A Theoretical and Practical Introduction to Neural Language Models:
Evaluating and Exploring their Linguistic Abilities

Autumn School in AI | Digital Humanities PhD, November 11 2025

Alessio Miaschi
ItaliaNLP Lab, Istituto di Linguistica Computazionale (CNR-ILC), Pisa
alessio.miaschi@ilc.cnr.it
https://alemiaschi.github.io/

Cristiano Ciaccio
Università di Pisa
ItaliaNLP Lab, Istituto di Linguistica Computazionale (CNR-ILC), Pisa
cristiano.ciaccio@ilc.cnr.it
https://www.ilc.cnr.it/people/cristiano-ciaccio/

About me and...

I am a full-time researcher (RTD) at the ItaliaNLP Lab, Institute for Computational Linguistics “A. Zampolli” (CNR-ILC, Pisa). In 2022, I received my PhD in Computer Science at the University of Pisa.

My research interests lie primarily in the context of Natural Language Processing (NLP) and in the study of Language Models (LM). I am particularly interested in the interpretability of large-scale LMs and in the evaluation of their internal representations, with a specific emphasis on understanding their inner linguistic abilities.

About me and... the team!

I am a full-time researcher (RTD) at the ItaliaNLP Lab, Institute for Computational Linguistics "A. Zampolli" (CNR-ILC, Pisa). In 2022, I received my PhD in Computer Science at the University of Pisa.

My research interests lie primarily in the context of Natural Language Processing (NLP) and in the study of Language Models (LM). I am particularly interested in the interpretability of large-scale LMs and in the evaluation of their internal representations, with a specific emphasis on understanding their inner linguistic abilities.

Istituto di Linguistica Computazionale "Antonio Zampolli"

Consiglio Nazionale delle Ricerche

The **ItaliaNLP Lab** (**CNR-ILC**) gathers researchers, postdocs and students from computational linguistics, computer science and linguistics who work on developing resources and algorithms for processing and understanding human languages.

Permanent Researchers:
- Felice Dell'Orletta
- Simonetta Montemagni
- Dominique Brunato
- Franco Alberto Cardillo
- Giulia Venturi
- Giulia Benotto

Researchers (TD):
- Chiara Alzetta
- Alessio Miaschi

Research Fellows:
- Agnese Bonfigli
- Chiara Fazzone
- Ruben Piperno

PhD Students:
- Cristiano Ciaccio
- Luca Dini
- Lucia Domenichelli
- Michele Papucci
- Marta Sartor

+ Master/Undergraduate/Visiting Students

Link to website: http://www.italianlp.it/

About me and… the team!

I am a full-time researcher (RTD) at the ItaliaNLP Lab, Institute for Computational Linguistics "A. Zampolli" (CNR-ILC, Pisa). In 2022, I received my PhD in Computer Science at the University of Pisa.

My research interests lie primarily in the context of Natural Language Processing (NLP) and in the study of Language Models (LM). I am particularly interested in the interpretability of large-scale LMs and in the evaluation of their internal representations, with a specific emphasis on understanding their inner linguistic abilities.

Istituto di Linguistica Computazionale "Antonio Zampolli"

Consiglio Nazionale delle Ricerche

The **ItaliaNLP Lab** (**CNR-ILC**) gathers researchers, postdocs and students from computational linguistics, computer science and linguistics who work on developing resources and algorithms for processing and understanding human languages.

Permanent Researchers:
- Felice Dell'Orletta
- Simonetta Montemagni
- Dominique Brunato
- Franco Alberto Cardillo
- Giulia Venturi
- Giulia Benotto

Researchers (TD):
- Chiara Alzetta
- Alessio Miaschi

Research Fellows:
- Agnese Bonfigli
- Chiara Fazzone
- Ruben Piperno

PhD Students:
- **Cristiano Ciaccio**
- Luca Dini
- Lucia Domenichelli
- Michele Papucci
- Marta Sartor

+ Master/Undergraduate/Visiting Students

Link to website: http://www.italianlp.it/

Materiali

Github Repository: https://github.com/alemiaschi/introduction_NLMs_Autumn_School_AI

Outline

Part I:

1. An introduction to Language Models (LMs)
2. Neural Language Models (NLMs)
3. Transformer-based LMs

Part II:

4. Interpreting and Evaluating NLMs

Part I

An introduction to Language Models (LMs)

Language Models

- In the context of numerous studies in Computational Linguistics (CL) and Natural Language Processing (NLP), it is assumed that language can be viewed as a *probabilistic system*

- To describe and explain the functioning of a probabilistic system, it is necessary to define a *(probabilistic) model*

- A **language model**, therefore, is nothing more than a system capable of assigning a probability to word sequences

Probabilistic Language Models

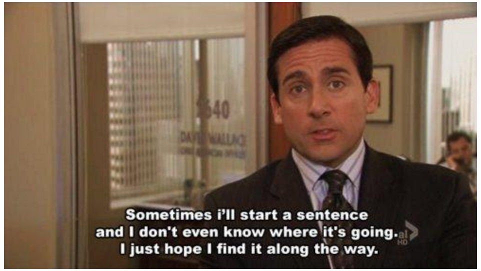
- Given a sequence of words $w_1, \ldots, w_n$, we can represent the sequence as:

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2|w_1)\ldots p(w_n|w_1, \ldots, w_{n-1})$$

Probabilistic Language Models

- Given a sequence of words $w_1, \ldots, w_n$, we can represent the sequence as:

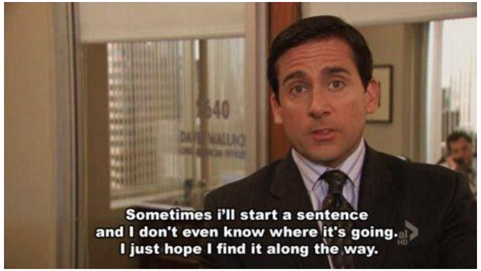
$$p(w_1, \ldots, w_n) = p(w_1)p(w_2|w_1)\ldots p(w_n|w_1, \ldots, w_{n-1})$$



Probabilistic Language Models

- Given a sequence of words $w_1, \ldots, w_n$, we can represent the sequence as:

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2|w_1)\ldots p(w_n|w_1, \ldots, w_{n-1})$$

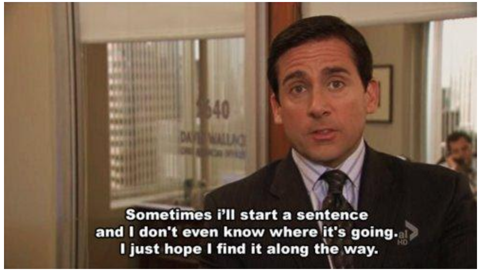


- As a consequence, the probability of the next word in a sequence given the preceding context can be defined as:

$$p(w_n|w_1, \ldots, w_{n-1}) = \frac{Count(w_1, \ldots, w_{n-1}, w_n)}{Count(w_1, \ldots, w_{n-1})}$$

Probabilistic Language Models

- Given a sequence of words $w_1, \ldots, w_n$, we can represent the sequence as:



$$p(w_1, \ldots, w_n) = p(w_1)p(w_2|w_1)\ldots p(w_n|w_1, \ldots, w_{n-1})$$

- As a consequence, the probability of the next word in a sequence given the preceding context can be defined as:

$$P(the|its\ water\ is\ so\ transparent\ that) = \frac{C(its\ water\ is\ so\ transparent\ that\ the)}{C(its\ water\ is\ so\ transparent\ that)}$$

Probabilistic Language Models (ngrams)

- N-grams LMs can be exploited to approximate the probability of the next word as follows:

$$p(w_i|w_1, ..., w_{t-1}) \approx p(w_i|w_{i-N}, ..., w_{i-1})$$

Probabilistic Language Models (ngrams)

- N-grams LMs can be exploited to approximate the probability of the next word as follows:

$$p(w_i|w_1, ..., w_{t-1}) \approx p(w_i|w_{i-N}, ..., w_{i-1})$$

- As *N* increases, the approximation becomes more accurate, but the complexity grows exponentially.
- Conversely, when *N=1*, the model requires less information, but its performance is significantly lower.

Probabilistic Language Models (ngrams)

Before

P(I saw a cat on a mat) =

P(I)
· P(saw | I)
· P(a | I saw)
· P(cat | I saw a)
· P(on | I saw a cat)
· P(a | I saw a cat on)
· P(mat | I saw a cat on a)

After (3-gram)

P(I saw a cat on a mat) =

P(I) → P(I)
· P(saw | I) → · P(saw | I)
· P(a | I saw) → · P(a | I saw)
· P(cat | ~~I~~ saw a) → · P(cat | saw a)
· P(on | ~~I saw~~ a cat) → · P(on | a cat)
· P(a | ~~I saw a~~ cat on) → · P(a | cat on)
· P(mat | ~~I saw a cat~~ on a) → · P(mat | on a)

ignore use

Language Modeling, NLP Course, Lena Voita,
https://lena-voita.github.io/nlp_course/language_modeling.html

Probabilistic Language Models (ngrams)

- N-gram-based language models, however, have several limitations:
  - Regardless of the value assigned to *N*, the model will always be an approximation of the true probability distribution.
  - Due to the exponential growth in complexity, the choice of *N* will always fall on particularly low values (usually 2 or 3).
  - An N-gram model cannot **generalize to new word sequences**.

Word representations

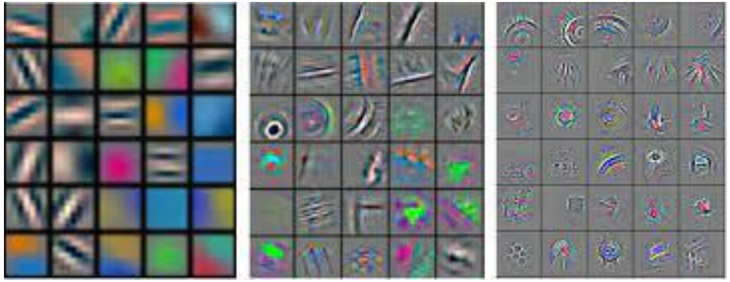
- Words can be considered the basic units of a language model

- To understand a language, it is first necessary to know the meaning of the words that compose it

- To comprehend a language, a (computational) language model should be able to *represent* the words of that language

A *representation* problem

- *Representation learning* is a central problem in the context of Artificial Intelligence, neuroscience, and semantics

