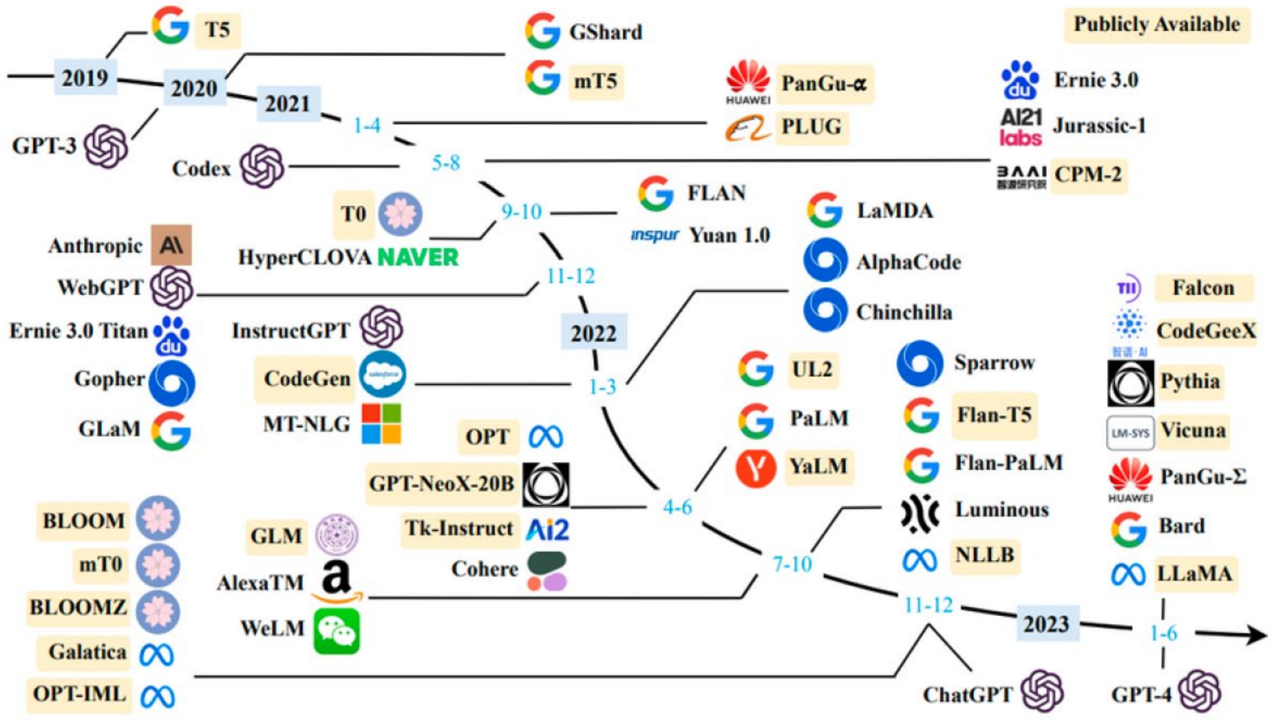
LARGE LANGUAGE MODELS

Source: Laconia Capital

Introduction

- LLM is a type of AI that process and generates text.
- Some of the tasks it can accomplish:
  - Question answering
  - Text generation
  - Summarization
  - Translation
  - Code generation
  - Data analysis
  - Creative writing
- Aspects to consider when selecting (?) LLM:
  - Model performance e.g., accuracy, language fluency, contextual understanding, ability to generate coherent response.
  - Customizability and fine-tuning
  - Ethical and responsible AI
  - Third-party verification and openness – model should be verified by third-party/external audits
  - Observability and debugging
  - Cost and scalability
  - Security and privacy



A timeline of existing large language models, arXiv:2303.18223

History

Google T5

Transfer learning – ***pre-training*** a model on abundantly-available unlabeled text data with a self-supervised task such as language modelling or filling in missing words. After that, the model can be ***fine-tuned*** on smaller labeled datasets.

