5 – Generative models and evaluation of LLMs

IASD/MASH – LLMs course

Florian Le Bronnec

November 19, 2024

Table of Contents

Table of Contents

Standard tranformer model

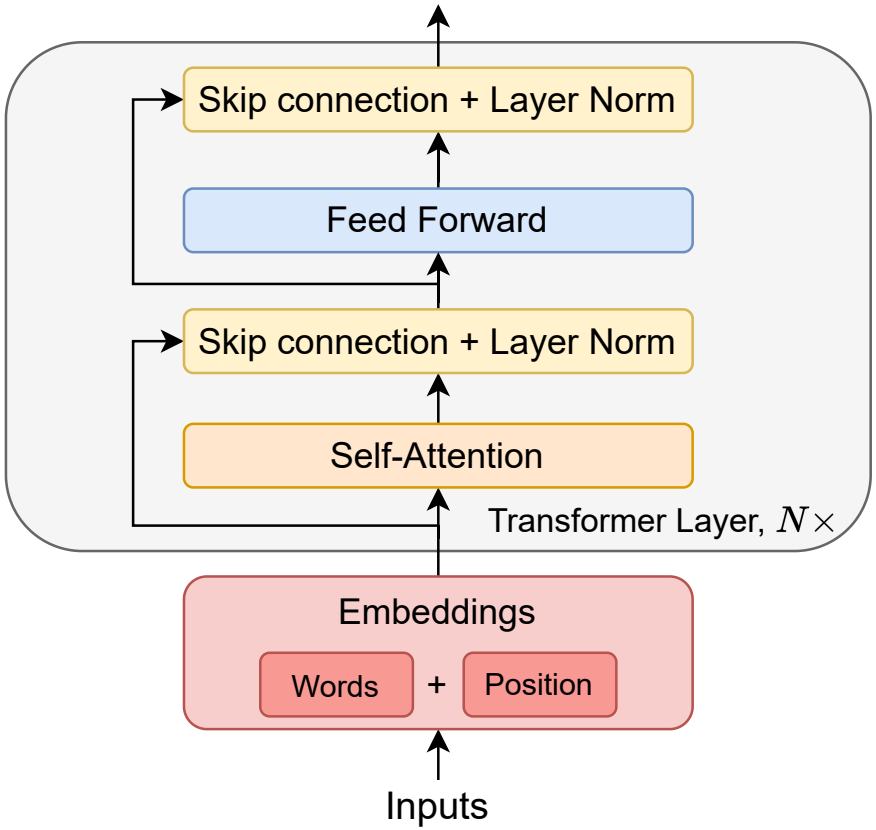

Figure 1: Standard stack of transformer layers.

BERT's MLM

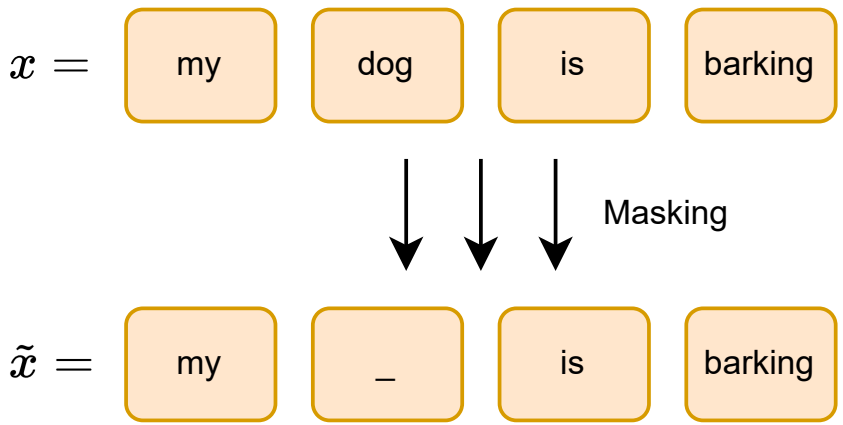

Figure 2: BERT masked language modeling.

$$\mathcal{L} = \sum_{\substack{w \in x \\ w \text{ is masked}}} -\log \mathrm{P}_\theta(w \mid \tilde{x})$$

BERT's Next sentence prediction

- Extract a sentence s_1 from a document $x \in \mathcal{D}$.
- With 50% chance, take s_2 the sentence following s_1.
- With 50% chance, take s_2 a random sequence from $\mathcal{D}$.