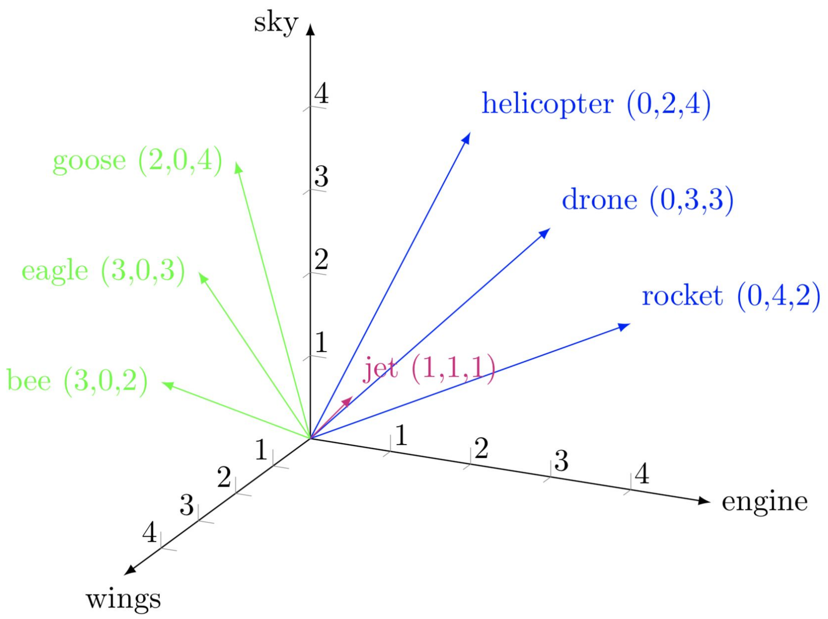


representation

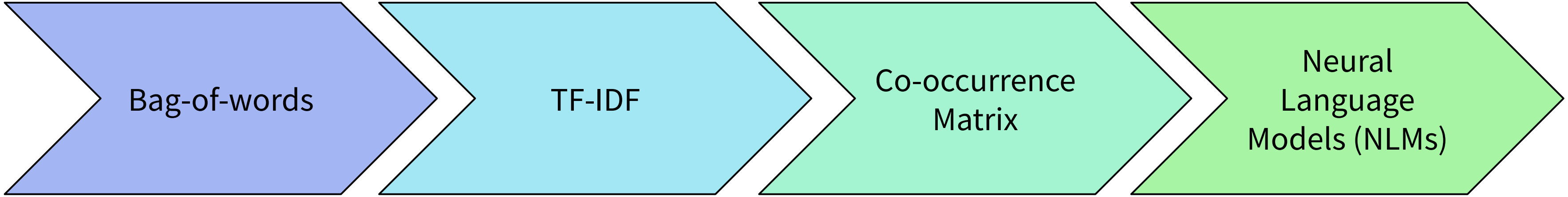
“monkey”

Word representations

- From a computational perspective, the most intuitive method to represent a word is to associate it with a **vector of numbers**



Word representations



Neural Language Model (NLM)

Neural Networks

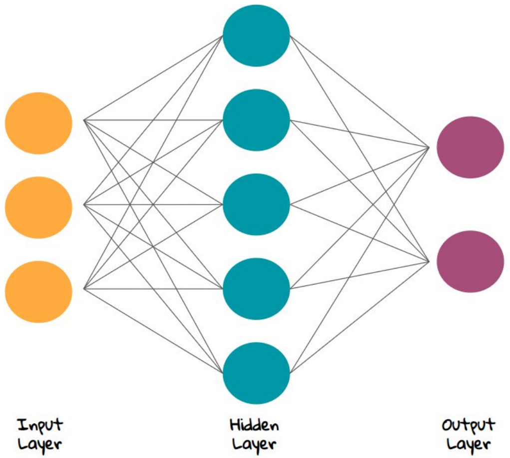
- In the context of machine learning, a neural network (NN) is a computational model composed of connected units called *artificial neurons*

Neural Networks

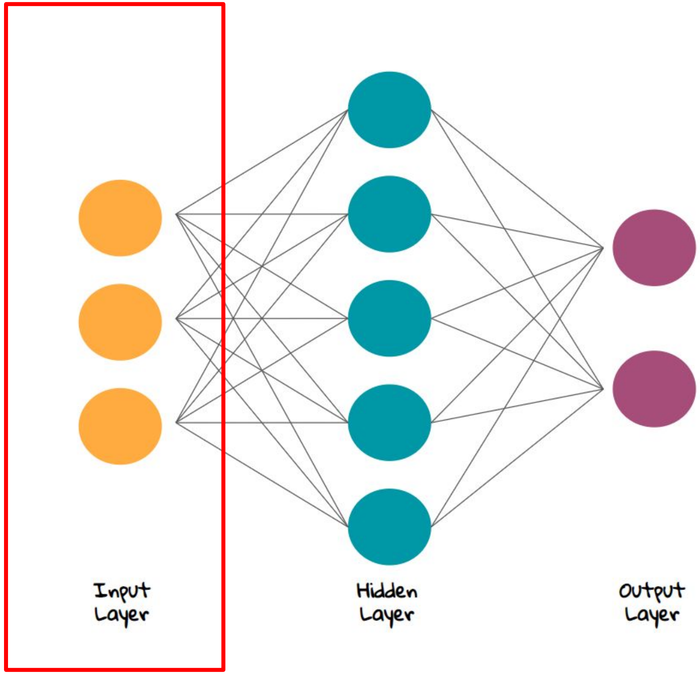
- In the context of machine learning, a neural network (NN) is a computational model composed of connected units called *artificial neurons*

A NN is composed of:

