

Introduction to Language Models

CS RUB Preparatory Courses

Prof. Dr. Ivan Habernal

September 29, 2026

www.trusthlt.org

Chair of Trustworthy Human Language Technologies (TrustHLT)

Ruhr University Bochum & Research Center Trustworthy Data Science and Security



CENTER FOR TRUSTWORTHY
DATA SCIENCE AND SECURITY

Motivation

- 1 Motivation
- 2 What is a language model?
- 3 Language models
- 4 Large Neural Language Models
- 5 Fully open models
- 6 Want to learn more?

The thing called “AI”

What comes to your mind if you hear AI?

What is a language model?

- 1 Motivation
- 2 What is a language model?**
- 3 Language models
- 4 Large Neural Language Models
- 5 Fully open models
- 6 Want to learn more?

What is a language model?

So I was at this party at Joe's place last night, but man I was so tired, so like at about midnight I said, Joe, I'm __

Continue with the next word

With high probability:

- leaving
- out
- calling (it a night)
- gonna (head home)
- ...

With low probability:

- hungry
- in (love)
- ..."

This is what a language model does

What is a LARGE language model?

Language models are

- machine learning systems (models)
- mostly neural networks
- therefore complicated functions with parameters
- these parameters are learned from data

Large language models = neural networks for language modeling with a 'large' number of trainable parameters (in billions)

Language models

- 1 Motivation
- 2 What is a language model?
- 3 Language models**
- 4 Large Neural Language Models
- 5 Fully open models
- 6 Want to learn more?

Language models

Probability refresher

Probability refresher 1

Categorical random variables

For example, the first word in a sentence

$W_1 \in \{\text{the, be, to, of, and, ...}\}$, we assume a fixed vocabulary

Probability distribution over random variables

For example, probability of '*the*' at position 1

$$\Pr(W_1 = w_1) = \Pr(W_1 = \text{the}) = 0.00024$$

Notation shortcuts: $\Pr(W_1 = w_1) \rightarrow P(W_1), P(\text{the}), \text{etc.}$

Probability refresher 2

Joint probability

For example, probability of '*the*' at position 1 and '*cat*' at position 2

$$\Pr(W_1 = \text{the} \cap W_2 = \text{cat}) = 0.0000074$$

Notation shortcuts: $P(W_1, W_2) = P(W_2, W_1)$

Conditional probability

For example, probability of '*cat*' at position 2, **given** '*the*' at position 1

$$\Pr(W_2 = \text{cat} | W_1 = \text{the}) = \frac{P(W_1, W_2)}{P(W_1)}$$

Language models

'Classical' language models

Goal of language modeling

Assign a probability to sentences in a language

Example

“What is the probability of seeing the sentence *the lazy dog barked loudly*?”

Assigns a probability for the likelihood of given word (or a sequence of words) to follow a sequence of words

Example

“What is the probability of seeing the word *barked* after the seeing sequence *the lazy dog*?”

Language models formally

Sequence of words $w_{1:n} = w_1 w_2 w_3 \dots w_n$ estimate

$$\Pr(w_{1:n}) = \Pr(w_1, w_2, \dots, w_n)$$

Note: We misuse notation and usually omit the RVs

$$\Pr(W_1 = w_1, W_2 = w_2, \dots, W_n = w_n)$$

We *factorize* the joint probability into a product

- One factorization is very useful: left-to-right

$$\begin{aligned} \Pr(w_{1:n}) &= \Pr(w_1 | \langle S \rangle) \Pr(w_2 | \langle S \rangle, w_1) \Pr(w_3 | \langle S \rangle, w_1, w_2) \dots \\ &\quad \dots \Pr(w_n | \langle S \rangle, w_1, w_2, \dots, w_{n-1}) \end{aligned}$$

Simplifications in 'classical' language models

Despite factorization, the last term of $\Pr(w_{1:n}) = \Pr(w_1|\langle s \rangle) \Pr(w_2|\langle s \rangle, w_1) \Pr(w_3|\langle s \rangle, w_1, w_2) \cdots \Pr(w_k|\langle s \rangle, w_1, w_2, \dots, w_{n-1})$ still depends on all the previous words of the sequence

k -th order markov-assumption

The next word depends only on the last k words

$$\Pr(w_i|w_{1:i-1}) \approx \Pr(w_i|w_{i-k:i-1}) \quad (\text{inclusive indexing!})$$