Predict if s_2 follows s_1:

$$\mathcal{L} = -\mathbf{1}_{s_2 \text{ follows } s_1} \log \mathrm{P}_\theta(s_1, s_2) - \mathbf{1}_{\text{random } s_2} \log(1 - \mathrm{P}_\theta(s_1, s_2))$$

Different kinds of attention

Attention in BERT

$$\begin{cases} s_{ij} &= \boldsymbol{q}_i^T \boldsymbol{k}_j \in \mathbb{R},\ 1 \leq j \leq \boldsymbol{L}, \\ \boldsymbol{\alpha}_i &= \text{Softmax}(\boldsymbol{s}_i) \in \mathbb{R}^{\boldsymbol{L}}, \\ \boldsymbol{y}_i &= \sum_{j=1}^{\boldsymbol{L}} \alpha_{ij} \boldsymbol{v}_j \in \mathbb{R}^d. \end{cases}$$

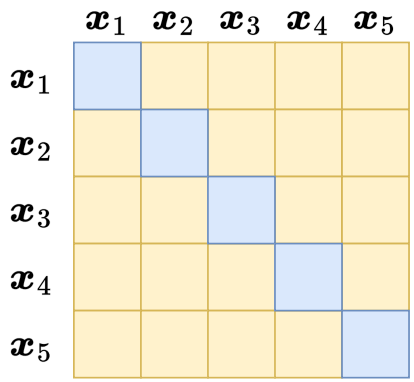

Figure 3: **Bidirectional** attention, tokens attend to every token.

Attention for generation

$$\begin{cases} s_{ij} &= \boldsymbol{q}_i^T \boldsymbol{k}_j \in \mathbb{R},\ 1 \leq j \leq \boldsymbol{i}, \\ \boldsymbol{\alpha}_i &= \text{Softmax}(\boldsymbol{s}_i) \in \mathbb{R}^{\boldsymbol{i}}, \\ \boldsymbol{y}_i &= \sum_{j=1}^{\boldsymbol{i}} \alpha_{ij} \boldsymbol{v}_j \in \mathbb{R}^d. \end{cases}$$

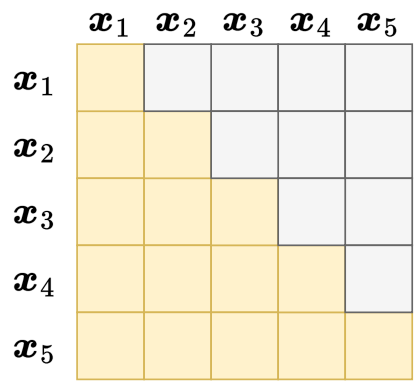

Figure 4: **Unidirectional** attention, tokens can only attend backward.

Generative models pre-training

$$\log \mathrm{P}_\theta(x) = \sum_{i=1}^{L} \log \mathrm{P}_\theta(x_i \mid x_{<i}).$$

Next token prediction objective.

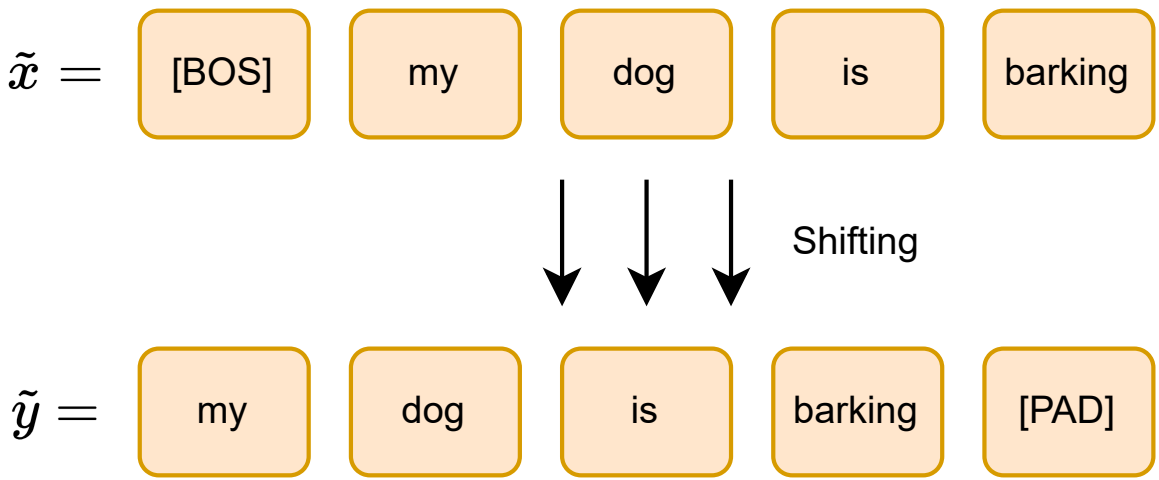

Figure 5: Generative models pre-training.