- an *input layer*
- one (or more) *hidden layers*
- an *output layer*

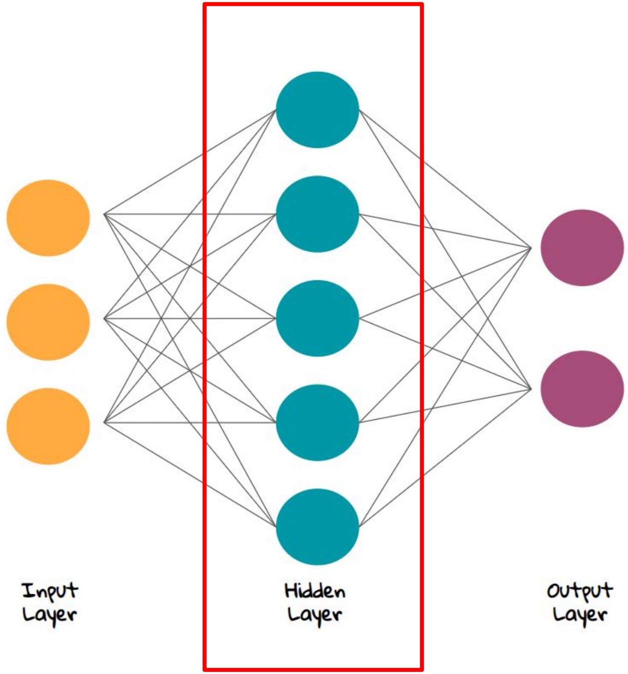


Neural Networks



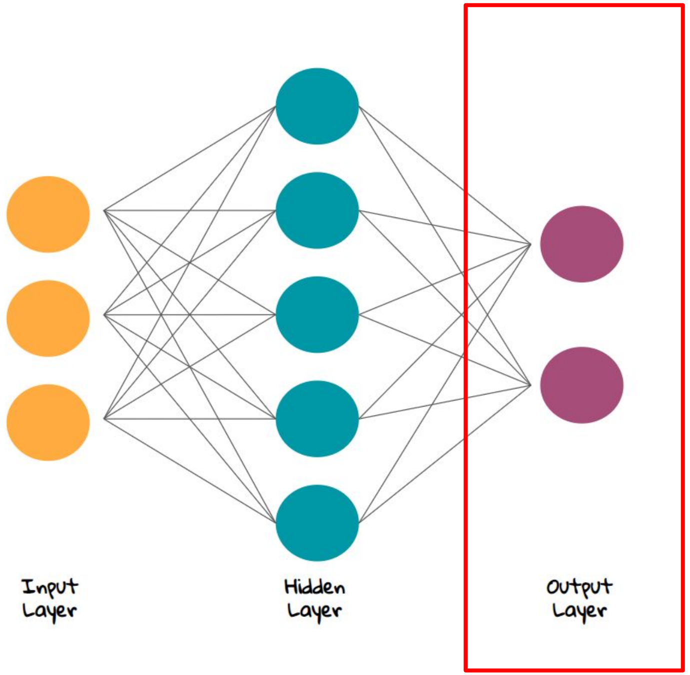
- Input data, e.g. images, words, etc

Neural Networks



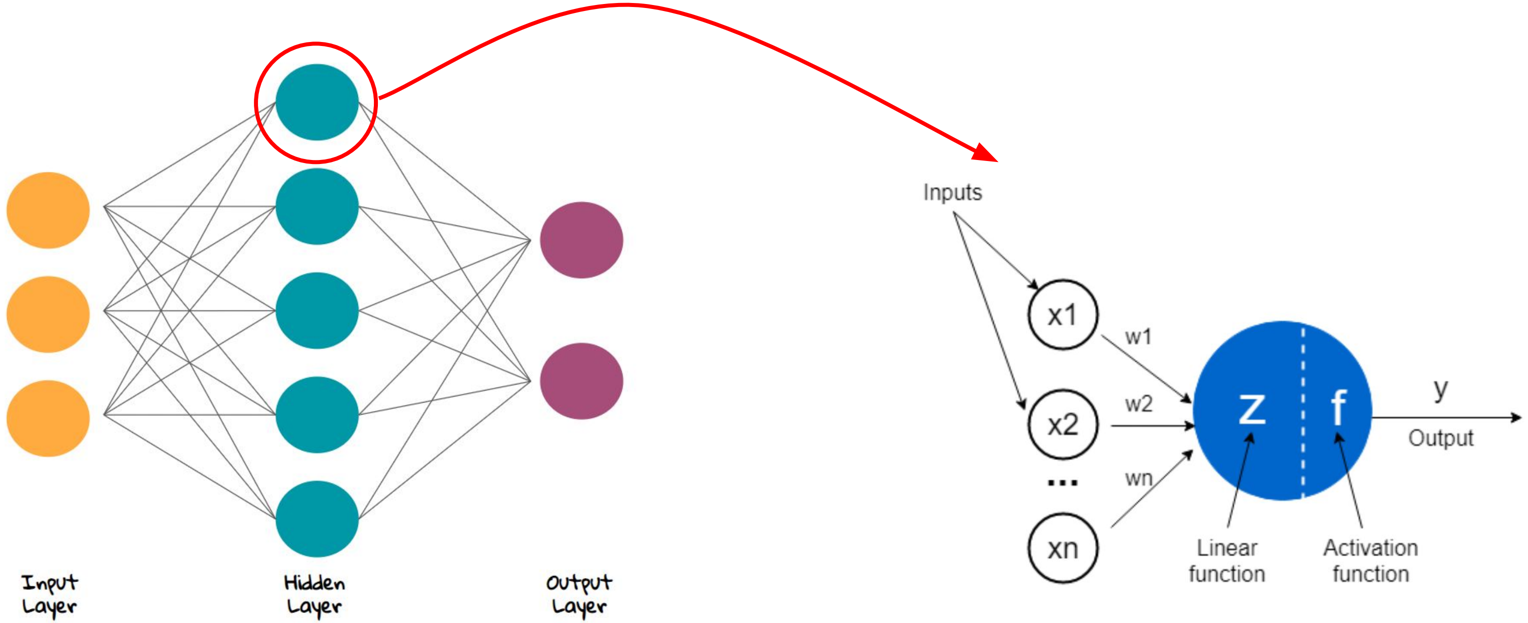
- Internal representations

Neural Networks

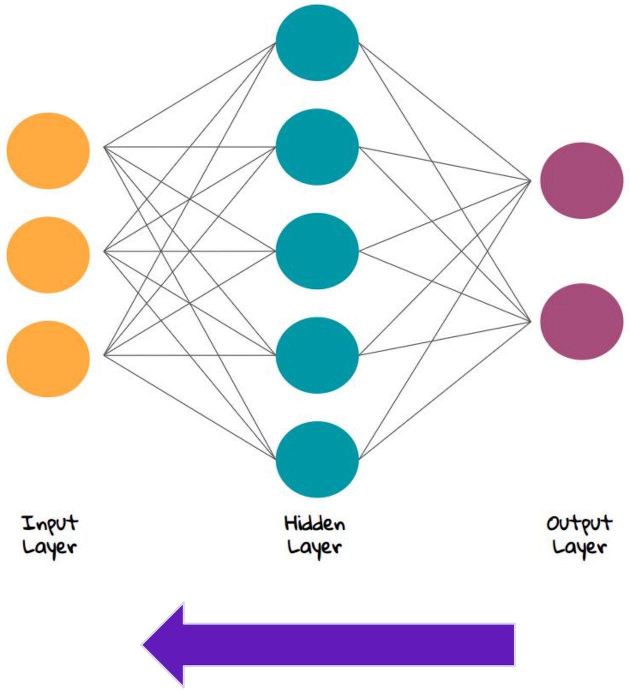


- Output, e.g. whether the image contains a cat

Neural Networks

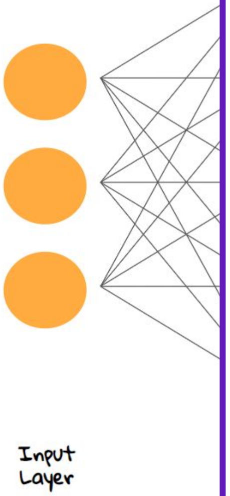


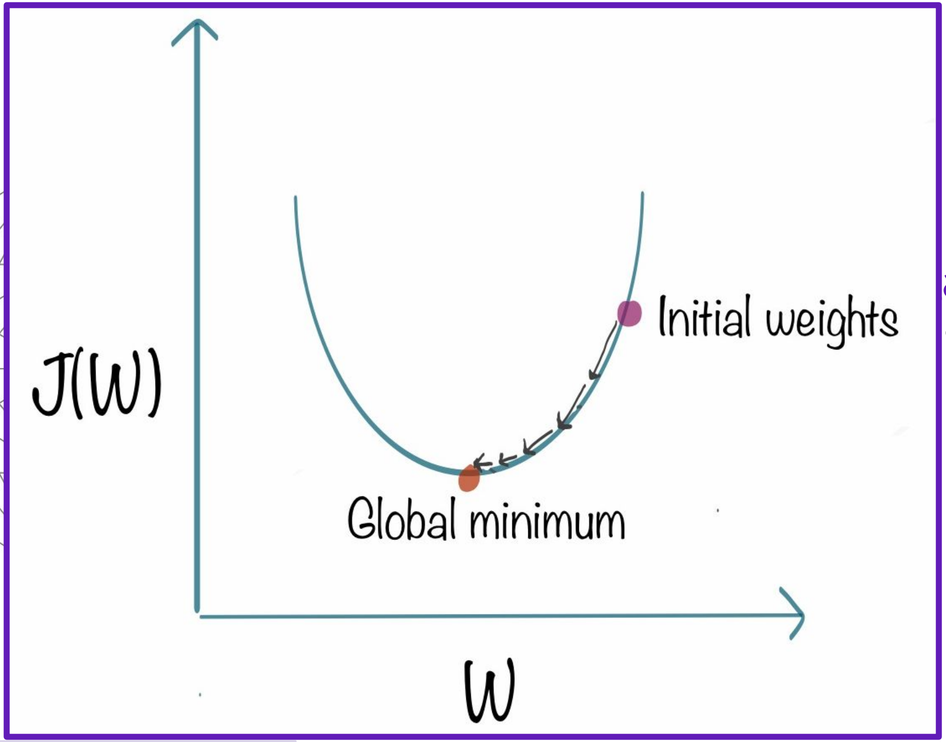
Neural Networks



- **Backpropagation**: training algorithm of the NN

Neural Networks





ation: training
the NN

Neural Language Model (NLM)

- A NLM is a Neural Network (NN) trained to approximate the **language modeling** function

Neural Language Model (NLM)

- A NLM is a Neural Network (NN) trained to approximate the **language modeling** function

- A probabilistic LM defines the probability of a sequence $s = [w_1, w_2, \ldots, w_n]$ as:

$$P(s) = \prod_{i=1}^{N} P(w_i | w_1, w_2, ..., w_{i-1})$$

Neural Language Model (NLM)

- A NLM is a Neural Network (NN) trained to approximate the **language modeling** function

- A probabilistic LM defines the probability of a sequence $s = [w_1, w_2, \dots, w_n]$ as:

$$P(s) = \prod_{i=1}^{N} P(w_i | w_1, w_2, \dots, w_{i-1})$$

- Bengio et al. (2003) proposed a model that approximate the LM function relying on the architecture of a NN → **Neural Probabilistic Language Model**

Neural Language Model (NLM)

Input		**Model**		**Output**
Mi piace la focaccia _	→	(Neural) Language Model	→	genovese 60% barese 35% unta 3%

Neural Language Model (NLM)

Input

Mi piace la focaccia _

Model

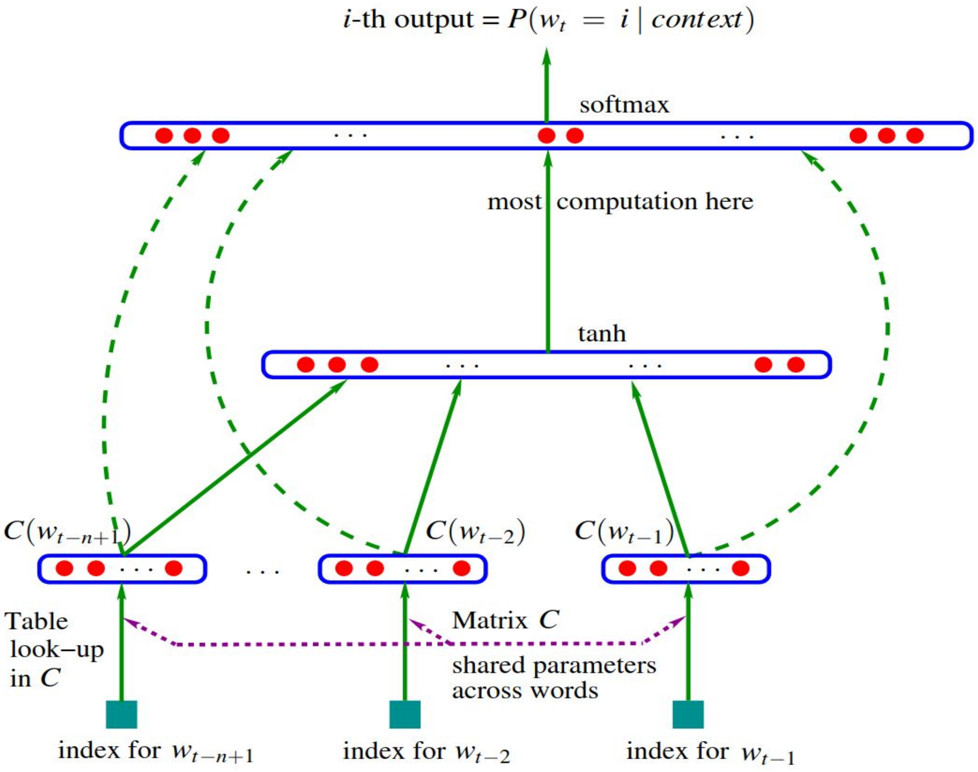
(Neural)
Language Model

Output

genovese	barese	unta
60%	35%	3%

|V|

Neural Language Model (NLM)



Transformer Model

Transformer

- The most widely used architecture nowadays is the **Transformer**, first introduced in: Attention is All you Need (Vaswani et al., 2017)

Transformer

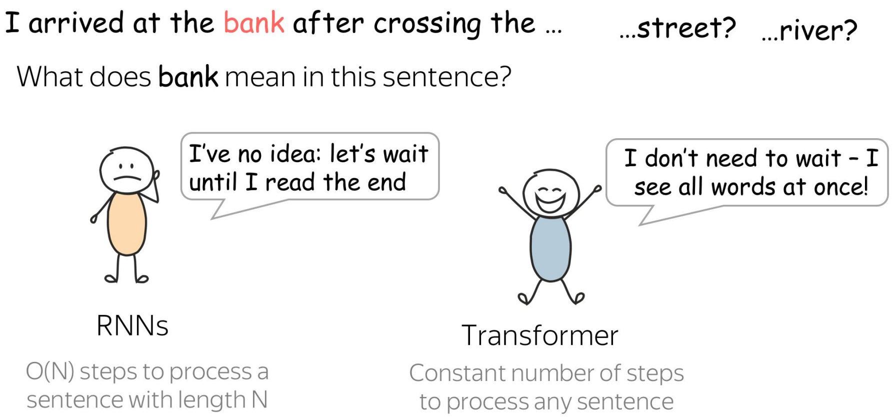
- The most widely used architecture nowadays is the **Transformer**, first introduced in: Attention is All you Need (Vaswani et al., 2017)

- The Transformer is a neural network (Encoder-Decoder) that leverages a specific mechanism, **Attention**, to focus on key portions of a sentence and create contextual word representations.

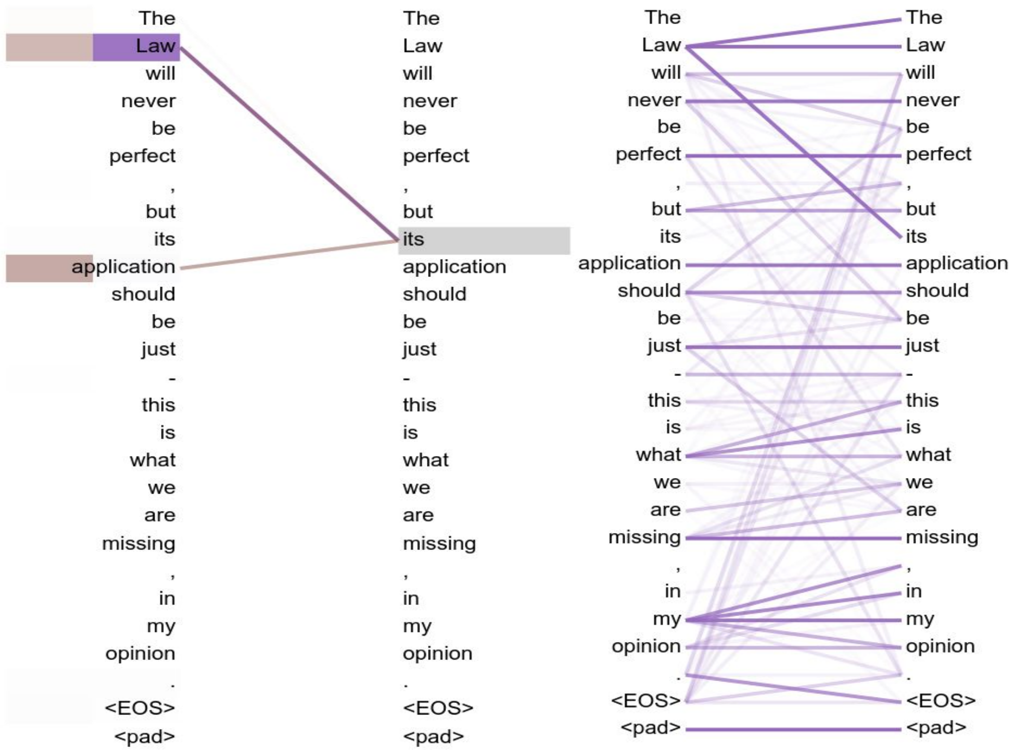
Transformer

- The most widely used architecture nowadays is the **Transformer**, first introduced in: [Attention is All you Need (Vaswani et al., 2017)]

- The Transformer is a neural network (Encoder-Decoder) that leverages a specific mechanism, **Attention**, to focus on key portions of a sentence and create contextual word representations.

