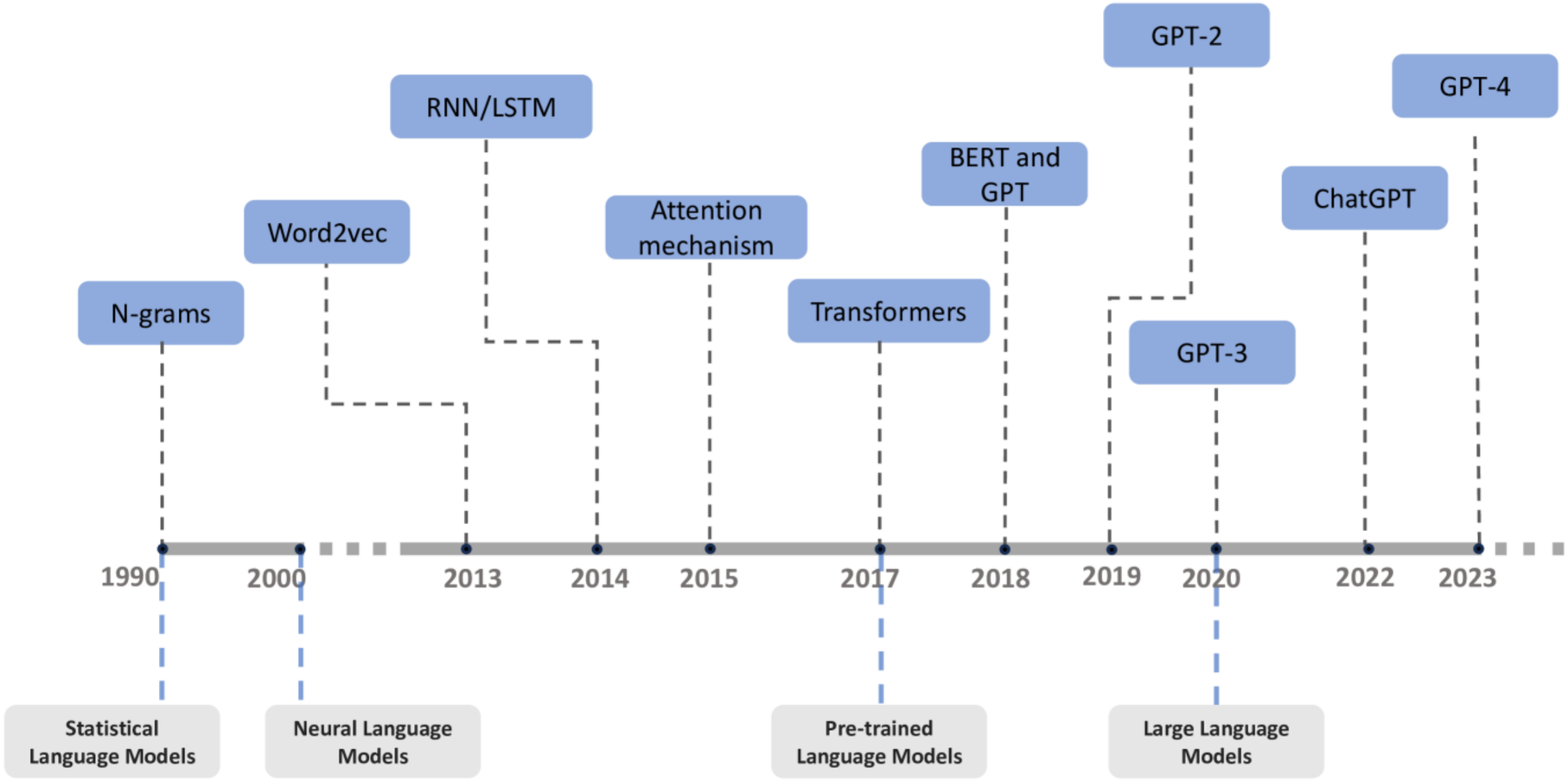


Building a large language model



History of pre-trained models

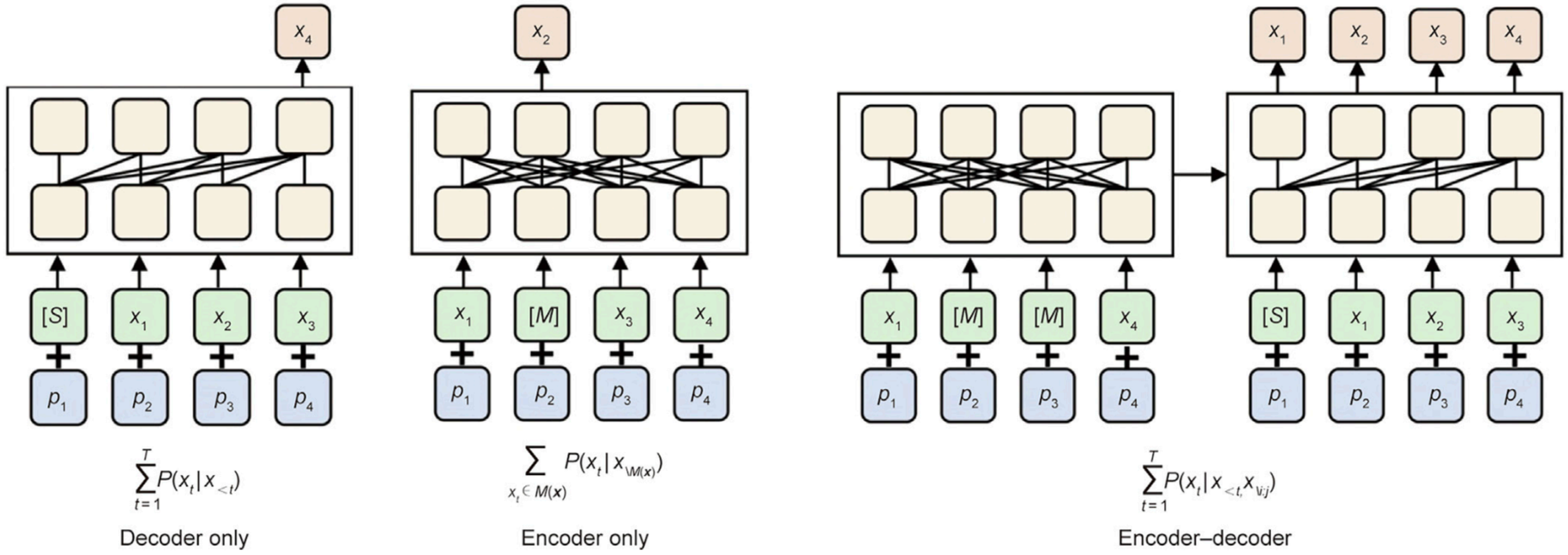
History of pre-trained models



Methods of PTMs

Methods of PTMs

- Transformer decoders only
- Transformer encoders only
- Transformer encoder–decoders



Transformer decoders only

Decoder-only models use transformer decoders by applying autoregressive masks to prevent the current token from attending to future tokens.

- Specialize in text generation (e.g., GPT, Llama).
- Predict the next word based on previous words.
- Used for chatbots, text completion, and story writing.
- Cannot process an entire input sequence at once (no bidirectional context).
- Struggles with tasks requiring deep comprehension, like classification.

Transformer encoders only

Encoder-only models at scale employ a bidirectional transformer encoder to learn contextual representation

- Focus on understanding input text (e.g., BERT).
- Capture contextual meaning efficiently.
- Used for tasks like classification, sentiment analysis, and question answering.
- Requires fine-tuning for different tasks.

Transformer encoder–decoders

Process input using an encoder and generate output with a decoder (e.g., T5, BART).

- Suitable for tasks like translation, summarization, and text transformation.
- Typically larger and more resource-intensive.