Finetuning

Finetuning to specific supervised tasks.

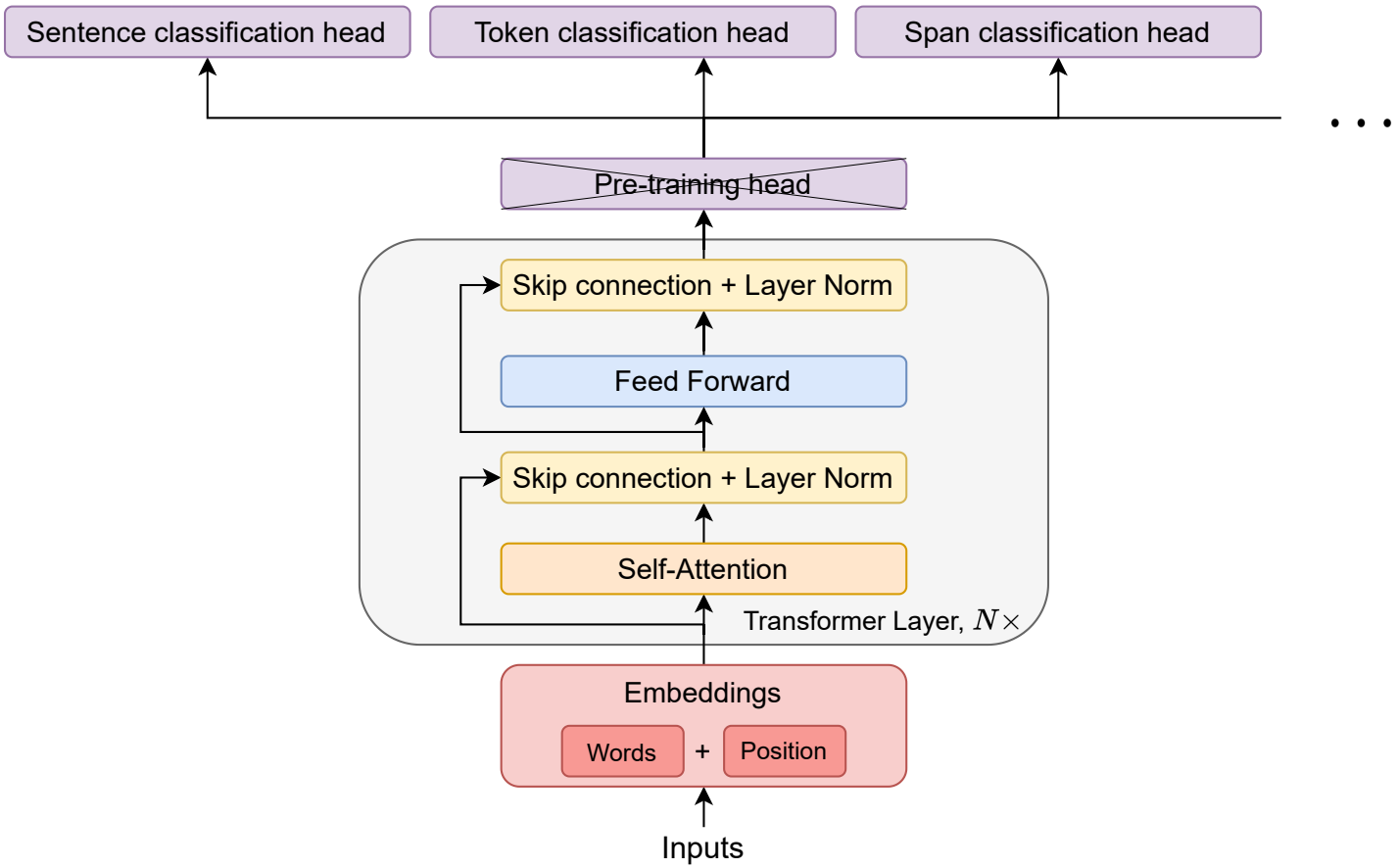

Figure 6: Switching from pretraining to finetuning.