Additional ref: Improving Language Understanding by Generative Pre-Training by OpenAI

Demonstrate large gains can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task.

The ability to learn effectively from raw text is crucial to alleviating the dependence on supervised learning in natural language processing.

Approach: semi-supervised for language understanding tasks using a combination of unsupervised pre-training and supervised fine-tuning.

Aim: learn a universal representation that transfers with little adaptation to a wide range of tasks.

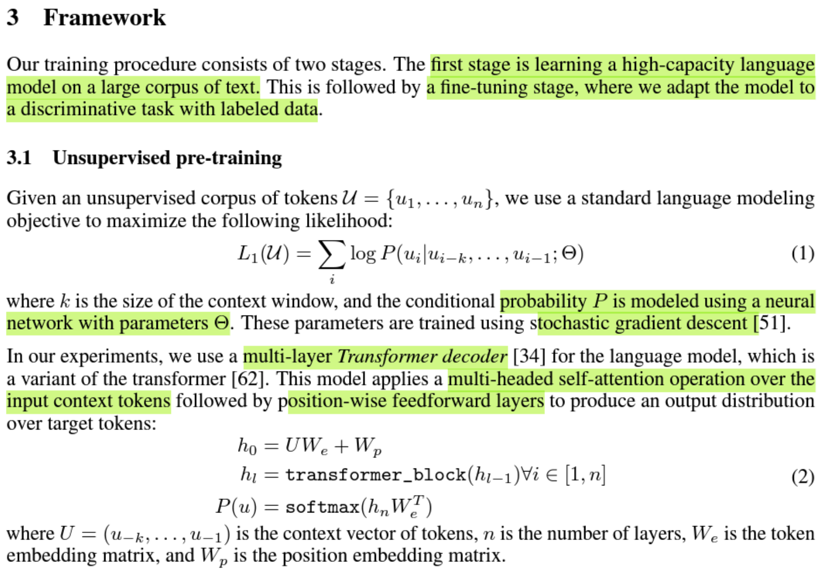
Steps: 1. Use language modeling objective on the unlabeled data to learn the initial parameters of a neural network model.

2. Adapt these parameters to a target task using the corresponding supervised objective.

Evaluation criteria:

1. Natural language inference
2. Question answering
3. Semantic similarity
4. Text classification

Multi-layer transformer decoder
- applies a multi-headed self-attention operation over the input context tokens
- position-wise feedforward layers



3 Framework

Our training procedure consists of two stages. The first stage is learning a high-capacity language model on a large corpus of text. This is followed by a fine-tuning stage, where we adapt the model to a discriminative task with labeled data.

3.1 Unsupervised pre-training

Given an unsupervised corpus of tokens $\mathcal{U} = \{u_1, \ldots, u_n\}$, we use a standard language modeling objective to maximize the following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta) \quad (1)$$

where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameters Θ. These parameters are trained using stochastic gradient descent [51].

In our experiments, we use a multi-layer *Transformer decoder* [34] for the language model, which is a variant of the transformer [62]. This model applies a multi-headed self-attention operation over the input context tokens followed by position-wise feedforward layers to produce an output distribution over target tokens:

$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n] \\ P(u) &= \texttt{softmax}(h_n W_e^T) \end{aligned} \quad (2)$$

where $U = (u_{-k}, \ldots, u_{-1})$ is the context vector of tokens, n is the number of layers, W_e is the token embedding matrix, and W_p is the position embedding matrix.

Experiments

Unsupervised pre-training:

Data: BookCorpus dataset for training the language model. It contains over 7000 unique unpublished books from a variety of genres including Adventure, Fantasy and Romance.