PTMs at scale

PTMs at scale

Summary of large-scale pre-trained language models.

Model	Number of parameters	Model architecture	Knowledge learning	Language	Pre-training data	Training strategy	Training platform
$DeBERTa_{1.5B}$	1.5 billion	Encoder only	—	English	English data (78 GB)	—	PyTorch
T5	11 billion	Encoder–decoder (seq2seq)	—	English	C4 (750 GB)	Model/data parallelism	TensorFlow
GPT-3	175 billion	Decoder only	—	English	Cleaned CommonCrawl, WebText	Model parallelism	—
CPM	2.6 billion	Decoder only	—	Chinese	Chinese corpus (100 GB)	—	PyTorch
PanGu-α	200 billion	Decoder only	—	Chinese	Chinese data (1.1 TB, 250 billion tokens)	MindSpore auto-parallel	MindSpore
ERNIE 3.0	10 billion	Encoder–decoder (unified)	√	Chinese, English	Chinese data (4 TB), English data	Model/pipeline/ tensor parallelism	PaddlePaddle
Turing-NLG	17 billion	Decoder only	—	English	English data	DeepSpeed/ZeRO	—
HyperCLOVA	204 billion	Decoder only	—	Korean	Korean data	—	—
CPM-2	11 billion	Encoder–decoder (seq2seq)	—	Chinese, English	WuDao corpus (2.3 TB Chinese + 300 GB English)	—	PyTorch
CPM-2-MoE	198 billion	Encoder–decoder (seq2seq)	—	Chinese, English	WuDao corpus (2.3 TB Chinese + 300 GB English)	Mixture of Experts (MoE)	PyTorch
Switch transformers	1751 billion	Encoder–decoder (seq2seq)	—	English	C4 (750 GB)	MoE	TensorFlow
Yuan 1.0	245 billion	Encoder–decoder (unified)	—	Chinese	Chinese data (5 TB)	Model/pipeline/ tensor parallelism	—
GLaM	1.2 trillion	Encoder only	—	English	English data (1.6 trillion tokens)	MoE/model parallelism	TensorFlow
Gopher	280 billion	Decoder only	—	English	English data (10.5 TB)	Model/data parallelism	Jax

ZeRO: zero redundancy optimizer; MoE: mixture-to-expert.

PTMs at scale - cont.

- The dramatic progress in language PTMs has attracted research interest on multimodal pre-training
- DALL-E is a 12-billion variant of GPT-3 that was trained on 250 million English text–image pairs to generate images according to language descriptions, thereby improving the zero-shot learning performance.

Large-scale multimodal PTMs.

Model	Number of parameters	Pre-training paradigm		Pre-training Data	Training parallelism	Training platform
		Denosing auto-encoder	Causal language model			
DALL-E	12 billion	×	√	250 million English text–image pairs	Mixed-precision training	PyTorch
CogView	4 billion	×	√	30 million English text–image pairs	—	PyTorch
M6	100 billion	√	×	1.9 TB images + 292 GB Chinese	MoE	—
ERNIE-ViLG	10 billion	√	√	145 million Chinese text–image pairs	Mixed-precision training	PaddlePaddle

Impact and challenges of PTMs

Impact and challenges of PTMs

Impacts:

- Natural language understanding
- Natural language generation
- Dialogue

Challenges:

- Deployability
- Model trustworthiness
- Commonsense knowledge and reasoning
- Model security