Pre-training vs Fine-tuning

- **Pre-training**: Training the model on large, generic text data to learn language patterns.
  - **Objective**: Masked language modeling (MLM), where tokens are hidden, and the model learns to predict them.
- **Fine-tuning**: Adapting the pre-trained model to specific tasks by further training on a labeled dataset.
  - **Why it Helps**: Leverages pre-trained representations tailored for general language-related tasks to more specific applications. Faster and more data-efficient than training from scratch.

Sequence-Level Task Example: Sentiment Classification

- **Task**: Classify the sentiment of a sentence as positive, negative, or neutral.

Task Description	Model Input	Output Prediction
Text: "The movie was fantastic!"	`[CLS] The movie was fantastic [SEP]`	`3 classes: Positive, Negative, Neutral`

- **Example Label**: Positive
- **Common Dataset**: IMDb, SST-2

Sequence-Level Task Example: NLI

- **Task**: Classify sentence pairs as entailment, contradiction, or neutral.

Task Description	Model Input	Output Prediction
Sentence 1: "A person is playing a guitar." Sentence 2: "Someone is making music." Label: Entailment	`[CLS] A person is playing a guitar [SEP] Someone is making music [SEP]`	`3 classes: Entailment, Contradiction, Neutral`

- **Common Dataset**: MNLI

Token-Level Task Example: NER

- **Task**: Identify and label specific entities within a text.

Task Description	Model Input	Output Prediction
Text: "Apple was founded by Steve Jobs in Cupertino." Labels: `Apple (B-ORG)`, `was (O)`, `founded (O)`, `by (O)`, `Steve (B-PER)`, `Jobs (I-PER)`, `in (O)`, `Cupertino (B-LOC)`	`[CLS] Apple was founded by Steve Jobs in Cupertino [SEP]`	`Token-wise classes: B-ORG, I-ORG, B-PER, I-PER, B-LOC, I-LOC, O`

- **Example Labels**: `Apple (B-ORG)`, `Steve Jobs (B-PER, I-PER)`, `Cupertino (B-LOC)`
- **Common Dataset**: CoNLL-2003

Handling NER with SentencePiece Tokenizer

Sentence: `"Apple was founded by Steve Jobs in Cupertino."`

Word-Level Labels: `[B-ORG, O, O, O, B-PER, I-PER, O, B-LOC]`

After tokenization:

Tokenized: `_Apple, _was, _founded, _by, _Steve, _Jobs, _in, _Cuper, tino, .`

Aligned Labels: `[B-ORG, O, O, O, B-PER, I-PER, O, B-LOC, I-LOC, O]`

- Tokens prefixed with _ indicate the start of a new word.
- Subwords following the first are labeled as `I-` (inside entity).
- Non-entity tokens are labeled as `O`.

Classical NLP Tasks

Text Classification

- Sentiment Analysis
- Spam Detection
- Topic Classification

Classical NLP Tasks

Sequence Pairs Classification

- Natural Language Inference (NLI)
- Paraphrase Detection
- Stance Detection

Classical NLP Tasks

Token Classification

- Named Entity Recognition (NER)
- Part-of-Speech (POS) Tagging
- Chunking (Phrase Extraction)

Encoder Model in Practice

- **Summary**:
  - Encoder models like BERT can be fine-tuned for both sequence-level and token-level tasks.
  - Pre-training provides a foundation, and fine-tuning specializes the model for specific applications.
- **Transition**: Understanding encoder-based tasks is foundational; we'll build upon this with chat model architectures in the next slides.

Fine-Tuning for Generative Models

- **Goal**: Adapt pre-trained generative models (e.g., GPT) for specific tasks.
- **Focus**: GPT-like, decoder-only models.
- **Applications**: Translation, summarization, open-ended generation.

- During fine-tuning, model learns to generate sequences conditioned on specific inputs.
- Fine-tuning tasks are adapted by defining inputs and expected outputs in a way that aligns with task objectives.

Translation Task Example – Training

- **Task**: Translate sentences from English to French.

Task Description	Model Input (Training)	Output Prediction
Translate: "The cat is sleeping on the couch." Target: "Le chat dort sur le canapé."	`[BOS] Translate to French: The cat is sleeping on the couch. Le chat dort sur le canapé`	`Translate to French: The cat is sleeping on the couch. Le chat dort sur le canapé [EOS]`

- **Objective**: Model learns to predict the next token.

Translation Task Example – Inference

- **Task**: Translate sentences from English to French.

