Transformers, Self-Attention, PLMs and LLMs

CS-585

Natural Language Processing

Sonjia Waxmonsky

Based on slides from Kai Shu and Derrick Higgins

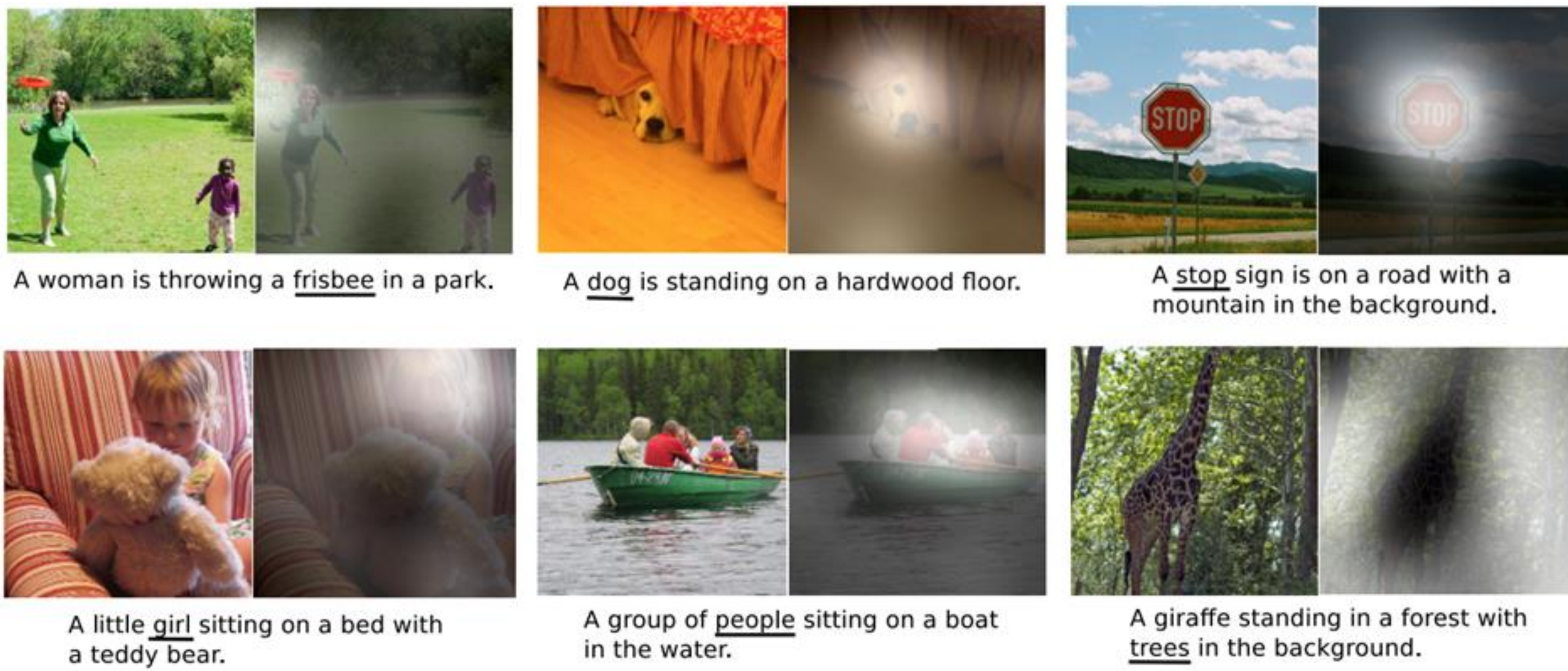
TRANSFORMERS AND SELF-ATTENTION

Attention

- Attention is a mechanism in neural network models to determine **how much weight** is given to different evidence (pixels, time steps or word vectors) in making a prediction
- Imagine a two-step process
  - First we determine **what information is relevant** for the prediction we want to make
  - Then we make a prediction using only the relevant information (or giving it **more weight**)