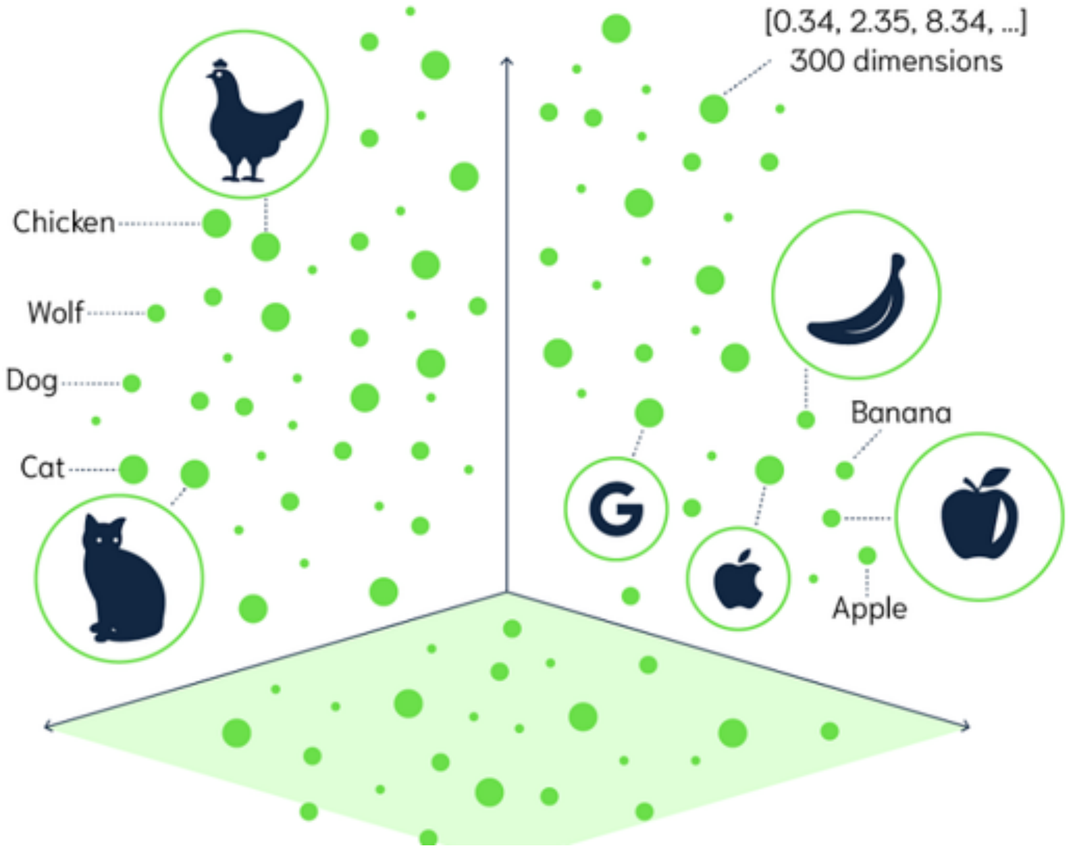
Applications of PTMs

Applications of PTMs

- Document intelligence
- Content creation
- Virtual assistants
- Intelligent search

Semantic search uses meaning and context to improve search results, rather than just relying on keyword matching.

- Use text embedding models(e.g., BGE) to get embeddings and store in vector databases(e.g., ChromaDB)
- Use text embedding models(e.g., BGE) to get embeddings of the query
- Get Cosine similarity between query and data in the vector space of embeddings



Conclusion

- The emergence of PTMs opens up a new “pre-training then finetuning” paradigm for NLP.
- With the increase of model parameters, PTMs show promising performance in zero-shot learning or fewshot learning.
- PTMs have demonstrated limited capability for commonsense awareness and reasoning, which require further improvement.
- In summary, there is still a long way to go for PTMs to be able to make reliable decisions and carry out reliable planning, which are essential elements of AI.

References

- Pre-Trained Language Models and Their Applications - Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, Yu Sun
- Improving Language Understanding by Generative Pre-Training - OpenAI
- Build a Large Language Model - Sebastian Raschka
- History, Development, and Principles of Large Language Models—An Introductory Survey
- Speech and Language Processing (3rd ed.) - Dan Jurafsky, James H. Martin
- Deep learning - Hamid Beigy
- Google cloud tech - YouTube
- HuggingFace - YouTube

Questions?