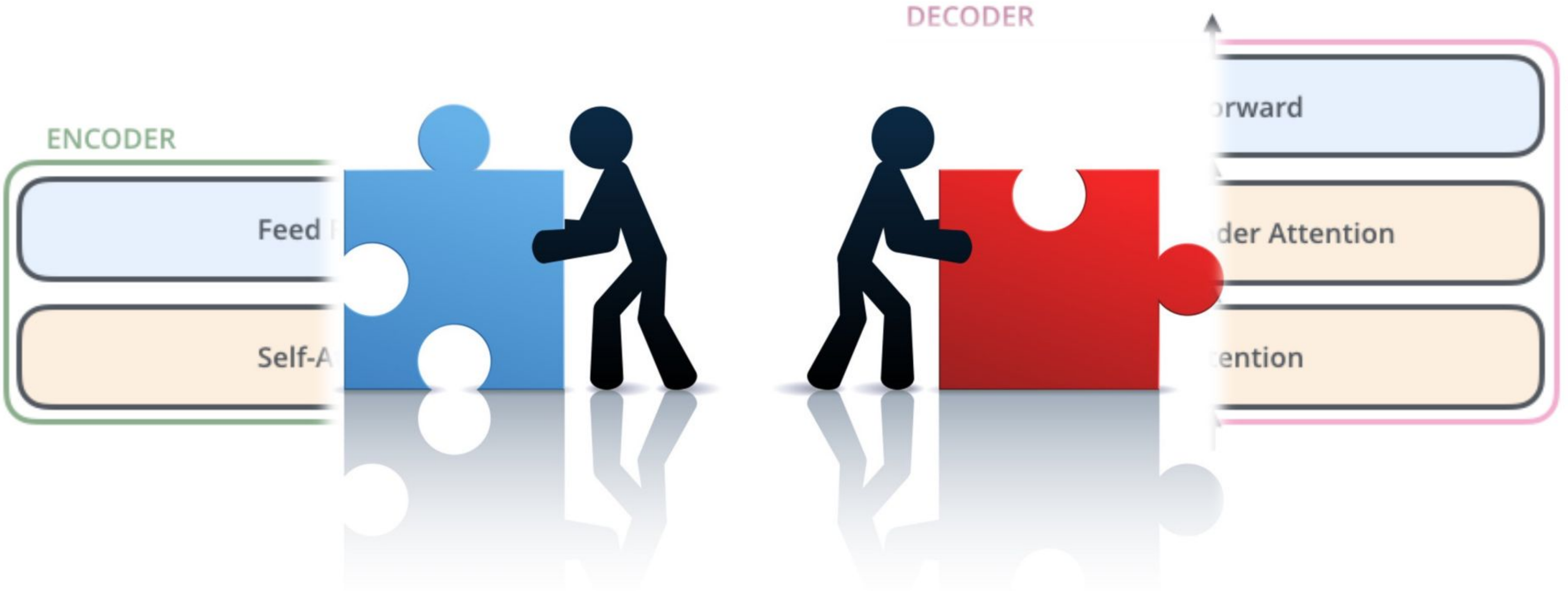


Transformer - Attention



Transformer-based NLMs

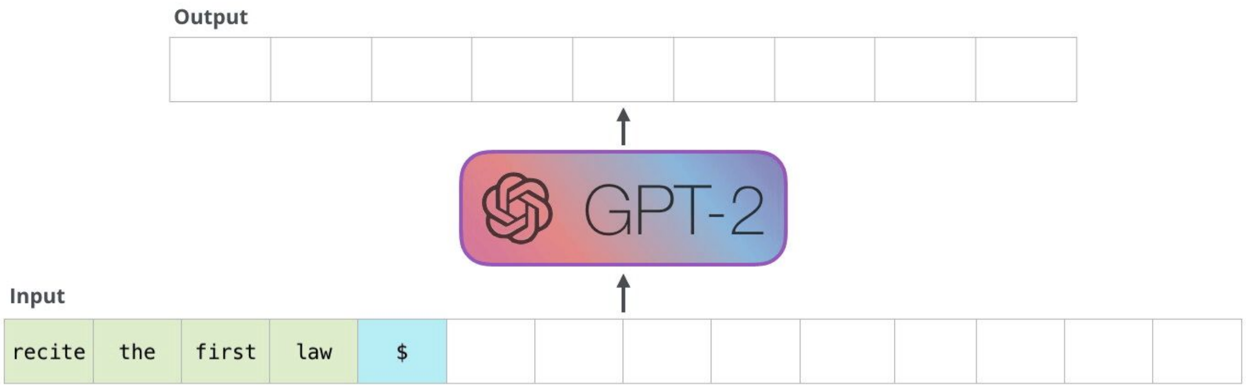
Transformer-based NLMs



Source: Large Language Models and How To Instruct Them (CLiC-it 2023 Tutorial)

GPT (Radford et al, 2018), GPT-2 (Radford et al, 2019), etc

- Decoder Transformer model
- Trained on the **Language Modeling** (**LM**) task
- Generative model



Improving Language Understanding by Generative Pre-Training (Radford et al., 2018), https://openai.com/research/language-unsupervised

Language Models are Unsupervised Multitask Learners (Radford et al., 2019), https://openai.com/research/better-language-models

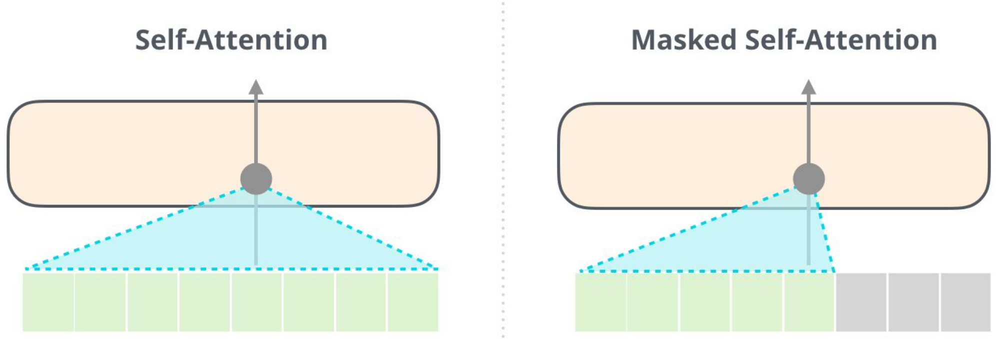
GPT (Radford et al, 2018), GPT-2 (Radford et al, 2019), etc

- Decoder Transformer model
- Trained or
- Generative

Inp

re

Masked/Causal Self-Attention



[Improving Language Understanding by Generative Pre-Training (Radford et al., 2018), https://openai.com/research/language-unsupervised](https://openai.com/research/language-unsupervised)

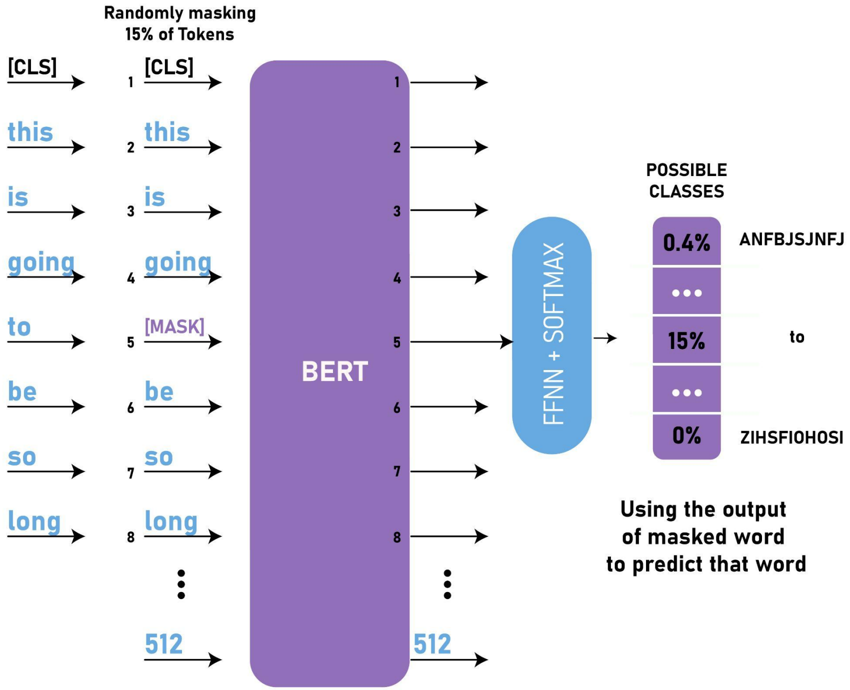
[Language Models are Unsupervised Multitask Learners (Radford et al., 2019), https://openai.com/research/better-language-models](https://openai.com/research/better-language-models)

BERT (Devlin et al., 2019) 🤓

- Encoder Transformer model (12/24 layers)

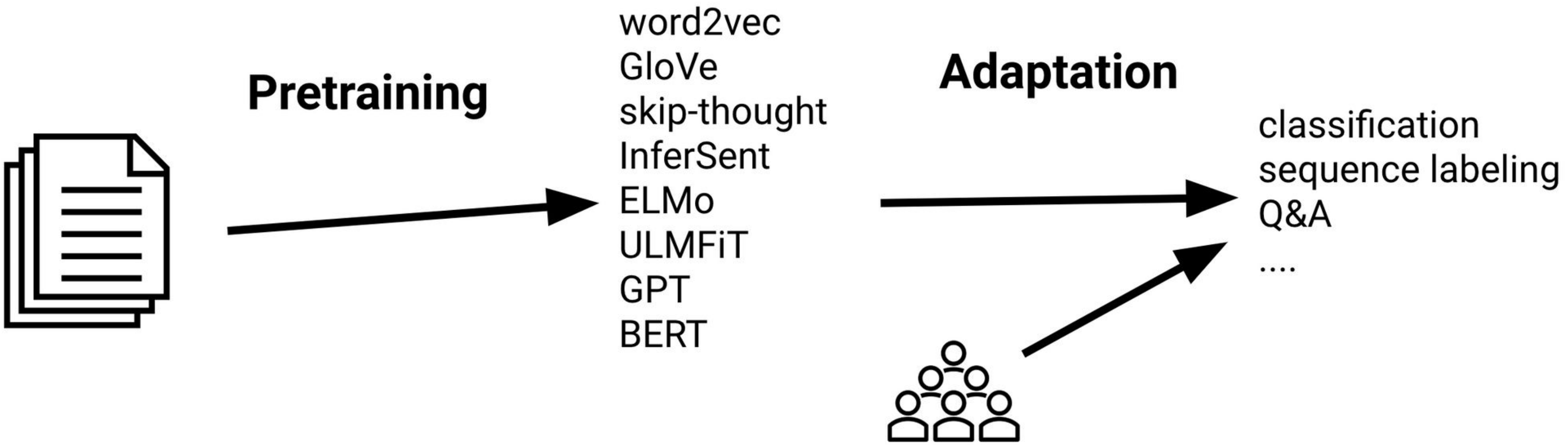
- Trained on the **Masked Language Modeling** (**MLM**)

- The model can be further trained (fine-tuning) for solving different NLP tasks:
    - Sentiment analysis;
    - Question answering;
    - Textual entailment;
    - etc.