Attention

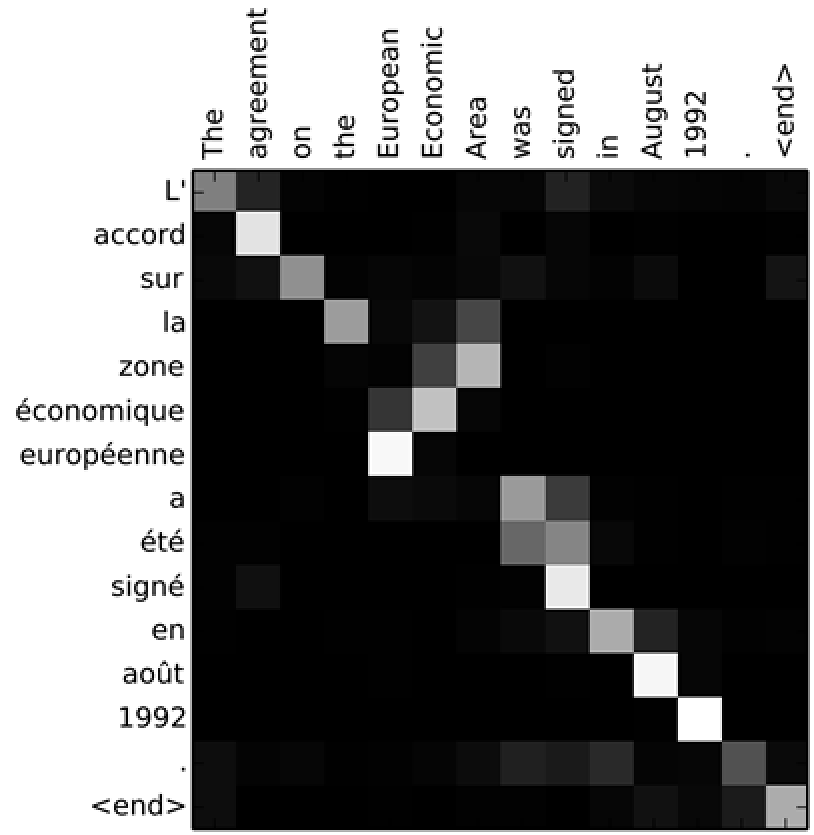
In computer vision, attention mechanisms are used to identify regions of the image that are relevant for classification



Xu, Kelvin, et al. *"Show, attend and tell: Neural image caption generation with visual attention."* *International Conference on Machine Learning*. 2015.

Attention

- Attention can also be used to determine how strongly words or context vectors are weighted in NLP

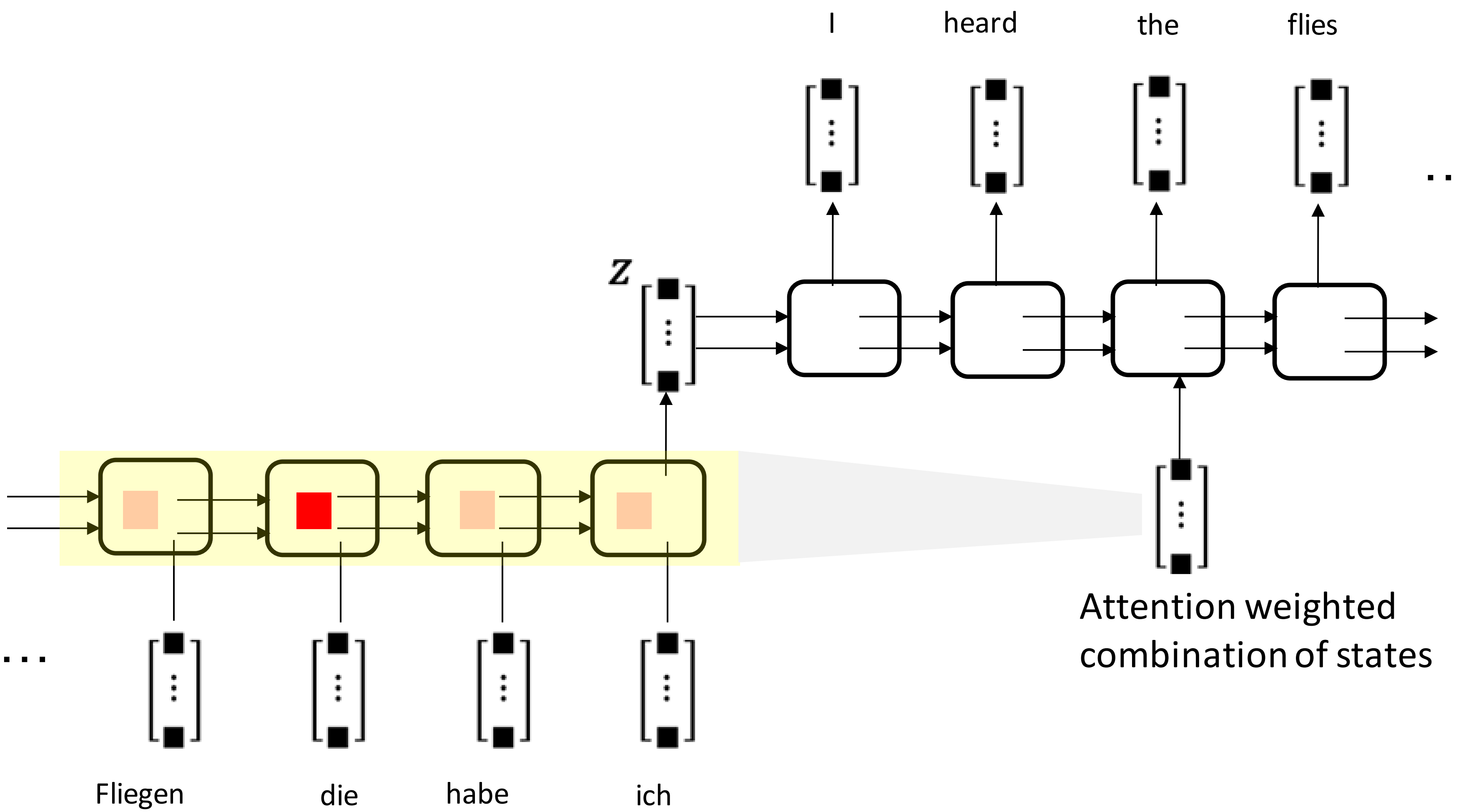


English → French translation

Bahdanau, D. et al. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR.