Task Description	Model Input (Inference)	Output Prediction
Translate: "The sun is shining."	`[BOS] Translate to French: The sun is shining.`	`Le soleil brille [EOS]`

- **Inference Process**: The model generates the translation autoregressively, conditioned on the Model Input until reaching `[EOS]`.

Summarization Task Example – Training

- **Task**: Summarize a long document into a concise summary.

Task Description	Model Input (Training)	Output Prediction
Document: "The global market saw a rise in stocks today as companies reported higher earnings than expected..." Summary: "Stocks rise with better-than-expected earnings."	`[BOS] Summarize: The global market saw a rise in stocks today... Stocks rise with better-than-expected earnings.`	`Summarize: The global market saw a rise in stocks today... Stocks rise with better-than-expected earnings [EOS]`

- **Objective**: Model learns to predict the output sequence, shifted by one token, ending with `[EOS]`.

Summarization Task Example – Inference

- **Task**: Generate a summary for a new document.

Task Description	Model Input (Inference)	Output Prediction
Document: "The recent advancements in AI have allowed for significant improvements in automation across industries..."	`[BOS] Summarize: The recent advancements in AI have allowed...`	`AI advancements boost automation in industries [EOS]`

- **Inference Process**: Model generates the summary autoregressively until reaching `[EOS]`.

Classical Generative Tasks in NLP

Examples of Generative Tasks

- **Text Summarization**: Generate a concise summary of a longer document.
- **Machine Translation**: Translate text from one language to another.
- **Dialogue Generation**: Produce conversational responses in chatbots.
- **Code Generation**: Generate programming code based on natural language descriptions.
- **Data-to-Text Generation**: Convert structured data (e.g., tables) into natural language descriptions.
- **Paraphrasing**: Rephrase text while preserving its meaning.
- **Question Answering**: Generate answers to questions based on a given context.

Conclusion: Fine-Tuning Generative Models

Training Recap

- **Goal**: Align inputs and outputs for task-specific data.
- **Objective**: Predict the next token based on input.
- Tasks framed as **sequence generation**, e.g., translation or summarization.

Inference Recap

- **Process**: Generate output token by token, conditioned on input.
- Stops at special token (`[EOS]`).
- Output tailored for task, e.g., **translation** or **summary**.

Table of Contents

Traditional evaluation

Evaluating classification models is easily done with **accuracy, precision, recall, F1-score**.

But what about **generative models**?

Automatic traditional evaluation of generative models

BLEU [1] and ROUGE [2] metrics compare the **n-gram overlap** between a generated text and a reference one.

Reference	**Candidate**	**BLEU**
The quick brown fox jumps over the lazy dog.	The fast brown fox jumps over the sleeping dog.	47

Table 1: Examples of BLEU (as a rule of thumb BLEU is high when it's over 40).

Basically, the **more words in common, the higher they are**.

Automatic traditional evaluation of generative models

BLEU [1] and ROUGE [2] metrics compare the **n-gram overlap** between a generated text and a reference one.

Reference	**Candidate**	**BLEU**
The quick brown fox jumps over the lazy dog.	The fast brown fox jumps over the sleeping dog.	47
The weather is pleasant.	The climate is nice.	0
The conference room has a big problem.	The conference room has a large table.	54

Table 1: Examples of BLEU (as a rule of thumb BLEU is high when it's over 40).

Basically, the **more words in common, the higher they are**.

More accuracte evaluation

Problem: How can we capture semantic meaning more accurately?

Solution: Ask humans to rate the generated texts.

$\Longrightarrow$ Human preferences is still today the reference way to evaluate generative models.

Human based evaluation

The quality of a generative model is very often assessed through a **human evaluation**, on top of the automatic evaluation.

Human evaluation

Humans are asked to rate generated texts along several criteria:

- Fluency,
- Coherence,
- Relevance,
- Factuality,
- etc.

Human based evaluation

The quality of a generative model is very often assessed through a **human evaluation**, on top of the automatic evaluation.

Human evaluation

Humans are asked to rate generated texts along several criteria:

- Fluency,
- Coherence,
- Relevance,
- Factuality,
- etc.

But it is **time consuming** and **expensive**. In practice it is often performed on a reduced number of samples.

Model based evaluation

- Automatic n-gram based evaluation is **limited**.
- Human evaluation is **expensive**.

Model based evaluation

- Automatic n-gram based evaluation is **limited**.
- Human evaluation is **expensive**.