Model specifications:

- 12-layer decoder only transformer with masked self-attention heads
- Position-wise feed forward networks
- Adam optimization scheme with max learning rates of 2.5e-4 increased linearly from 0 over the first 2000 updates.
- Train for 100 epochs on minibatches of 64 randomly sampled, contiguous sequences of 512 tokens.
- Layernorm with initial weights of *N(0, 0.02)*
- Bytepair encoding (BPE) vocabulary with 40,000 merges
- Modified version of L2 regularization with *w=0.01* on all non-bias or gain weights
- Activation function: gaussian Error Linear Unit (GELU)
- Used *learned position embeddings* instead of the *sinusoidal version*
- Data cleaning: *fify* library, *spaCy* tokenizer

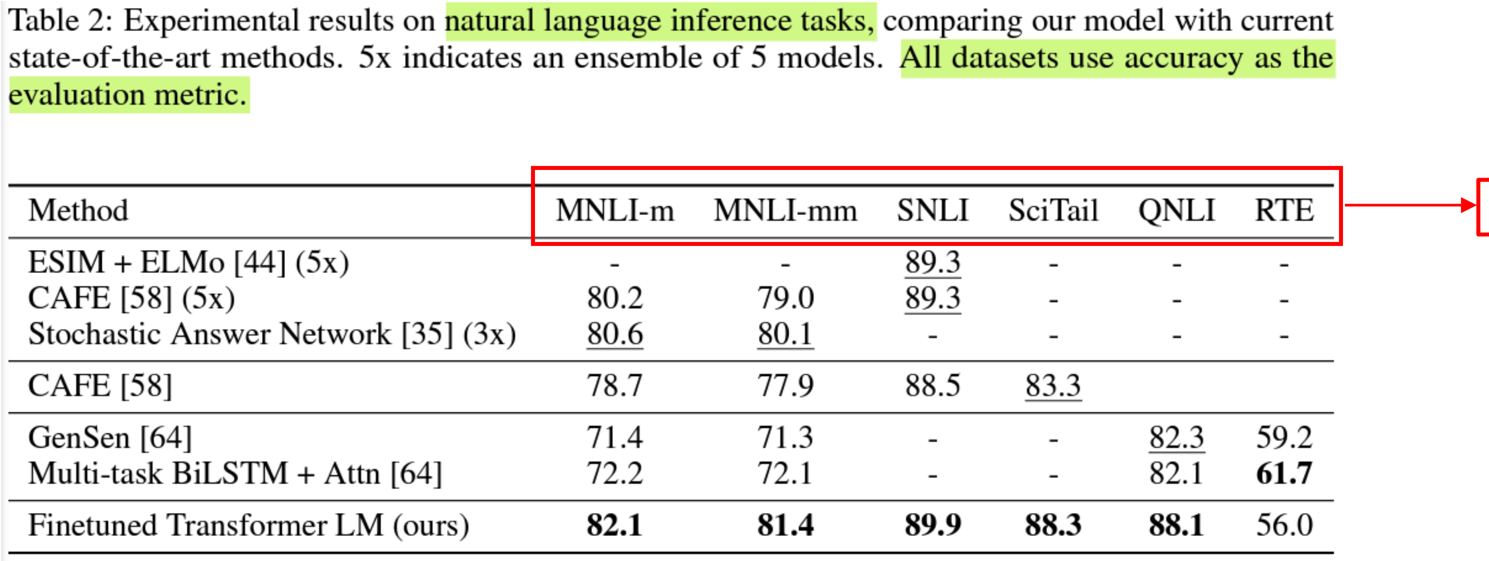


Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

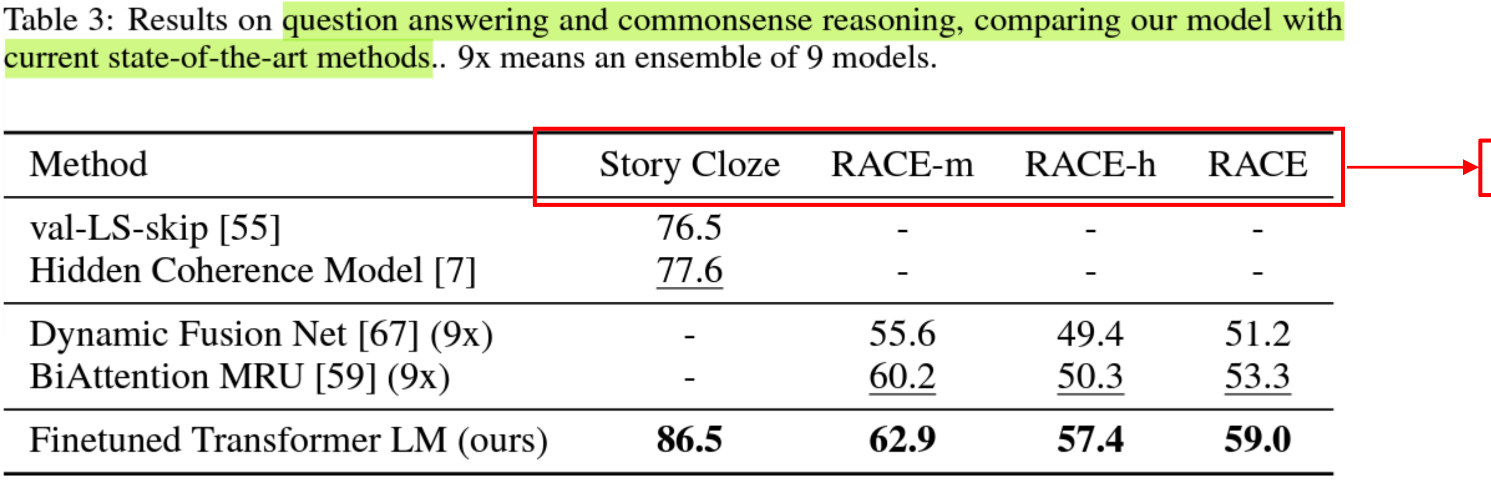


Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

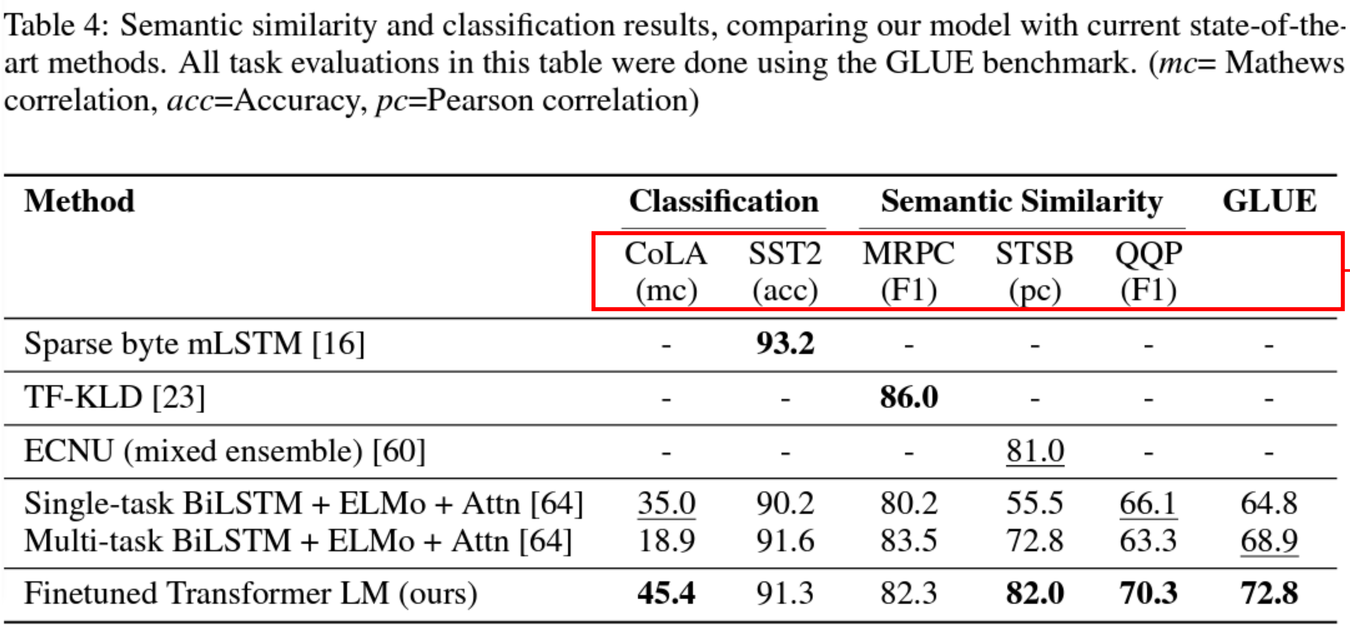


Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