Attention

From: Session 26



Self-Attention

- **Self-attention** is attention between tokens within the **same layer**
- Tokens are encoded with a representation weighted by other tokens in the sequence
- Implemented as mapping (key, value, query) to an output

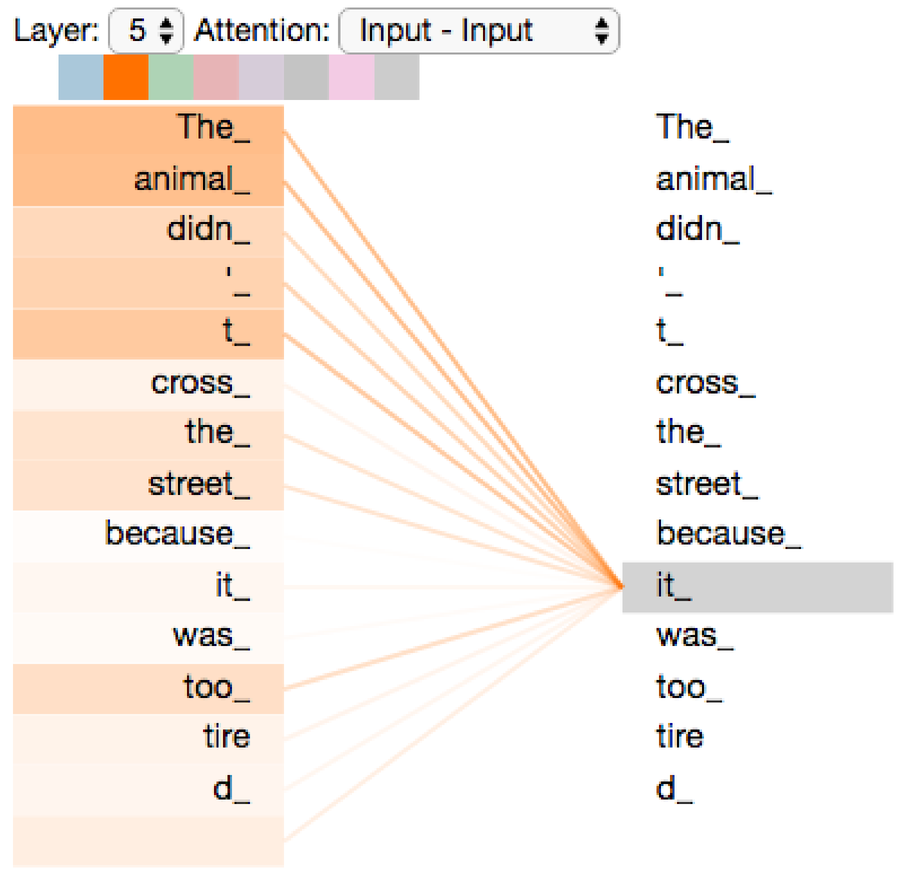


Image from "The Illustrated Transformer"
https://jalammar.github.io/illustrated-transformer/

Self-Attention

- Compares current focus of attention (q_1) to other words in the input sequence (k_1,k_2)
- Attention computes a distribution over input vectors (v_1,v_2)
- Outputs contextualized encoding (z_1) of inputs