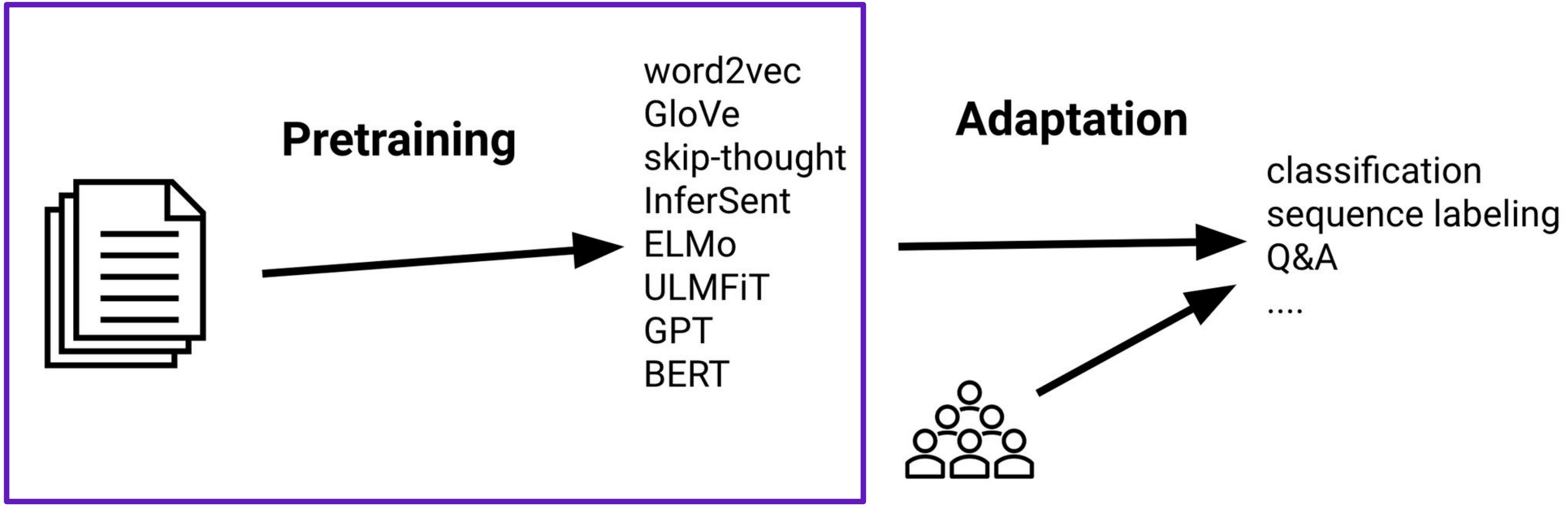
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2019), https://aclanthology.org/N19-1423.pdf

Transfer Learning



The State of Transfer Learning in NLP: https://ruder.io/state-of-transfer-learning-in-nlp/

Transfer Learning



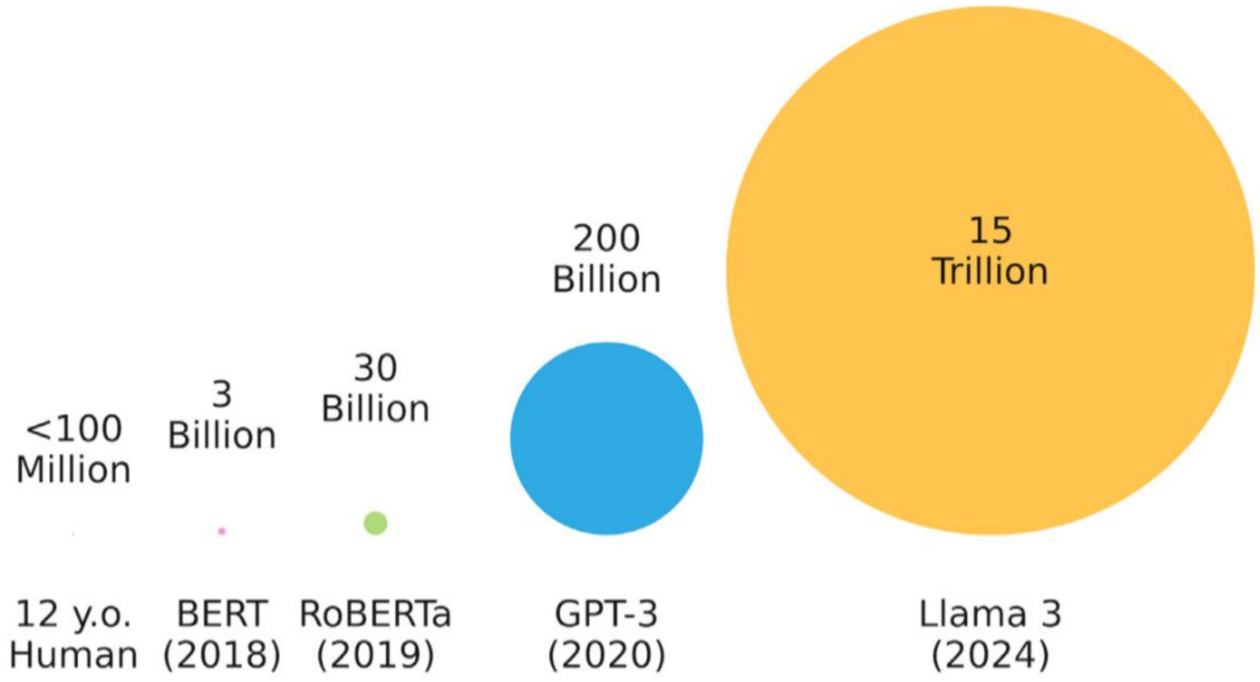
The State of Transfer Learning in NLP: https://ruder.io/state-of-transfer-learning-in-nlp/

Pre-training

- During the "*Pre-training*" phase, the model is trained in an unsupervised manner (e.g. LM, MLM) on a huge collection of raw text

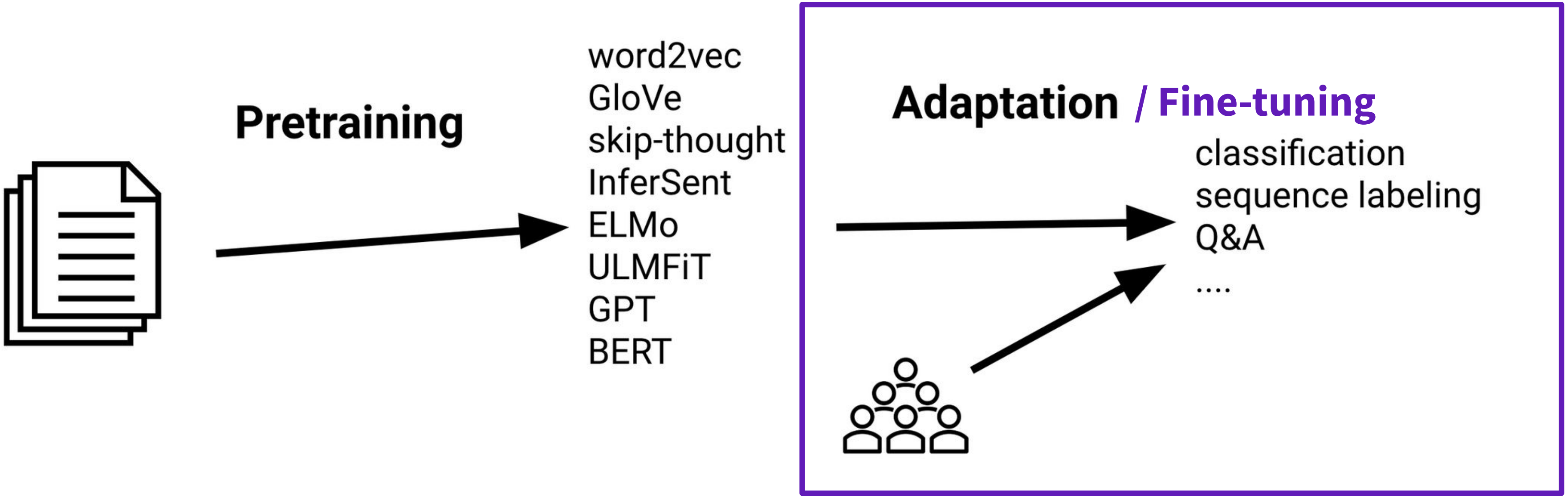
- Some exaomples:
  - **BERT training**: BookCorpus (800M words) + English Wikipedia (2500M words)
  - **GPT-3 training**: CommonCrawl + WebText2 + Books1 + Books2 + Wikipedia (around 500B words)

Pre-training



Source: https://www.sciencedirect.com/science/article/pii/S0749596X25000439

Transfer Learning



The State of Transfer Learning in NLP: https://ruder.io/state-of-transfer-learning-in-nlp/

Prompting → Large Language Models (LLMs)

- In recent years, the development of NLMs has shifted towards the creation of generative models:
  - Main goal: framing any task (e.g., classification, translation, question answering, etc.) as a **generation task**

Prompting → Large Language Models (LLMs)

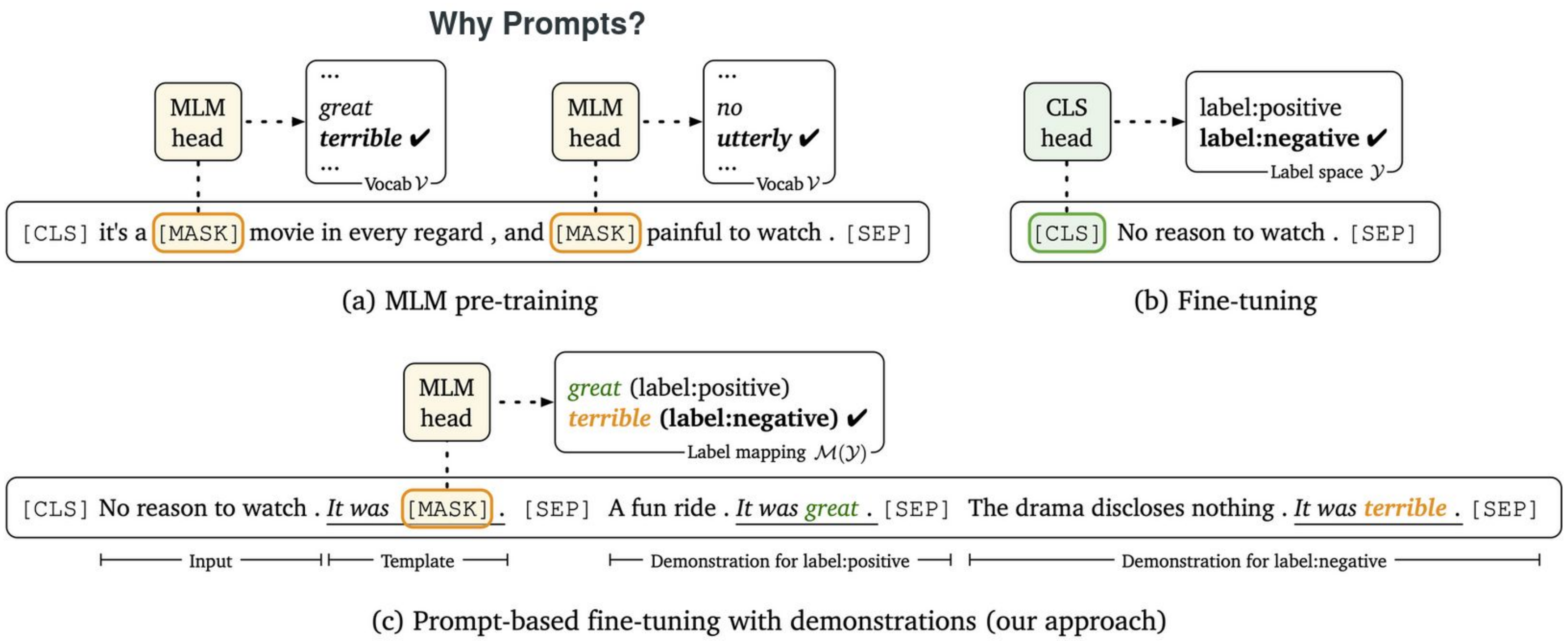
- In recent years, the development of NLMs has shifted towards the creation of generative models:
  - Main goal: framing any task (e.g., classification, translation, question answering, etc.) as a **generation task**

Prompting

"A prompt is a piece of text inserted in the input examples, so that the original task can be formulated as a (masked) language modeling problem."