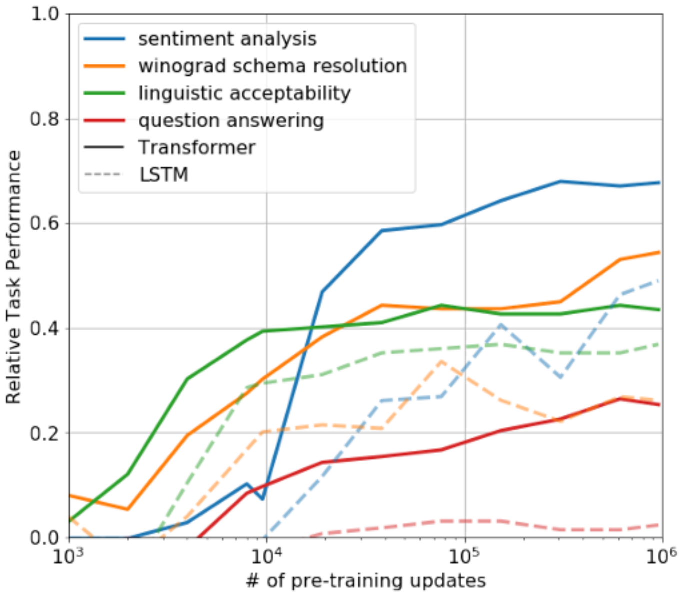
Method	**Classification**		**Semantic Similarity**			**GLUE**
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Analysis

1. Impact of number of layers transferred (from unsupervised pre-training to the supervised target task)

Findings: increasing the transferred layers improves performance. This indicates that each layer in the pre-trained model contains useful functionality for solving target tasks (caveat: result based on RACE and MultiNLI data sets)