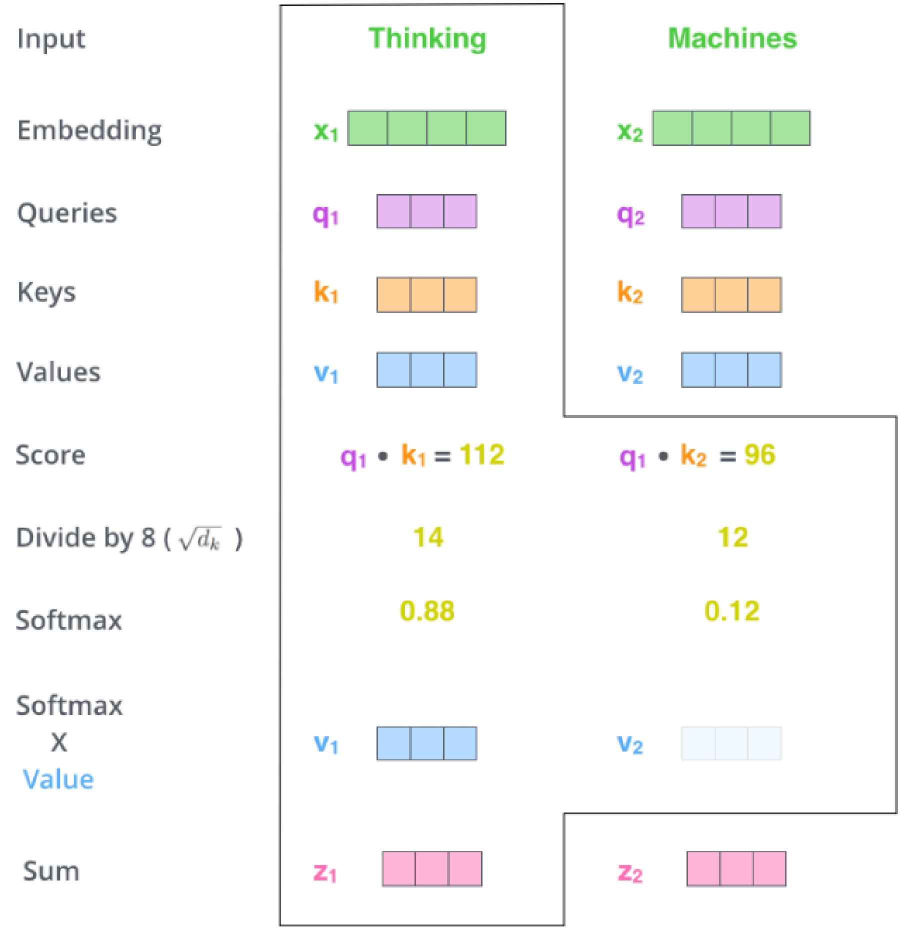


Image from "The Illustrated Transformer"
https://jalammar.github.io/illustrated-transformer/

The Transformer Architecture

Attention Is All You Need - Vaswani et al. 2017

- Transformer architecture applies **self-attention** to leverage **attention without recurrence** → faster training
- Multi-head attention: Parallel self-attention layers learn different relationships amount input words
- Positional encodings augment embeddings, encode word order information