$\langle s \rangle_0$ The₁ cat₂ sat₃ on₄ the₅ w_6

$$i = 6, k = 2 \rightarrow \Pr(w_i|w_{1:i-1}) \approx \Pr(w_6|W_4 = \text{on}, W_5 = \text{the})$$

Estimating probabilities in ‘classical’ language models

Maximum Likelihood Estimation (aka. counting and dividing)

$$\hat{P}_{\text{MLE}}(W_i = w | w_{i-k:i-1}) = \frac{\#(w_{i-k} \quad w_{i-k+1} \quad \dots \quad w_{i-1} \quad w)}{\#(w_{i-k} \quad w_{i-k+1} \quad \dots \quad w_{i-1})}$$

Shortcomings of n -gram language models

Y. Goldberg (2017). **Neural Network Methods for Natural Language Processing**. Morgan & Claypool, p. 108

Long-range dependencies

- To capture a dependency between the next word and the word 10 positions in the past, we need to see a relevant 11-gram in the text

Lack of generalization across contexts

- Having observed *black car* and *blue car* does not influence our estimates of the event *red car* if we haven't see it before

Generating text with language models

We can generate (“sample”) random sentences from the model according to their probability

- 1 Predict a probability distribution over the vocabulary conditioned on the start symbol $\langle s \rangle$
- 2 Draw the first word from the predicted distribution
- 3 Predict a probability distribution over the vocabulary conditioned on the start symbol and the first word
- 4 Draw a the second word from to the predicted distribution
- 5 Repeat until generated *end-of-sequence* symbol $\langle /s \rangle$ (or $\langle \text{EOS} \rangle$)

Language models

Neural language models

Basic building block: Multi-layer perceptron (MLP)

$$f(x) = \sigma \left(g \left(x W^1 + b^1 \right) W^2 + b^2 \right)$$

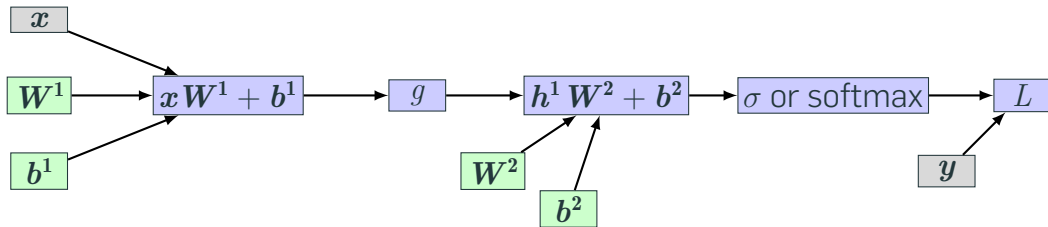


Figure 1: Computational graph; green boxes are trainable parameters, gray are constant inputs

Neural LMs

Let's build a neural network

- Input: a k -gram of words $w_{1:k}$
- Desired output: a probability distribution over the vocabulary V for the next word w_{k+1}

Training neural LMs

Where to get training examples?

Training examples are simply word k -grams from an unlabeled corpus

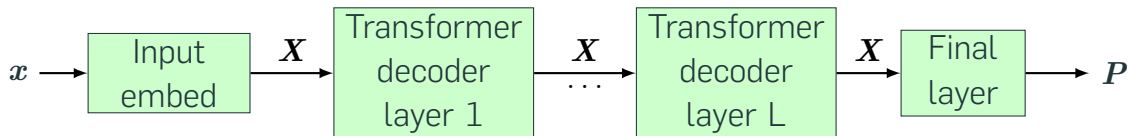
- Identities of the first $k - 1$ words are used as features
- The last word is used as the target label for the classification