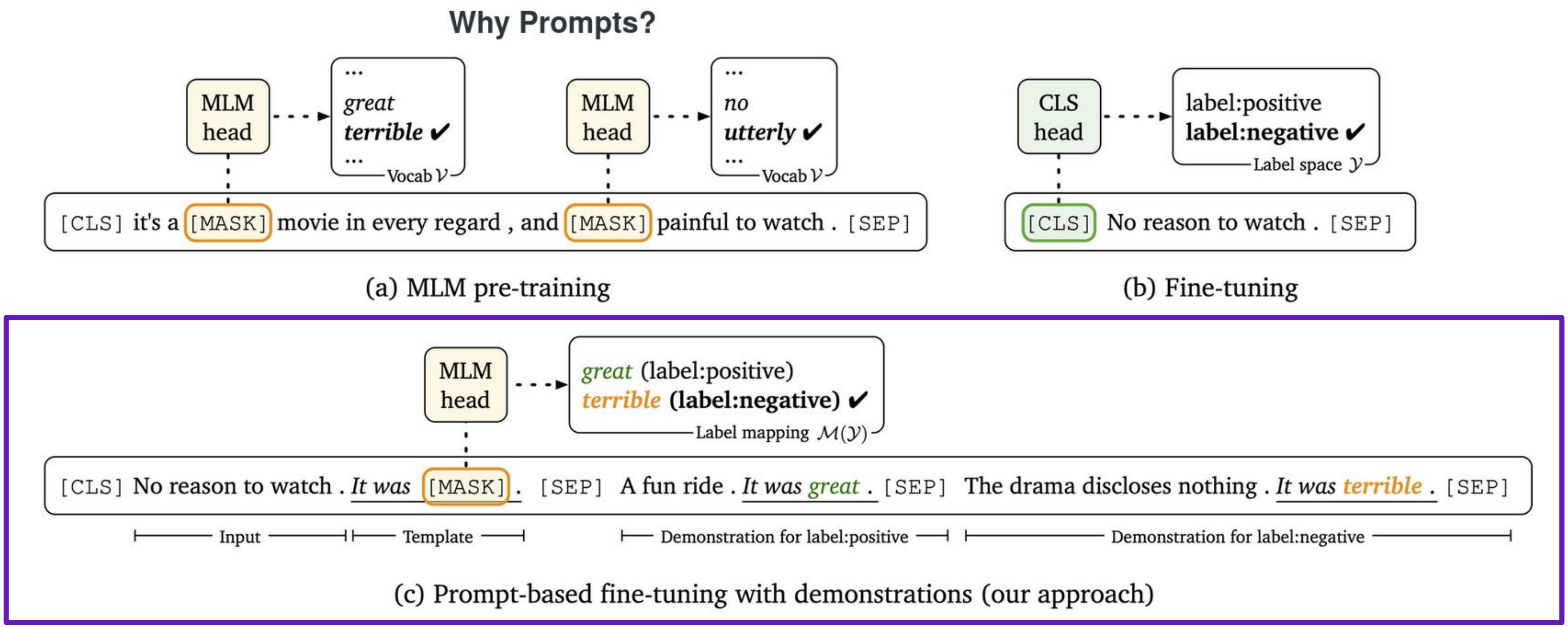
(Prompting: Better Ways of Using Language Models for NLP Tasks, The Gradient)

Prompting → Large Language Models (LLMs)



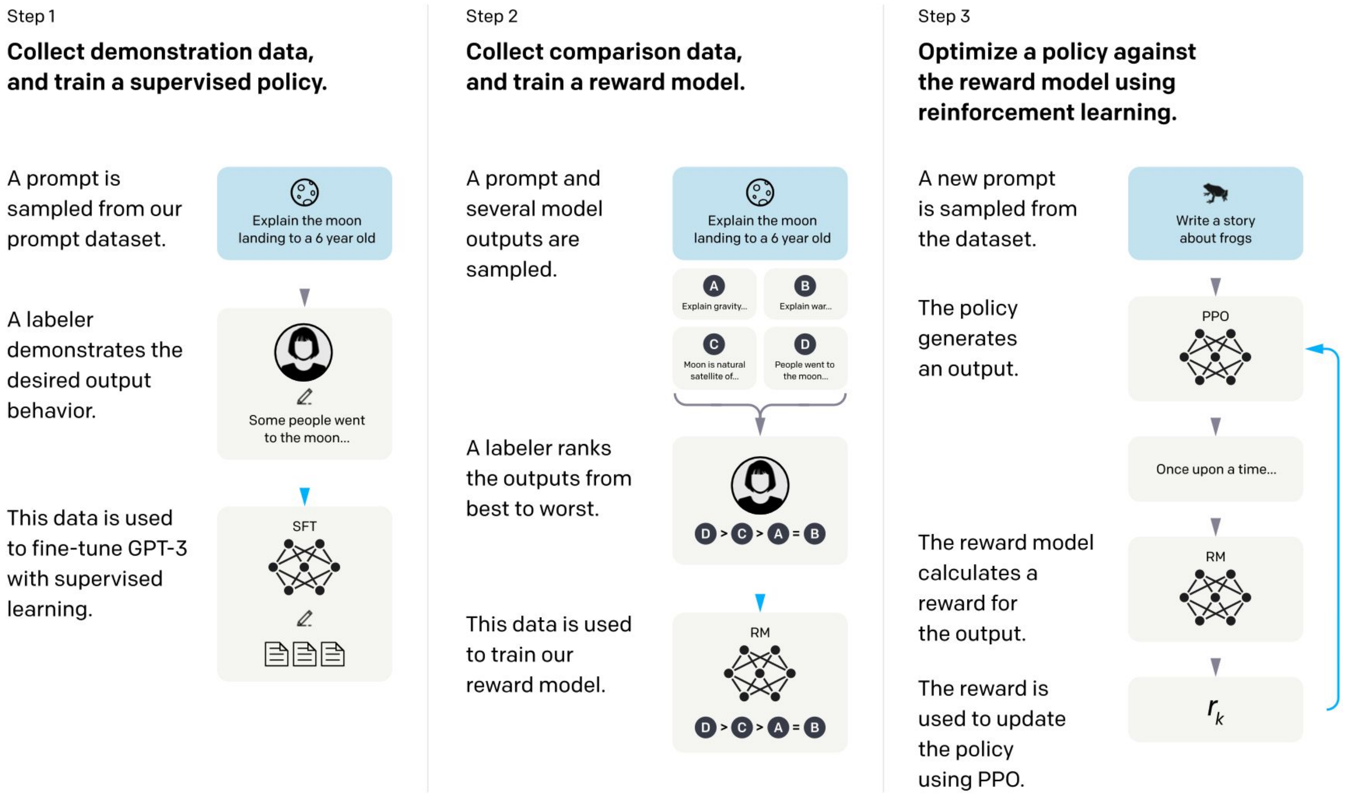
Source: [Prompting: Better Ways of Using Language Models for NLP Tasks, The Gradient]

Prompting → Large Language Models (LLMs)

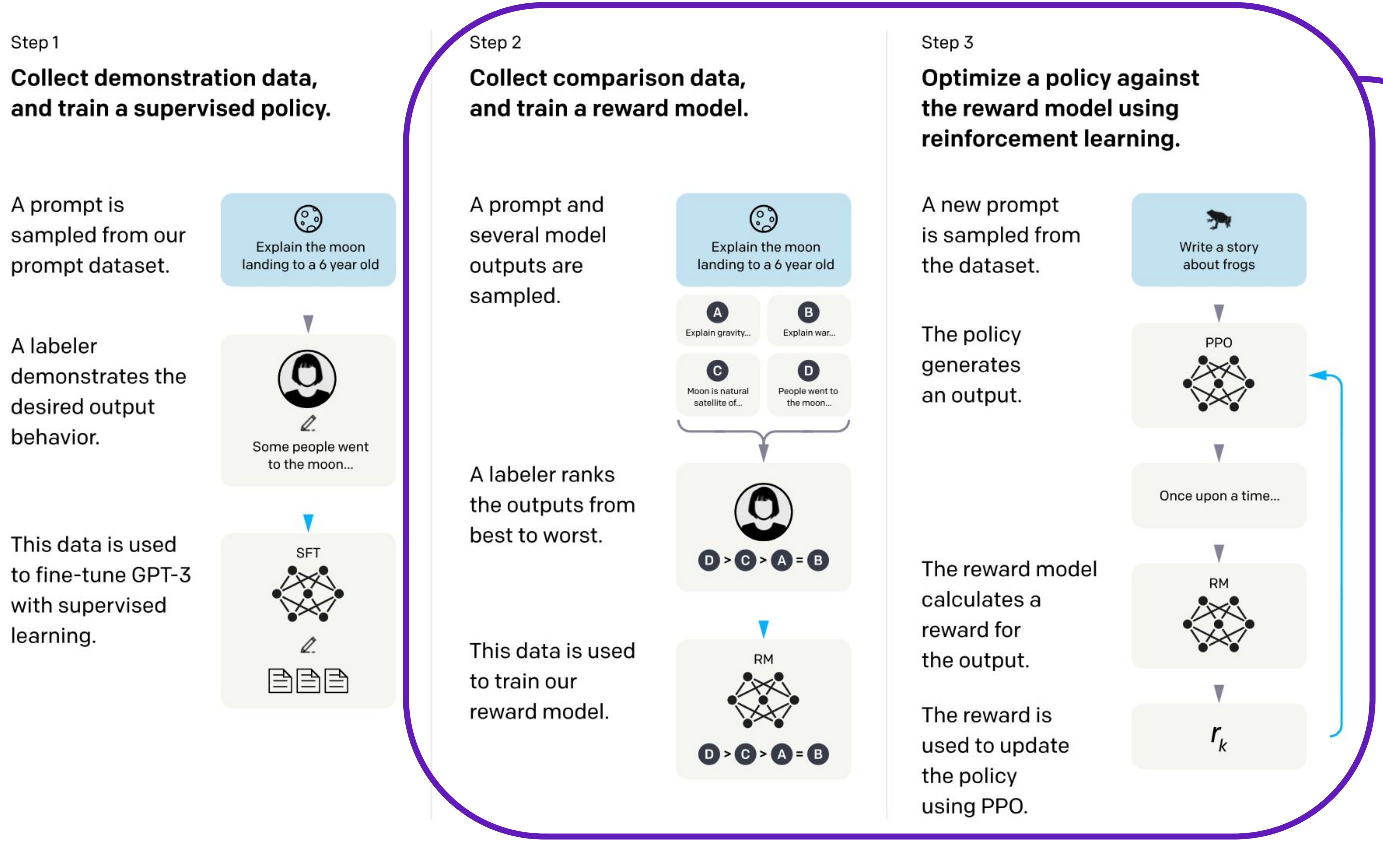


Source: [Prompting: Better Ways of Using Language Models for NLP Tasks, The Gradient]