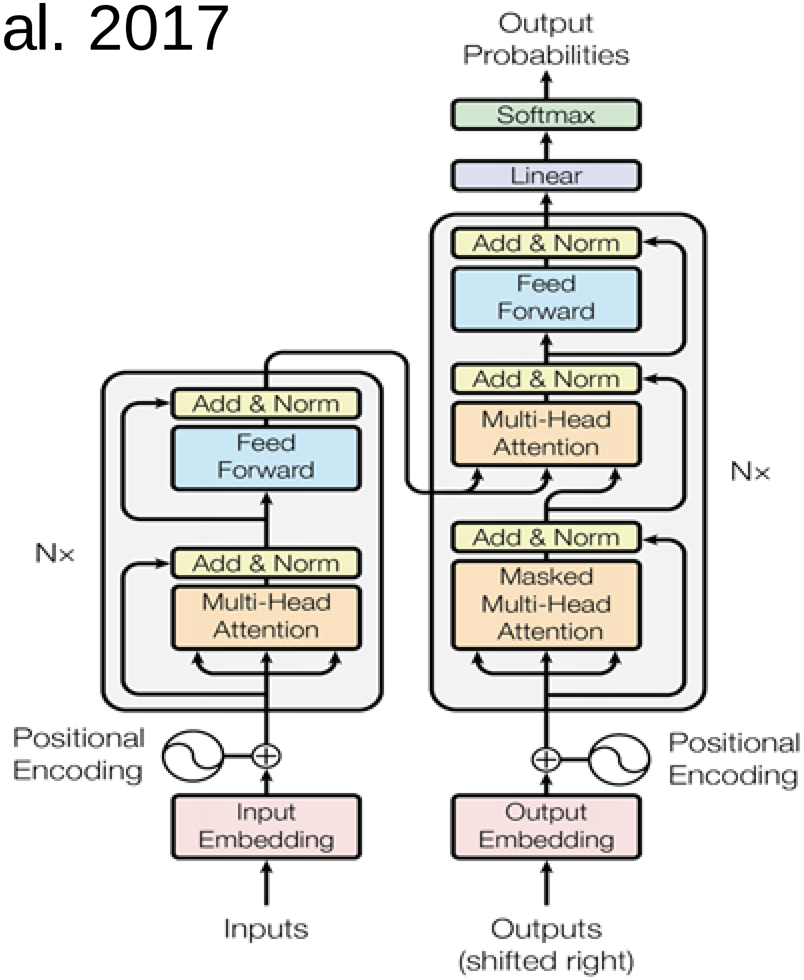


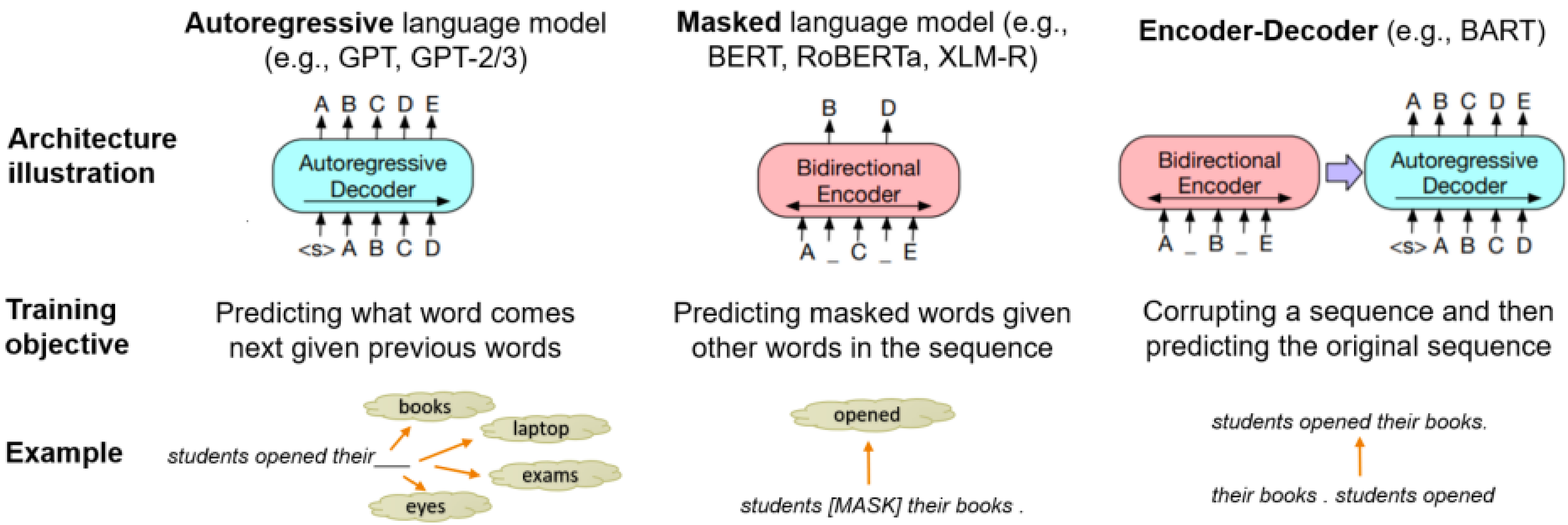
Figure 1: The Transformer - model architecture.

https://arxiv.org/pdf/1706.03762.pdf

PRETRAINED LANGUAGE MODELS

Pre-Trained Language Models

In recent years, NLP research labs have developed powerful transformer-based language models with varying configurations that have shown state-of-the-art results on a range of NLP tasks



Min et al. 2023 - https://dl.acm.org/doi/pdf/10.1145/3605943

BERT and friends

- BERT (Google AI, 2018)
  - **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
  - Transformer-based models trained with **Masked Language Modeling** (MLM) objective
- RoBERTa (Facebook AI, 2019)
  - **R**obustly **O**ptimized **BERT A**pproach
  - Training innovations on BERT: larger training corpus, more training iterations
- DistilBERT (Hugging Face 🤗 , 2019)
  - Applies "knowledge distillation" - Smaller model trained to reproduce a larger model
  - Fewer parameters and runs faster than BERT

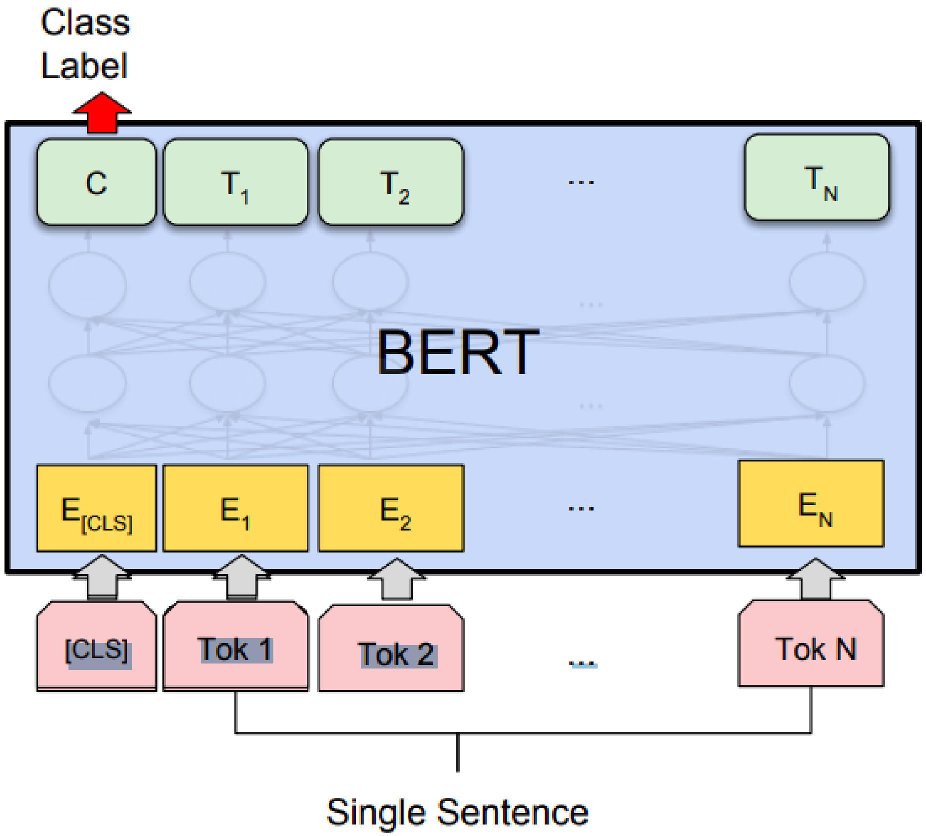
Transfer Learning with Pretrained Models