The model is trained using cross-entropy loss

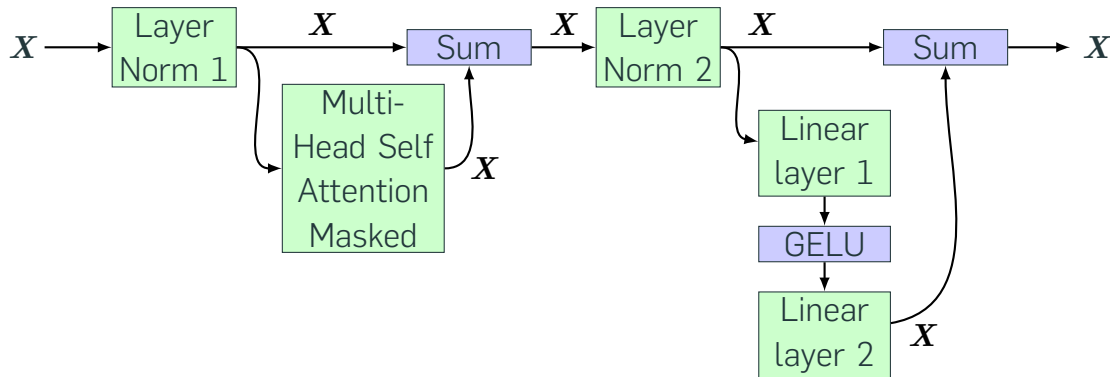
Large Neural Language Models

- 1 Motivation
- 2 What is a language model?
- 3 Language models
- 4 Large Neural Language Models**
- 5 Fully open models
- 6 Want to learn more?

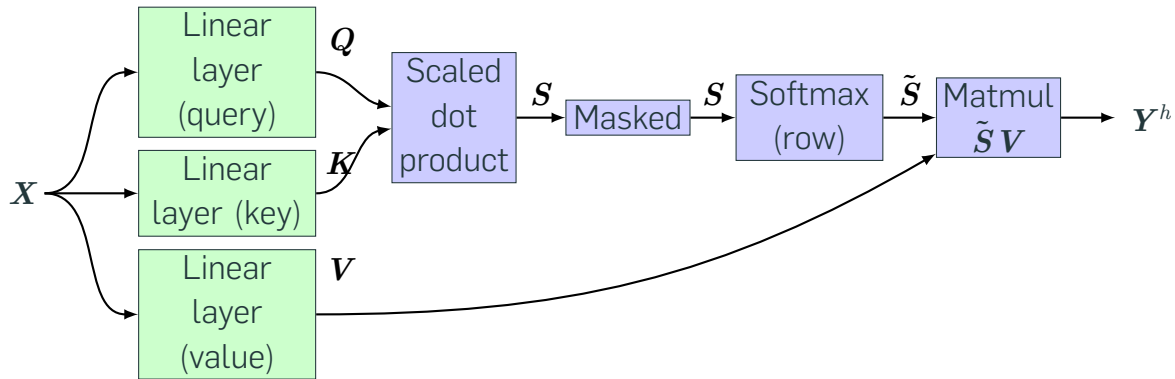
Transformer decoder (GPT)



GPT (decoding-only transformer, transformer layer)



Single masked self-attention head (GPT)



Large Neural Language Models

Every task is a text-to-text task

"The basic idea underlying our work is to treat every text processing problem as a "text-to-text" problem, i.e. taking text as input and producing new text as output."

Journal of Machine Learning Research 21 (2020) 1-67

Submitted 1/20; Revised 6/20; Published 6/20

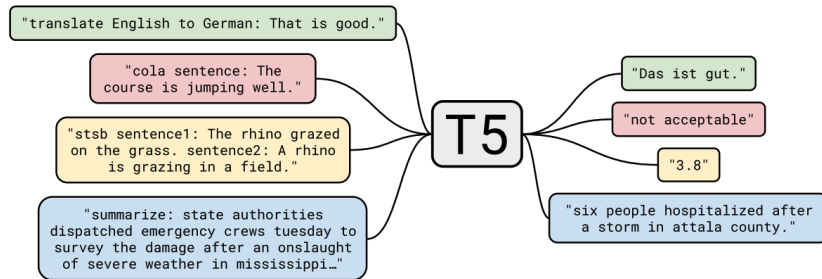
C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"**.
In: *Journal of Machine Learning Research* 21.140, pp. 1-67

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel*	CRAFFEL@GMAIL.COM
Noam Shazeer*	NOAM@GOOGLE.COM
Adam Roberts*	ADAROB@GOOGLE.COM
Katherine Lee*	KATHERINELEE@GOOGLE.COM
Sharan Narang	SHARANNARANG@GOOGLE.COM
Michael Matena	MMATENA@GOOGLE.COM
Yanqi Zhou	YANQIZ@GOOGLE.COM
Wei Li	MWEILI@GOOGLE.COM
Peter J. Liu	PETERJLIU@GOOGLE.COM

Google, Mountain View, CA 94043, USA

Editor: Ivan Titov



C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **“Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”**. In: *Journal of Machine Learning Research* 21.140, pp. 1–67

Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. “T5” refers to our model, which we dub the “**Text-to-Text Transfer Transformer**”.