Instruction Tuning e RLHF: from GPT-3 to InstructGPT



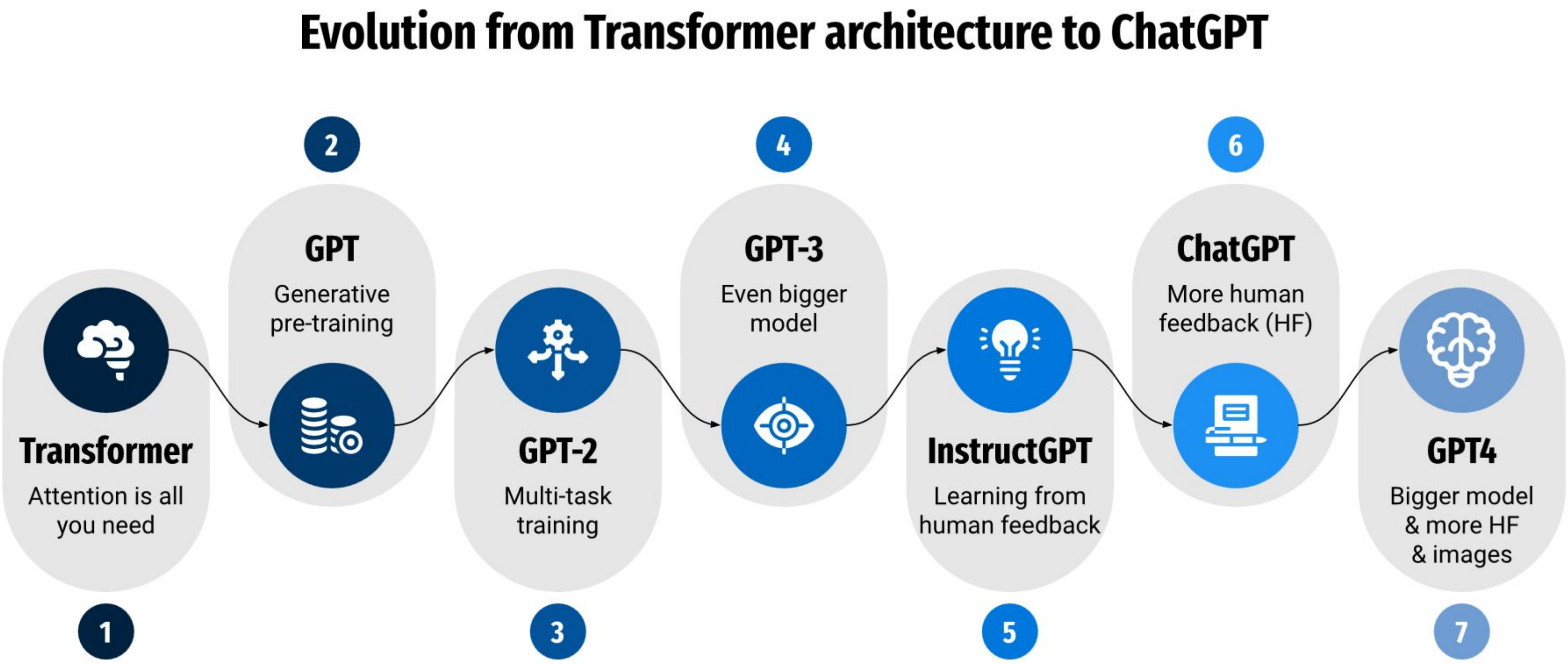
Instruction Tuning e RLHF: from GPT-3 to InstructGPT



Reinforcement Learning from Human Feedback (RLHF)

https://huggingface.co/blog/rlhf

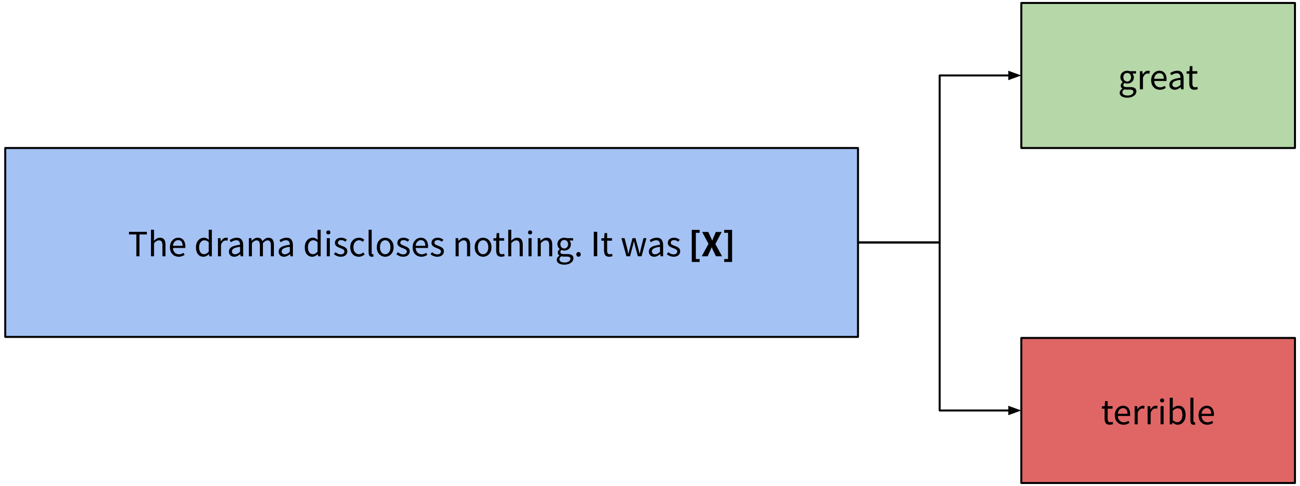
From Transformer to GPT4



[ChatGPT: Jack of all trades, master of none (Kocoń et al., 2023), https://www.sciencedirect.com/science/article/pii/S156625352300177X](https://www.sciencedirect.com/science/article/pii/S156625352300177X)

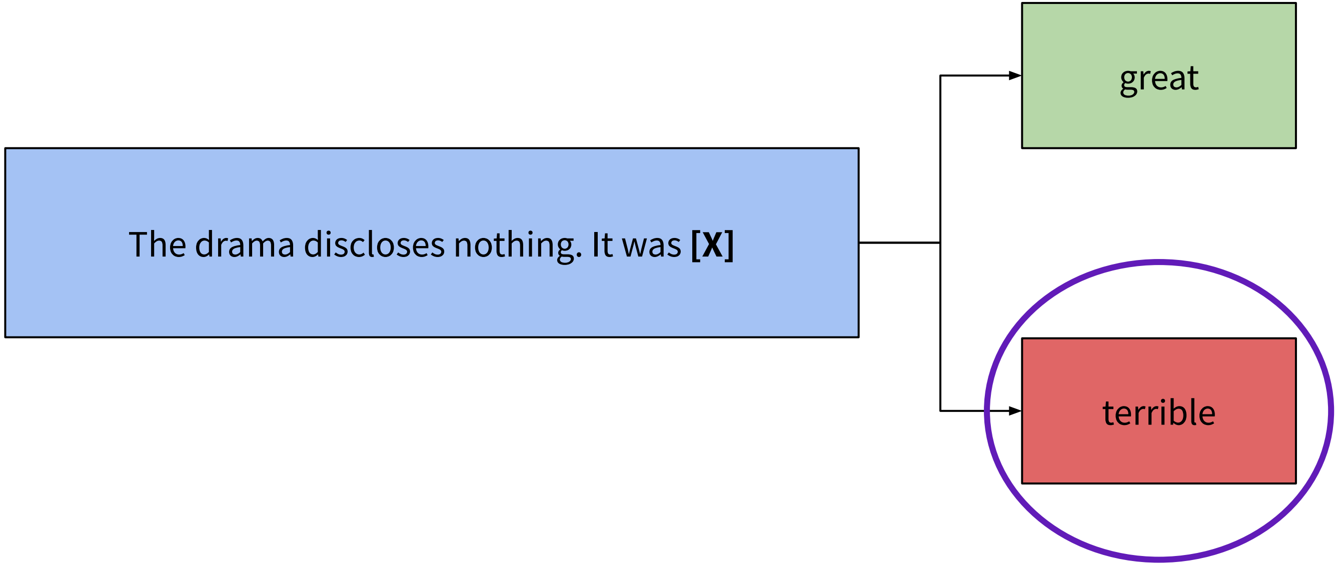
Large Language Models (LLMs)

Zero-Shot Text Classification



Large Language Models (LLMs)

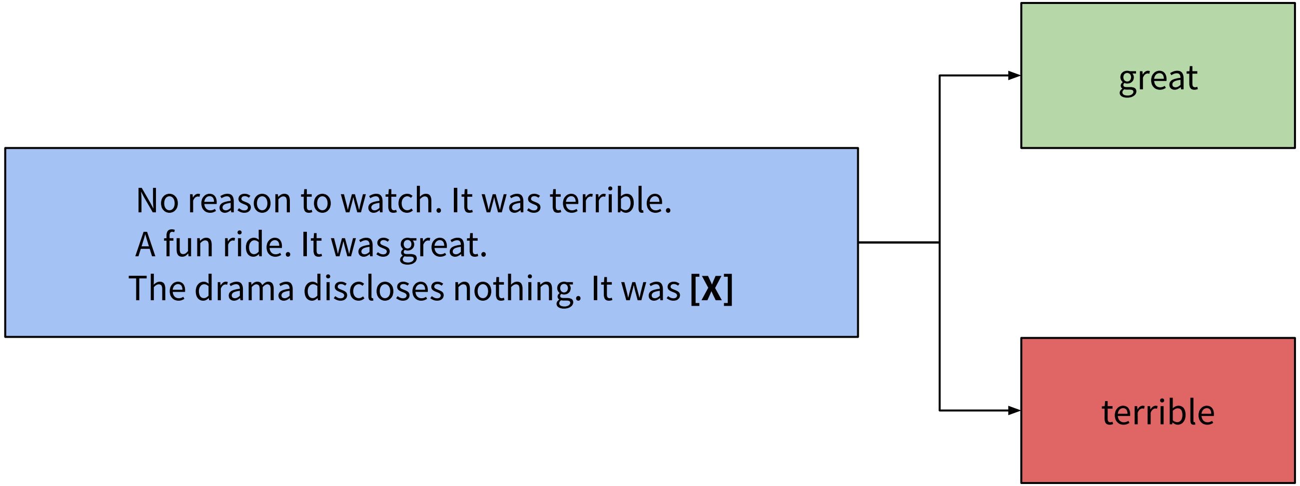
Zero-Shot Text Classification



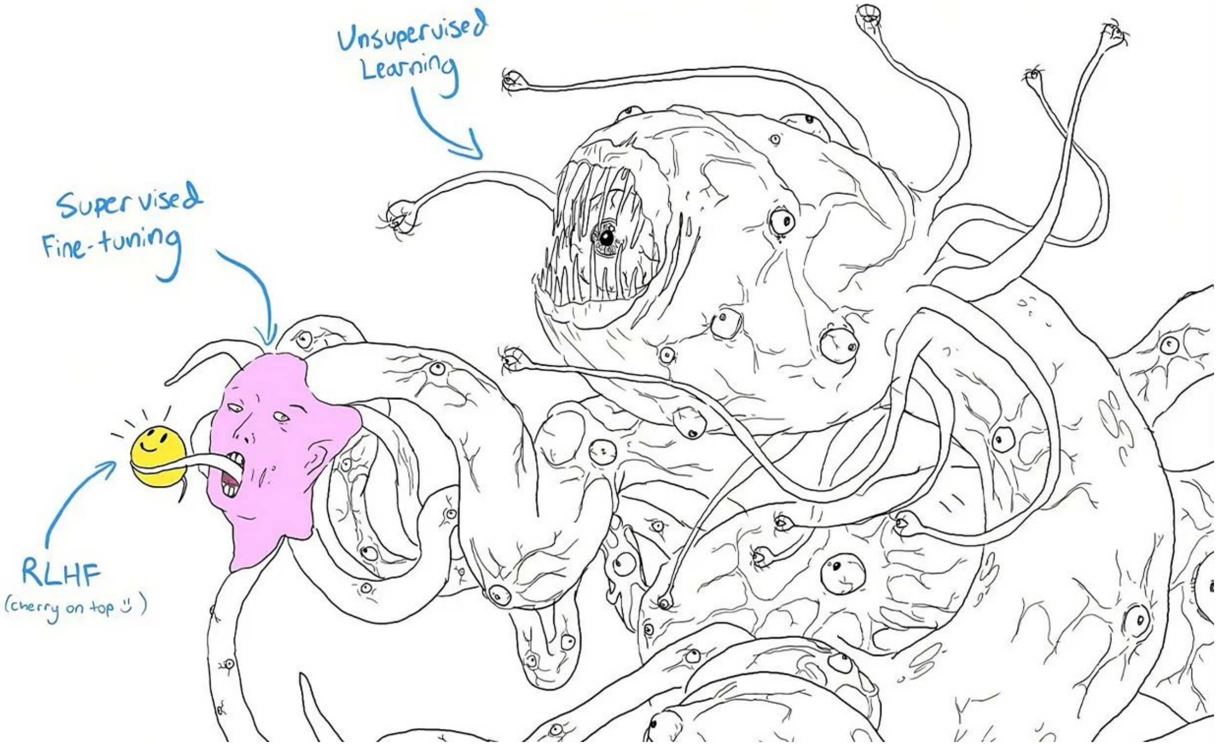
Comparison between the probability that the model will generate the token “”great” with respect to the token “terrible”