- **Transfer learning**: learning representations that will perform well across a range of tasks
  - Learn a latent representation of language from a generic task <u>once</u>
  - Then, apply it to many different NLP tasks
- **Fine-tuning** on Pretrained Language Models
  - Pretrained model is further trained on **task-specific** dataset
  - Output layer may be added/modified based on task

BERT for Text Classification

For classification:

- Use contextual representation of special initial [CLS] token to represent sentence
- Additional output layer with softmax → class probabilities



https://aclanthology.org/N19-1423.pdf

Contextual Word Embeddings

- Transformer models output **contextualized** word embeddings
  - Word representation depends on sentence context
  - Vector of each word/token is function of the entire input sentence
- **Sub-word encodings** for out-of-vocabulary (OOV) words
  - BERT: Applies WordPiece encoding → Limited vocabulary size (words, sub-words, characters)
  - Similarly, RoBERTA, GPT apply Byte Pair Encoding (BPE)
- Also see: **E**mbeddings from **L**anguage **Mo**dels (ELMo) (2018) - Applies Bi-Directional LSTM

Byte-pair encoding and word pieces

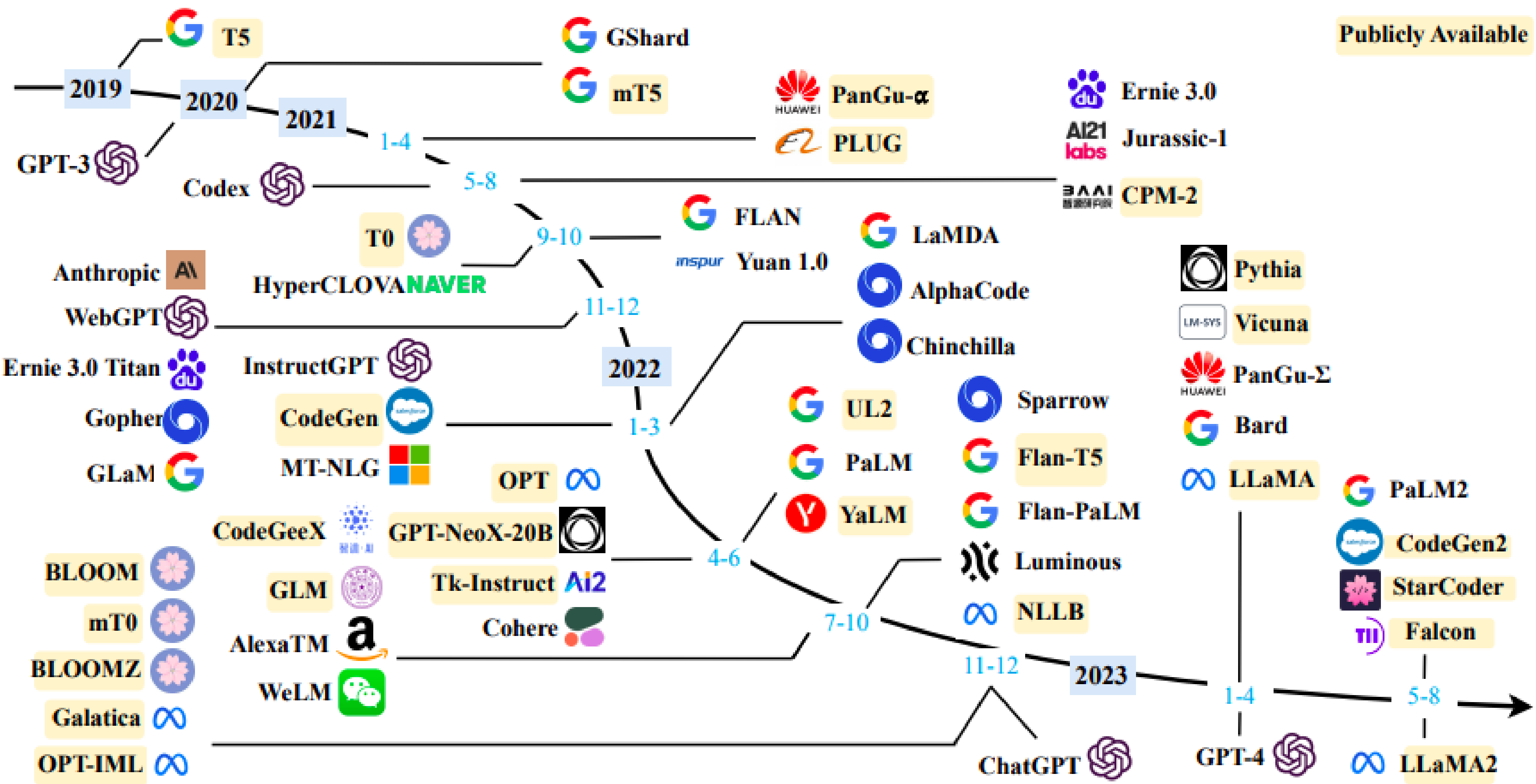
From Session 5: Lexical Representations

- *Byte pair encodings* and *word pieces* are two **unsupervised** methods for generating a **sub-word vocabulary** of a given size
- Based on Character *n-gram representation*
  - e.g. Bigram for *natural* : *#n, na, at, ur, ra, al, l#*
- More useful for **machine translation** and **natural language generation (**output individual tokens) than for text categorization
- Applied as tokenizers transformer models