2. Zero-shot behaviors



3. Ablation studies

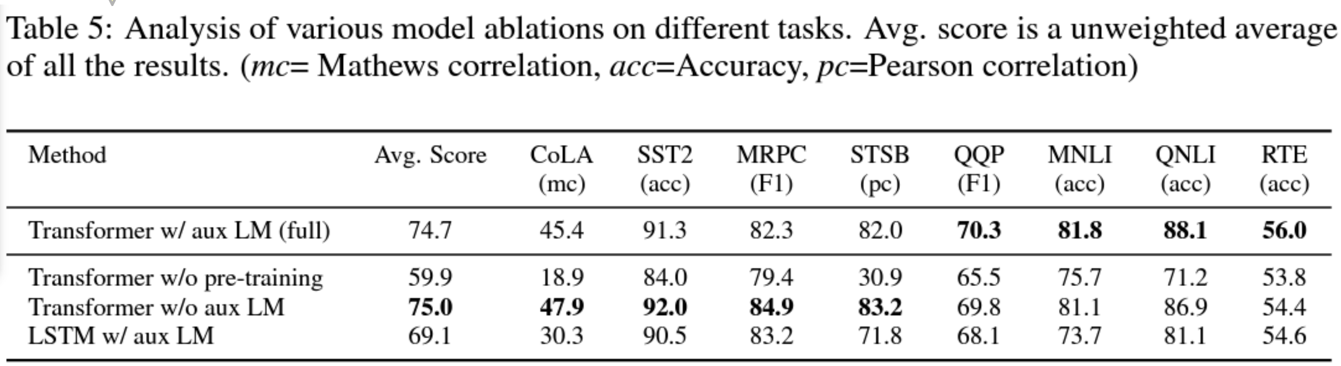
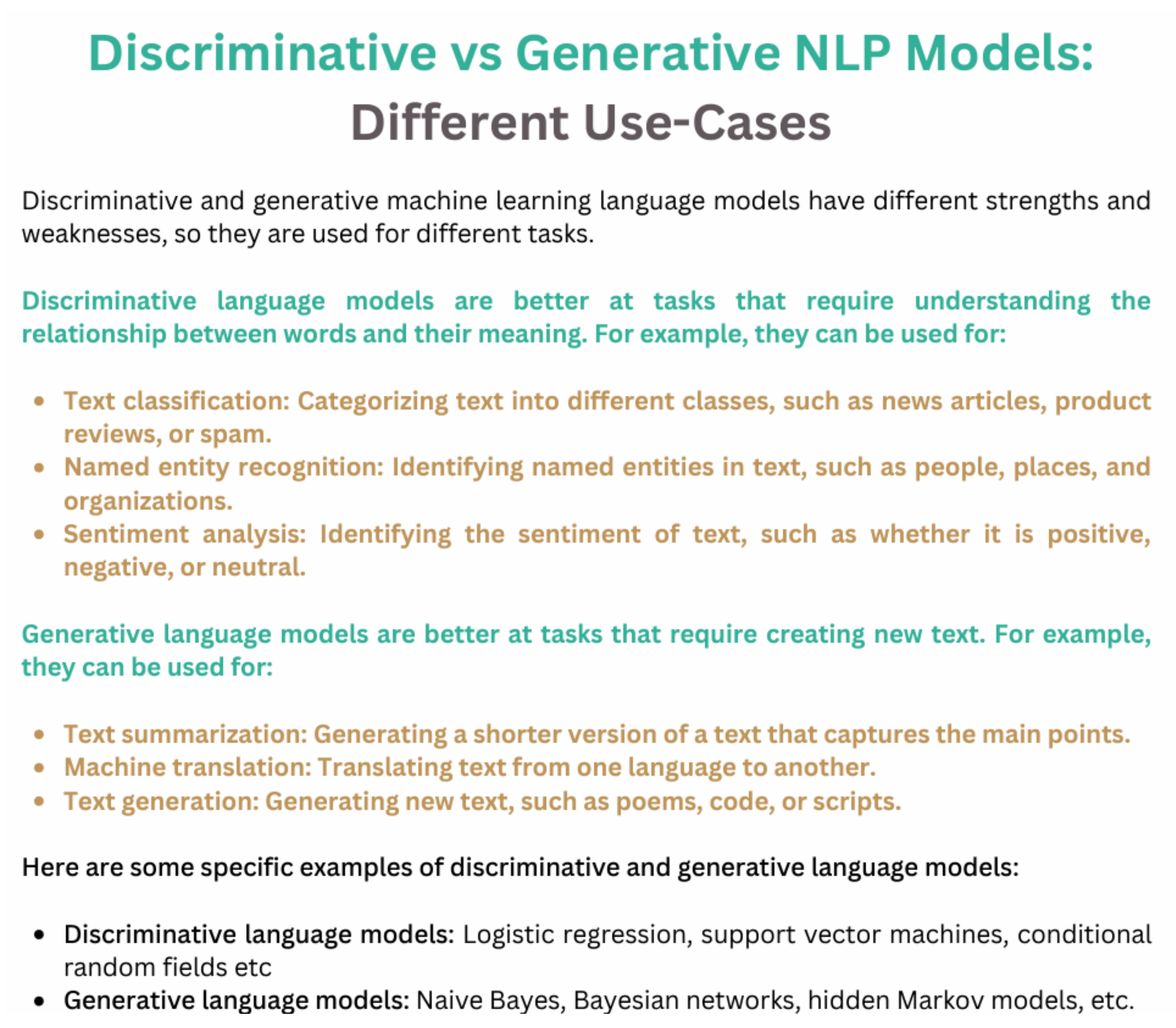


Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6



Discriminative vs Generative NLP Models: Different Use-Cases

Discriminative and generative machine learning language models have different strengths and weaknesses, so they are used for different tasks.

Discriminative language models are better at tasks that require understanding the relationship between words and their meaning. For example, they can be used for:

- **Text classification: Categorizing text into different classes, such as news articles, product reviews, or spam.**
- **Named entity recognition: Identifying named entities in text, such as people, places, and organizations.**
- **Sentiment analysis: Identifying the sentiment of text, such as whether it is positive, negative, or neutral.**

Generative language models are better at tasks that require creating new text. For example, they can be used for:

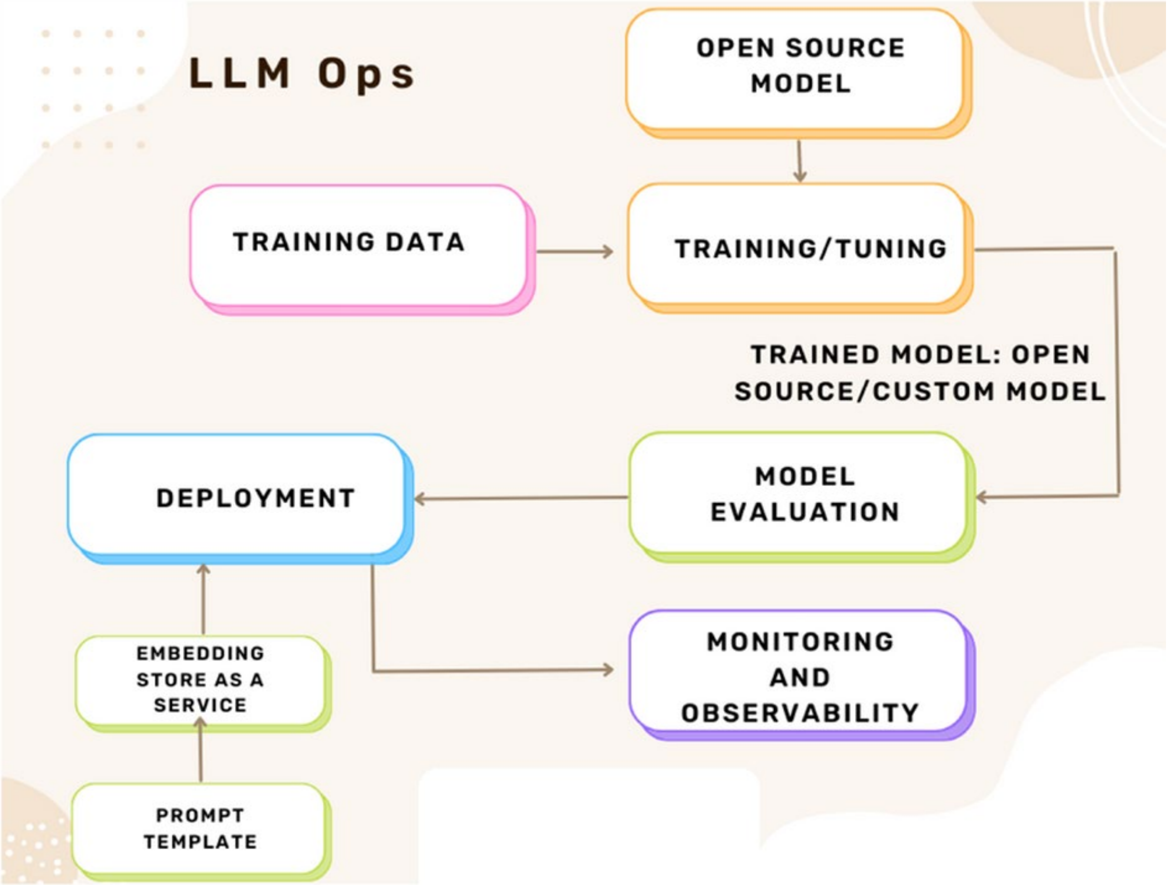
- **Text summarization: Generating a shorter version of a text that captures the main points.**
- **Machine translation: Translating text from one language to another.**
- **Text generation: Generating new text, such as poems, code, or scripts.**