T5 – Source data quality matters

"Common Crawl is a publicly-available web archive that provides "web extracted text" by removing markup and other non-text content from the scraped HTML files. This process produces around 20TB of scraped text data each month. Unfortunately, the majority of the resulting text is not natural language."

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"**.
In: *Journal of Machine Learning Research* 21.140, pp. 1–67

T5 – Colossal Clean Common Crawl corpus (about 750 GB)

language, placeholder text, source code, etc.). To address these issues, we used the following heuristics for cleaning up Common Crawl's web extracted text:

- We only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark).
- We discarded any page with fewer than 5 sentences and only retained lines that contained at least 3 words.
- We removed any page that contained any word on the “List of Dirty, Naughty, Obscene or Otherwise Bad Words”.⁶
- Many of the scraped pages contained warnings stating that Javascript should be enabled so we removed any line with the word Javascript.
- Some pages had placeholder “lorem ipsum” text; we removed any page where the phrase “lorem ipsum” appeared.
- Some pages inadvertently contained code. Since the curly bracket “{” appears in many programming languages (such as Javascript, widely used on the web) but not in natural text, we removed any pages that contained a curly bracket.
- To deduplicate the data set, we discarded all but one of any three-sentence span occurring more than once in the data set.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **“Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”**. In: *Journal of Machine Learning Research* 21.140, pp. 1–67

T5 – Scale matters the most

"scaling the model size to 11 billion parameters was the most important ingredient for achieving our best performance."

Model	GLUE Average	CoLA Matthew's	SST-2 Accuracy	MRPC F1	MRPC Accuracy	STS-B Pearson	STS-B Spearman
Previous best	89.4 ^a	69.2 ^b	97.1 ^a	93.6^b	91.5^b	92.7 ^b	92.3 ^b
T5-Small	77.4	41.0	91.8	89.7	86.6	85.6	85.0
T5-Base	82.7	51.1	95.2	90.7	87.5	89.4	88.6
T5-Large	86.4	61.2	96.3	92.4	89.9	89.9	89.2
T5-3B	88.5	67.1	97.4	92.5	90.0	90.6	89.8
T5-11B	90.3	71.6	97.5	92.8	90.4	93.1	92.8

Model	QQP F1	QQP Accuracy	MNLI-m Accuracy	MNLI-mm Accuracy	QNLI Accuracy	RTE Accuracy	WNLI Accuracy
Previous best	74.8 ^c	90.7^b	91.3 ^a	91.0 ^a	99.2^a	89.2 ^a	91.8 ^a
T5-Small	70.0	88.0	82.4	82.3	90.3	69.9	69.2
T5-Base	72.6	89.4	87.1	86.2	93.7	80.1	78.8
T5-Large	73.9	89.9	89.9	89.6	94.8	87.2	85.6
T5-3B	74.4	89.7	91.4	91.2	96.3	91.1	89.7
T5-11B	75.1	90.6	92.2	91.9	96.9	92.8	94.5

Model	SQuAD EM	SQuAD F1	SuperGLUE Average	BoolQ Accuracy	CB F1	CB Accuracy	COPA Accuracy
Previous best	90.1 ^a	95.5 ^a	84.6 ^d	87.1 ^d	90.5 ^d	95.2 ^d	90.6 ^d
T5-Small	79.10	87.24	63.3	76.4	56.9	81.6	46.0
T5-Base	85.44	92.08	76.2	81.4	86.2	94.0	71.2
T5-Large	86.66	93.79	82.3	85.4	91.6	94.8	83.4
T5-3B	88.53	94.95	86.4	89.9	90.3	94.4	92.0
T5-11B	91.26	96.22	88.9	91.2	93.9	96.8	94.8

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"**.
In: *Journal of Machine Learning Research* 21.140, pp. 1–67

Fully open models

- 1 Motivation
- 2 What is a language model?
- 3 Language models
- 4 Large Neural Language Models
- 5 Fully open models**
- 6 Want to learn more?

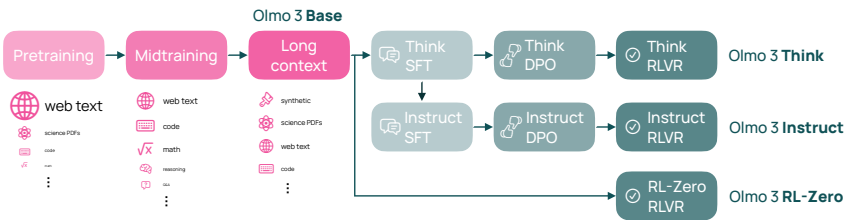
Fully open LLMs in 2026?