TOPICS IN LARGE LANGUAGE MODELS

Large Language Models

(Zhao et al. 2023) - LLMs with 10B+ model parameters



https://arxiv.org/pdf/2303.18223.pdf

Large Langue Models

We are now seeing LLMs outperforming **human baselines** on Natural Language Understanding (NLU) tasks

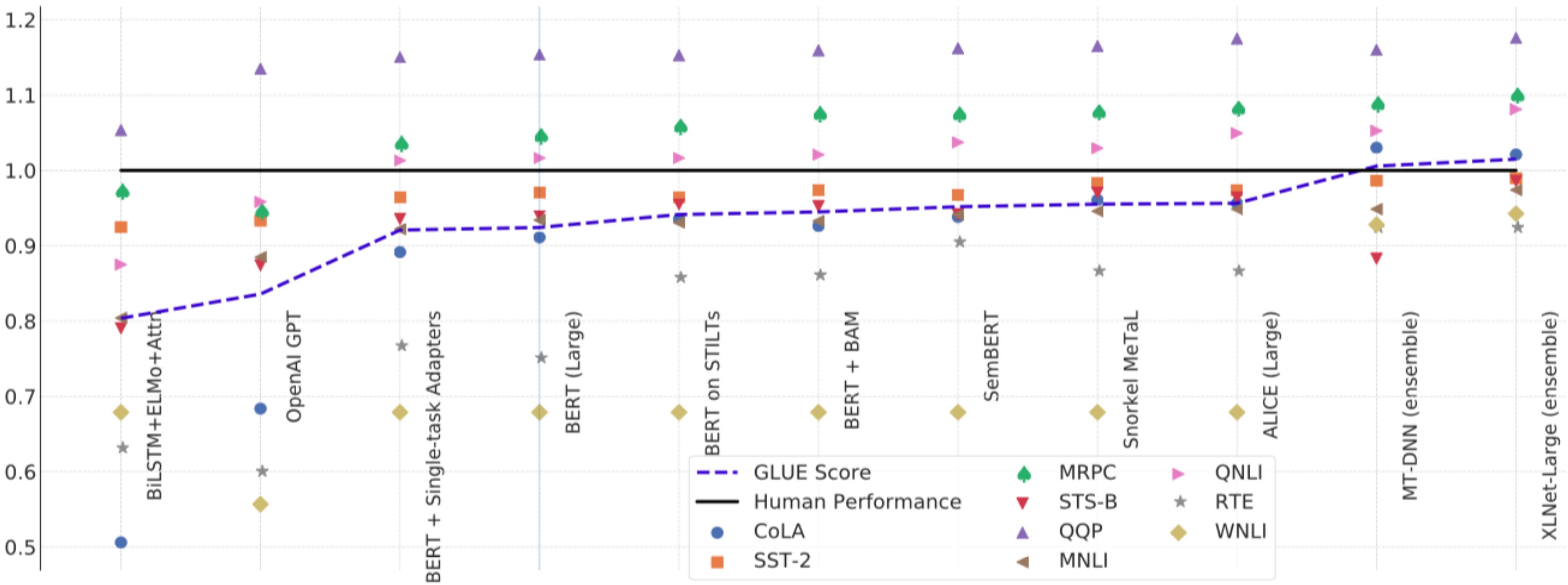


Chart: **GLUE performance** relative human performance

https://arxiv.org/pdf/1905.00537.pdf

LLMs and Prompt Engineering

Liu et al, 2021: LLMs bring a paradigm shift in NLP

- **Prompt engineering -** Identify the most appropriate prompt to allow a LM to solve the task at hand.
- Paradigm shift: “pretrain, fine-tune” → “pre-train, prompt, and predict.”
- Prompting function: Modifies input text into prompt
- **Cloze** prompt: LLM fills in blank
  - Prompting function: “[X] Overall, it was a [Z] movie"
  - Sample X: "I love this movie"
- **Prefix** prompt: LLM generates text following prompt
  - “Finnish: [X] English: [Z]” -