Here are some specific examples of discriminative and generative language models:

- **Discriminative language models:** Logistic regression, support vector machines, conditional random fields etc
- **Generative language models:** Naive Bayes, Bayesian networks, hidden Markov models, etc.

Large language models (LLMs) are generative models.

LLMOps System Architecture


LLM Ops
OPEN SOURCE MODEL
TRAINING DATA
TRAINING/TUNING
TRAINED MODEL: OPEN SOURCE/CUSTOM MODEL
DEPLOYMENT
MODEL EVALUATION
EMBEDDING STORE AS A SERVICE
MONITORING AND OBSERVABILITY
PROMPT TEMPLATE

There are many tools available for fine-tuning large language models. Some of the most popular ones include:

- **Hugging Face Transformers:** This is a popular open-source library that provides easy access to pre-trained language models and utilities for fine-tuning. It supports a wide range of tasks, including text classification, question answering, and summarization.
- **PyTorch Lightning:** This is a framework for training deep learning models that is designed to be easy to use and efficient. It includes a number of features that are useful for fine-tuning large language models, such as distributed training and hyperparameter optimization.
- **DeepSpeed:** This is a library that can accelerate the training and inference of large language models by using a number of techniques, such as mixed precision and distributed training.
- **Google AI Platform:** This is a cloud platform that provides a number of tools for training and deploying machine learning models. It includes a managed instance of TPUs, which are specialized hardware accelerators for machine learning.
- **Amazon SageMaker:** This is another cloud platform that provides tools for training and deploying machine learning models. It also includes a managed instance of TPUs.
- **LMFlow:** An Extensible Toolkit for Finetuning and Inference of Large Foundation Models.

Evaluating LLMs

When assessing the performance of a Language Model (LM) for a business, a systematic evaluation strategy becomes crucial. There exists a range of methodologies for evaluating LM performance, each offering distinct advantages and drawbacks.

Business Focused Benchmarks

GLUE Benchmark

The General Language Understanding Evaluation (GLUE) benchmark comprises nine diverse natural language understanding tasks, serving as a comprehensive yardstick for assessing various LM models. The GLUE benchmark proves effective in evaluating LMs designed for versatile applications.

Task-Specific Downstream Evaluation

Alternatively, it might be more pertinent to appraise an LM's efficacy based on its designated task. For instance, an LM tailored for text classification can be assessed through conventional classification metrics like precision, recall, and F1 score.

Perplexity

Perplexity serves as a statistical gauge of an LM's text prediction confidence. Lower perplexity values reflect adept test set prediction, while higher values denote inadequate prediction. This metric is pertinent for evaluating LMs designed for text generation and machine translation tasks.

BLEU Score

The BLEU (bilingual evaluation understudy) score, ranging from 0 to 1, gauges the quality of machine translation compared to a reference translation. Higher BLEU scores indicate greater similarity to the reference text. This metric holds significance in evaluating LMs geared towards machine translation tasks.

Human Evaluation

In conjunction with statistical and automated evaluation methodologies, human evaluators play a pivotal role in assessing LMs' attributes such as creativity, humor, and toxicity. Their insights offer valuable feedback on the quality of LM-generated content.