$\Longrightarrow$ Use LLMs instead!

Two kinds of model-based metrics

Models have been used in different ways:

- Embeddings-based.
- Human-like evaluation.

BERT Score

BERT Score [7] leverages contextualized information.

- Word embeddings are contextualized through a **BERT** model.
- The score is calculated based on the **cosine similarity** between the embeddings of words in the reference and generated texts.

$$R_{\text{BERT}} = \frac{1}{L} \sum_{i=1}^{L} \max_{1 \le j \le \hat{L}} \boldsymbol{y}_i^\top \hat{\boldsymbol{y}}_j,$$

$$P_{\text{BERT}} = \frac{1}{\hat{L}} \sum_{j=1}^{\hat{L}} \max_{1 \le i \le L} \boldsymbol{y}_i^\top \hat{\boldsymbol{y}}_j,$$

$$F_{\text{BERT}} = \frac{2 \cdot P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.$$

Other model based metrics

Below are examples of other model-based metrics, where we extract some features from the model.

- BLEURT [6],
- COMET [4],
- BARTScore [5],
- etc.

GPT Evaluation

Since human evaluation is expensive, simply ask GPT4 to do it for you [9].

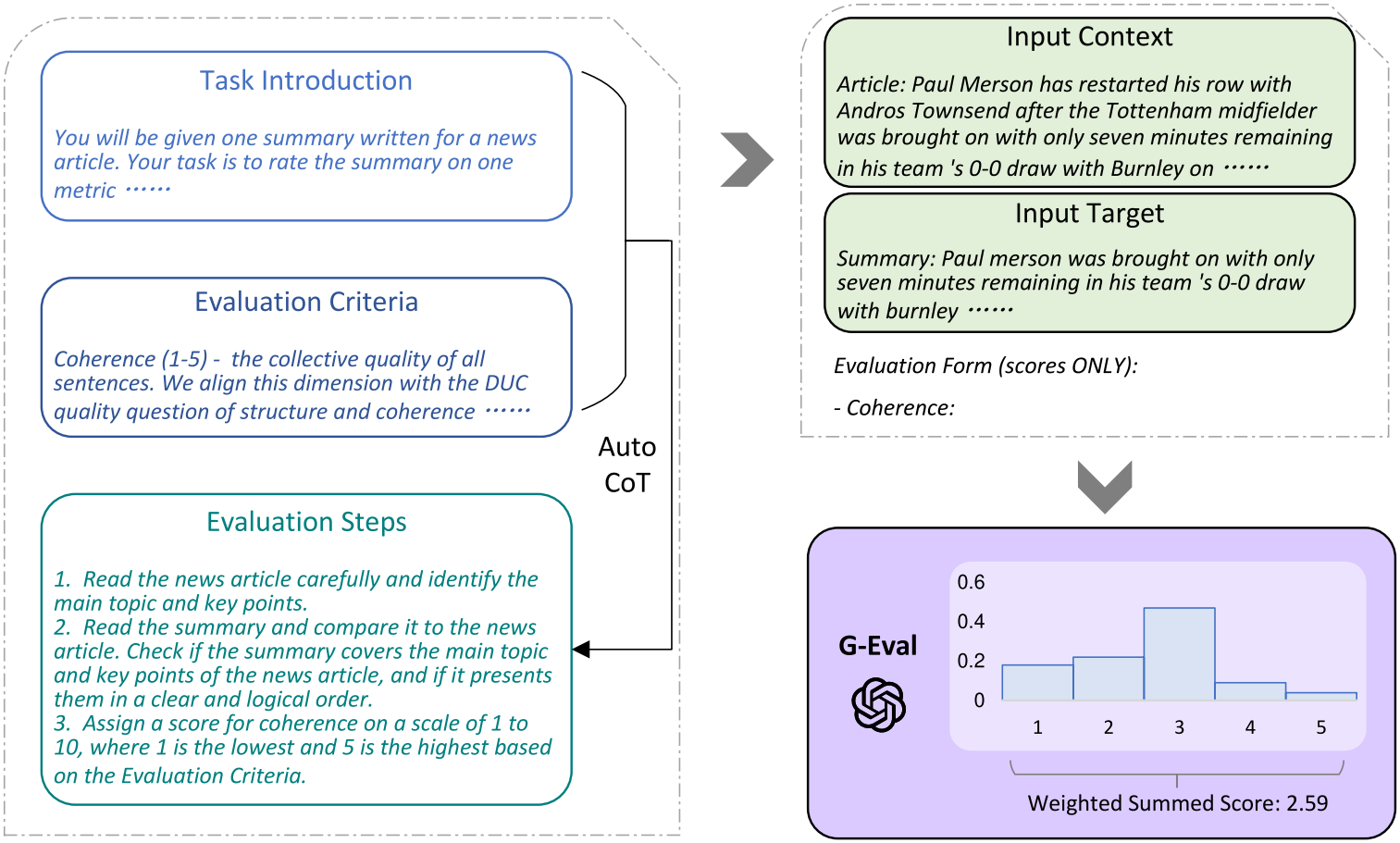

Figure 7: G-Eval, figure from [9].

Problems

Any thoughts?

Problems

Any thoughts?

A big blackbox model is evaluated by a big blackbox model.

LLM specific evaluation

In generalist LLM papers, there are often three kinds of evaluation:

- Human evaluation, performed on a limited number of samples,
- GPT4-Evaluation,
- Generalist benchmarks [8]. These benchmarks comprise a range of supervised tasks that are straightforward to assess, and the models are evaluated based on their performance in these tasks.

Biases in NLP

What are biases?

Biases in NLP are systematic deviations in model behavior that unfairly advantage or disadvantage certain groups, contexts, or tasks.

- **Source:** Imbalances in training data, model design, or evaluation.
- **Impact:** Stereotyping, unequal performance, and exclusion.
- **Challenge:** Ensuring fairness, equity, and inclusivity.