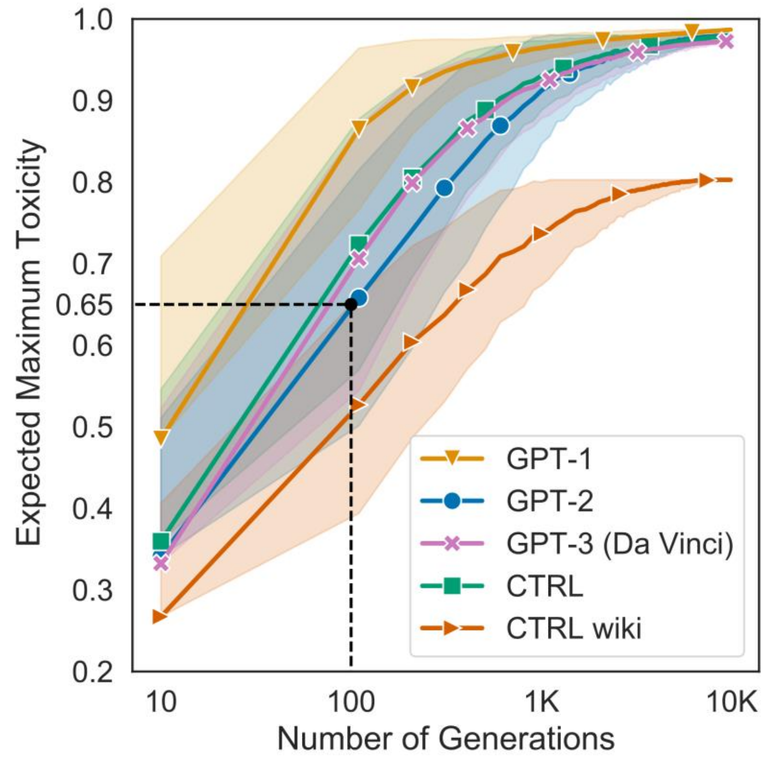
https://dl.acm.org/doi/pdf/10.1145/3560815

Bias in Pretrained Language Models

LLMs power brings risk of biased/toxic/hateful language

Gehman et al, 2020:

- LLMs can offensive toxic texts even when prompt do not include toxic language
- LLM **training datasets** contain non-trivial amount of offensive content
- "Toxicity" is subjective
- Best mitigation: Fine-tuning on non-toxic training data



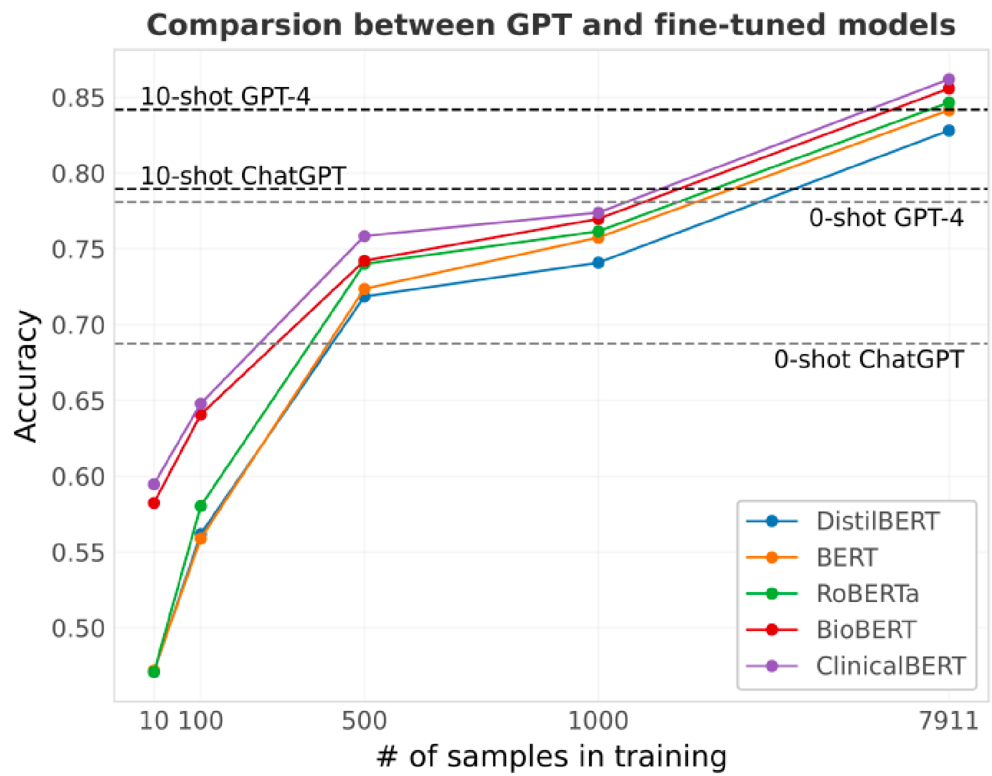
https://aclanthology.org/2020.findings-emnlp.301/

LLMs vs Fine-tuning

Open question: How do LLMs compare to fine-tuned models on **specialized domains** with specific vocabulary?

Wu et al 2023:

- Task: Natural Language Inference (NLI) in radiology domain
- LLMs surpass fine-tuned models at smaller training set
- At ~8K training samples: fine-tuned models match/ surpass LLMs



https://arxiv.org/pdf/2304.09138.pdf