"We introduce Olmo 3, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. Olmo 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire model flow, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, Olmo 3.1 Think 32B, is the strongest fully-open thinking model released to-date."

A. Ettinger et al. (2025). **"Olmo 3"**. In:
arXiv preprint

Training LLMs is complex...

A. Ettinger et al. (2025). **"Olmo 3"**. In:
arXiv preprint



...and very costly

A. Ettinger et al. (2025). **"Olmo 3"**. In:
arXiv preprint

"Pretraining: ~47 days (including midtraining and long-context stages) The initial pretraining phase on 5.5T tokens took about 9.5 days on 512 GPUs, followed by an additional 35 days on 1024 GPUs."

Meanwhile in Europe...

The screenshot shows a web browser window with the URL www.openeurolim.eu/deliverables. The page features a dark background with a vertical timeline on the left. The timeline has three circular markers, each containing the date '31 JUL' and a year. The first marker is for '2025' and is partially obscured. The second marker is for '2025' and has a green checkmark next to it. The third marker is for '2026'. To the right of the timeline, there are three main sections, each with a title in a rounded rectangle and a description below it. The first section is titled 'INITIAL CATALOGUE AND ANALYTICS REPORTS FOR EXISTING TRAINING DATASETS' and describes a catalogue of training datasets per language along with automatic analytics reports. The second section is titled 'COMMUNICATION, DISSEMINATION AND EXPLOITATION STRATEGY' and describes a full communication and dissemination and exploitation strategy and schedule of activities to improve project duration. The third section is titled 'EVALUATION CODE PACKAGE'. A 'DELIVERABLE' button with a download icon is visible next to the second section.

www.openeurolim.eu/deliverables

31 JUL 2025

INITIAL CATALOGUE AND ANALYTICS REPORTS FOR EXISTING TRAINING DATASETS

A catalogue of training datasets per language along with automatic analytics reports.

31 JUL 2025 ✓

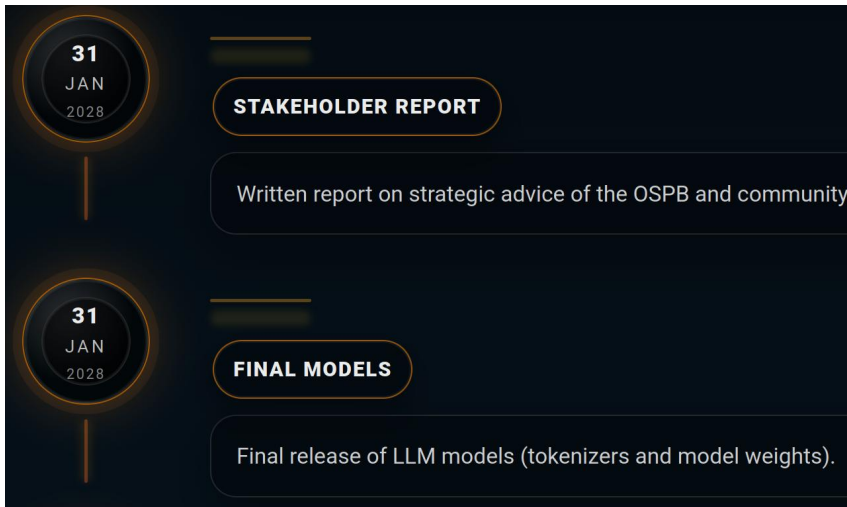
COMMUNICATION, DISSEMINATION AND EXPLOITATION STRATEGY **↓ DELIVERABLE**

Full communication and dissemination and exploitation strategy and schedule of activities to improve project duration.

31 JUL 2026

EVALUATION CODE PACKAGE

...things are moving sloooooow....



Want to learn more?

- 1 Motivation
- 2 What is a language model?
- 3 Language models
- 4 Large Neural Language Models
- 5 Fully open models
- 6 Want to learn more?**

Want to learn more?

Take the **Natural Language Processing with Deep Learning** course

- Winter term
- BSc, suggested in 3rd year
- Prerequisites
 - Decent math
 - Good programming skills in Python

License and credits

Licensed under Creative Commons
Attribution-ShareAlike 4.0 International
(CC BY-SA 4.0)



Credits

Ivan Habernal

Content from ACL Anthology papers licensed under CC-BY
<https://www.aclweb.org/anthology>