Biases are real

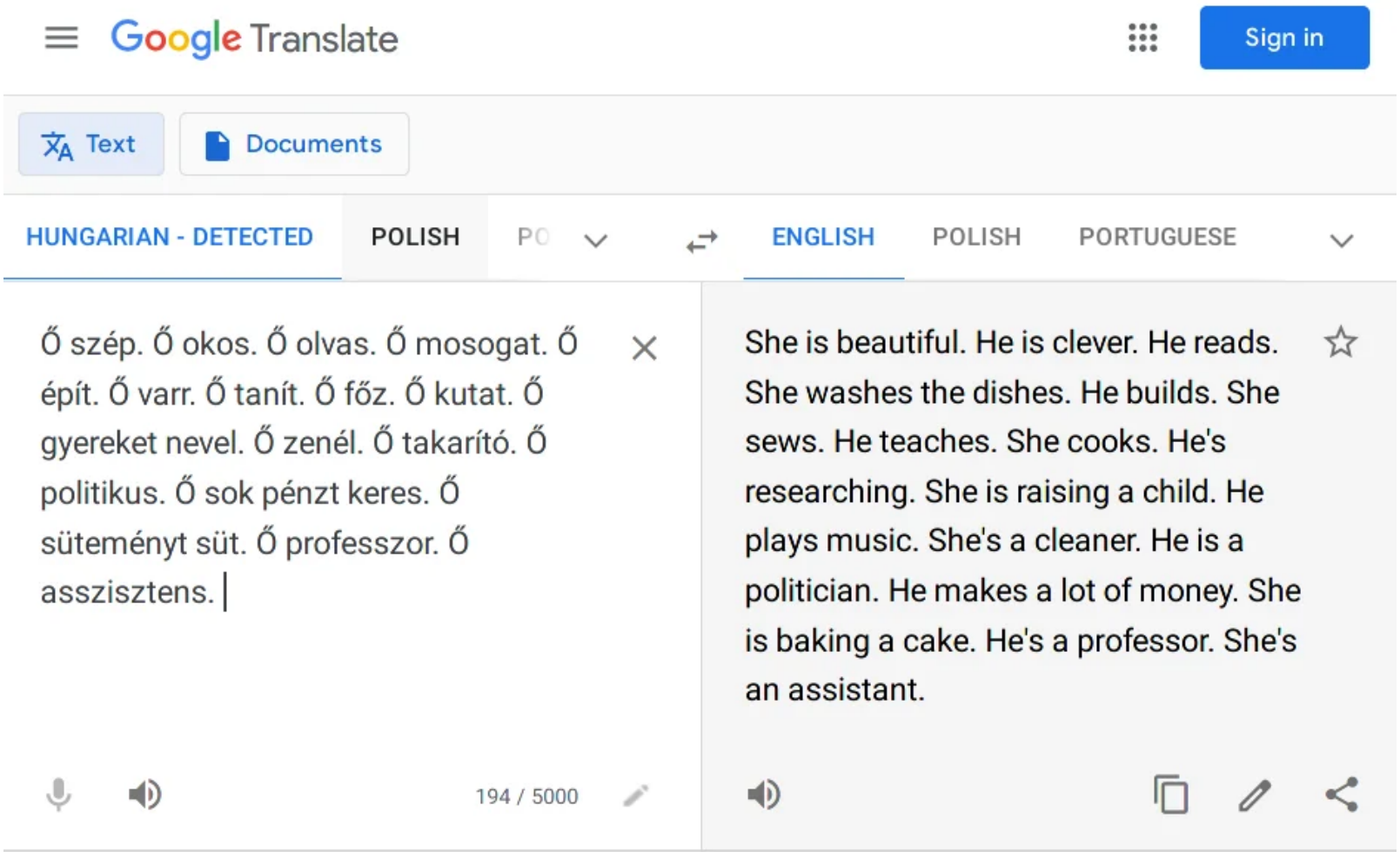

Figure 8: Hungarian has no gendered pronouns, so Google Translate makes some assumptions

Common Biases in NLP

Bias Type	Example	Possible Origin of Bias
Gender Bias	Associating "doctor" with men and "nurse" with women.	Imbalances in training data reflecting societal stereotypes.
Racial or Ethnic Bias	Linking certain ethnicities with crime or negative traits.	Overrepresentation of certain contexts in datasets (e.g., news articles) and lack of diverse data.
Cultural Bias	Assuming Western norms, such as MM/DD/YYYY date formats or holidays.	Dominance of Western-centric datasets during training.
Language Bias	Better performance in English compared to low-resource languages like Swahili.	Uneven representation of high-resource and low-resource languages in training data.

Toxicity in NLP

What is toxicity?

Toxicity in NLP refers to harmful or offensive content generated by language models, including hate speech, abusive language, or discriminatory remarks.

- **Examples:** Hate speech, slurs, harassment, and perpetuation of stereotypes.
- **Sources:** Presence of toxic data in training datasets, lack of robust filtering mechanisms, or amplification of biases.
- **Challenge:** Ensuring models avoid generating harmful content while maintaining freedom of expression and utility.

Biases and Toxicity – Evaluation and Mitigation

Evaluation Methods

- Test for biases in tasks like coreference resolution [3].
- Use toxicity classifiers (e.g., Perspective API).

Biases and Toxicity – Evaluation and Mitigation

Evaluation Methods

- Test for biases in tasks like coreference resolution [3].
- Use toxicity classifiers (e.g., Perspective API).

Mitigation Strategies

- Curate balanced and diverse training data.
- Apply data augmentation to reduce bias.
- Post-training adaptation (see next sessions).

Biases and Toxicity – Evaluation and Mitigation

Evaluation Methods

- Test for biases in tasks like coreference resolution [3].
- Use toxicity classifiers (e.g., Perspective API).

Mitigation Strategies

- Curate balanced and diverse training data.
- Apply data augmentation to reduce bias.
- Post-training adaptation (see next sessions).

Limitations

$\Longrightarrow$ No formal guarantees. Models like ChatGPT rely on empirical methods to minimize harm.

References I

[1] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. 2002, pp. 311–318.

[2] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text summarization branches out: Proceedings of the ACL-04 workshop*. 2004, pp. 74–81.

[3] Rachel Rudinger et al. *Gender Bias in Coreference Resolution*. 2018. arXiv: 1804.09301 [cs.CL].

[4] Mounica Maddela, Rico Sennrich, and Yvette Graham. *COMET: A Neural Framework for MT Evaluation*. 2020. arXiv: 2004.12867 [cs.CL].

[5] Feng Nan and Dan Klein. *BARTScore: Evaluating Generated Text as Text Generation*. 2020. arXiv: 2004.01970 [cs.CL].

References II

[6] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. *BLEURT: Learning Robust Metrics for Text Generation*. 2020. arXiv: 2004.04696 [cs.CL].

[7] Tianyi Zhang et al. *BERTScore: Evaluating Text Generation with BERT*. 2020. arXiv: 1904.09675 [cs.CL].

[8] Leo Gao et al. *A framework for few-shot language model evaluation*. Version v0.0.1. Sept. 2021. DOI: 10.5281/zenodo.5371628. URL: https://doi.org/10.5281/zenodo.5371628.

[9] Yang Liu et al. *G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment*. 2023. arXiv: 2303.16634 [cs.CL].