Large Language Models (LLMs)

Few-Shot Text Classification

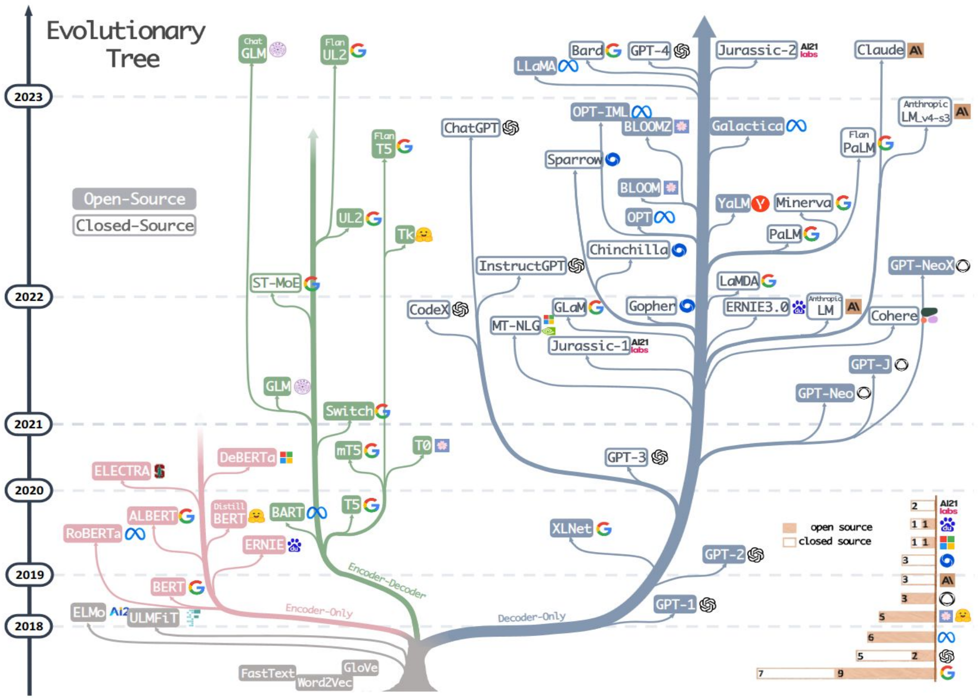


From Transformer to GPT4



From: https://medium.com/@mataciunasdeividas/the-simple-explanation-of-chatgpt-llm-rlhf-using-shoggoth-with-smiley-face-meme-947a0e9fb441

“Evolutionary Tree”



[Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond (Yang et al., 2024), https://dl.acm.org/doi/10.1145/3649506](https://dl.acm.org/doi/10.1145/3649506)

Before Practicing: Defining the Model's Vocabulary

- Even though NLMs are trained in an unsupervised way, i.e. learning directly from large text corpora, one crucial design choice must be made before training begins

Before Practicing: Defining the Model's Vocabulary

- Even though NLMs are trained in an unsupervised way, i.e. learning directly from large text corpora, one crucial design choice must be made before training begins

The vocabulary

- It defines what the model considers as the basic building blocks of language
- Every text the model sees will be represented as sequences of these predefined units (i.e. tokens)
- This brings us to a key concept in modern LMs and NLP in general:
  - **Tokenization**: how text is split into tokens according to the chosen vocabulary

Tokenization

- Before a sequence (e.g., sentence, document) can be passed to an NLM, it must first undergo **tokenization**

- Depending on the type of model used, there are several tokenizers capable of segmenting text:
  - Byte-Pair Encoding (BPE); WordPiece

- The principles behind the tokenizers most commonly used with recent NLM are:
  - Frequently used words should not be split into smaller subwords
  - Rare (less frequent) words should be split into meaningful subwords

Byte-Pair Encoding (BPE) Tokenization

- **Byte-Pair Encoding (BPE)** was initially developed as an algorithm for compressing text and was then used by OpenAI for tokenization during the pre-training of the first GPT

- Algorithm:
    1. Each word is broken down into individual characters
    2. Computation of the most frequent pair of adjacent characters in the text
    3. Merging of the pair into a new “subtoken” to be added to the vocabulary
    4. Repetition of steps 2-3 until the desired number of tokens is reached

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus		Vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s, t, w
2	l o w e s t _	
6	n e w e r _	
3	w i d e r _	
2	n e w _	

https://cs.usm.maine.edu/~behrooz.mansouri/courses/Slides_NLP_23/Natural%20Language%20Processing%20--%20Session%204%20-%20Tokenization%20and%20Stemming.pdf

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

	5	l o w __
	2	l o w e s t __
9 times	6	n e w e r __
	3	w i d e r __
	2	n e w __

Vocabulary

__, d, e, i, l, n, o, r, s, t, w

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er

https://cs.usm.maine.edu/~behrooz.mansouri/courses/Slides_NLP_23/Natural%20Language%20Processing%20--%20Session%204%20-%20Tokenization%20and%20Stemming.pdf

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

	5	l o w __
	2	l o w e s t __
9 times	6	n e w e r __
	3	w i d e r __
	2	n e w __

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__

https://cs.usm.maine.edu/~behrooz.mansouri/courses/Slides_NLP_23/Natural%20Language%20Processing%20--%20Session%204%20-%20Tokenization%20and%20Stemming.pdf

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

5	l o w __
2	l o w e s t __
6	n e w e r __
3	w i d e r __
2	n e w __

8 times

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne

https://cs.usm.maine.edu/~behrooz.mansouri/courses/Slides_NLP_23/Natural%20Language%20Processing%20--%20Session%204%20-%20Tokenization%20and%20Stemming.pdf

Byte-Pair Encoding (BPE) Tokenization

Training corpus: low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Corpus

8 times

5 l o w __
2 l o w e s t __
6 n e w e r __
3 w i d e r __
2 n e w __

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne

Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne, new

https://cs.usm.maine.edu/~behrooz.mansouri/courses/Slides_NLP_23/Natural%20Language%20Processing%20--%20Session%204%20-%20Tokenization%20and%20Stemming.pdf

Byte-Pair Encoding (BPE) Tokenization

Corpus

5 l o w __
2 l o w e s t __
6 n e w e r __
3 w i d e r __
2 n e w __

Final Vocabulary

__, d, e, i, l, n, o, r, s, t, w, er, er__, ne, new, lo, low, newer __, low__

Using BPE for tokenization:

Input: newer__ → **Tokens: newer__**

Input: lower__ → **Tokens: low, er__**

Merge based on the order we learned:
er → er__ → ne → new → newer__
er → er__ → lo → low

https://cs.usm.maine.edu/~behrooz.mansouri/courses/Slides_NLP_23/Natural%20Language%20Processing%20--%20Session%204%20-%20Tokenization%20and%20Stemming.pdf

The *Transformers* library 🤗

- The **Transformers** library (by **HuggingFace**) is currently the most widely used open source resource for easily downloading, modifying, and training Transformer models

🤗 **Transformers**

📝 **Natural Language Processing**: text classification, named entity recognition, question answering, language modeling, summarization, translation, multiple choice, and text generation.

🖼️ **Computer Vision**: image classification, object detection, and segmentation.

🗣️ **Audio**: automatic speech recognition and audio classification.

🐙 **Multimodal**: table question answering, optical character recognition, information extraction from scanned documents, video classification, and visual question answering.

Part II

Interpreting and Evaluating NLMs

Interpreting and Evaluating NLMs

- The rapid development and widespread adoption of state-of-the-art Neural Language Models (NLMs) have increased the need for studies focused on their **interpretability** and the **evaluation** of their abilities

NLMs Interpretability

NLMs Evaluation

Interpreting and Evaluating NLMs

- The rapid development and widespread adoption of state-of-the-art Neural Language Models (NLMs) have increased the need for studies focused on their **interpretability** and the **evaluation** of their abilities

NLMs Interpretability

NLMs Evaluation

The Case for Interpretability

- The development of powerful state-of-the-art NLMs comes at the cost of **interpretability**, since complex NN models offer little transparency about their inner workings and their abilities

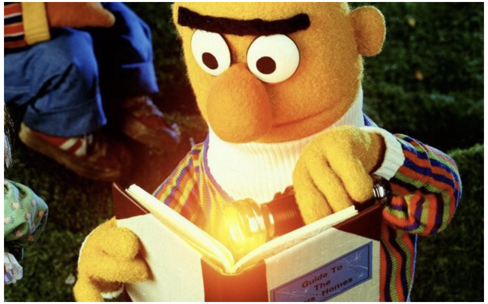
Objectives:

- **Understand the nature of AI systems** → be faithful to what influences the AI decisional process
- **Empower AI system users** → derive actionable useful insights from AI choices

Interpretability in NLP

"In the context of NLP, this question needs to be understood in light of earlier NLP work. [...] In some of these systems, features are more easily understood by humans. [...] In contrast, it is more difficult to understand what happens in an end-to-end neural network model that takes input (say, word embeddings) and generates an output."

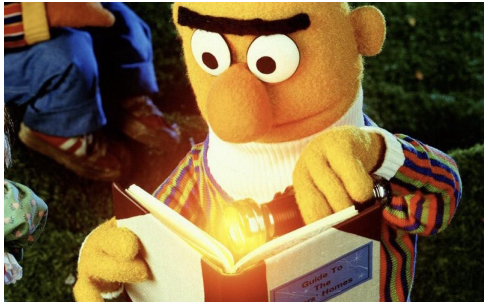
Belinkov and Glass, Analysis Methods in Neural Language Processing: A Survey (2019). In Transactions of ACL, Volume 7, pages 49-72.



Interpretability in NLP

“In the context of NLP, this question needs to be understood in light of earlier NLP work. […] In some of these systems, features are more easily understood by humans. […] In contrast, it is more difficult to understand what happens in an end-to-end neural network model that takes input (say, word embeddings) and generates an output.”

Belinkov and Glass, Analysis Methods in Neural Language Processing: A Survey (2019). In Transactions of ACL, Volume 7, pages 49-72.

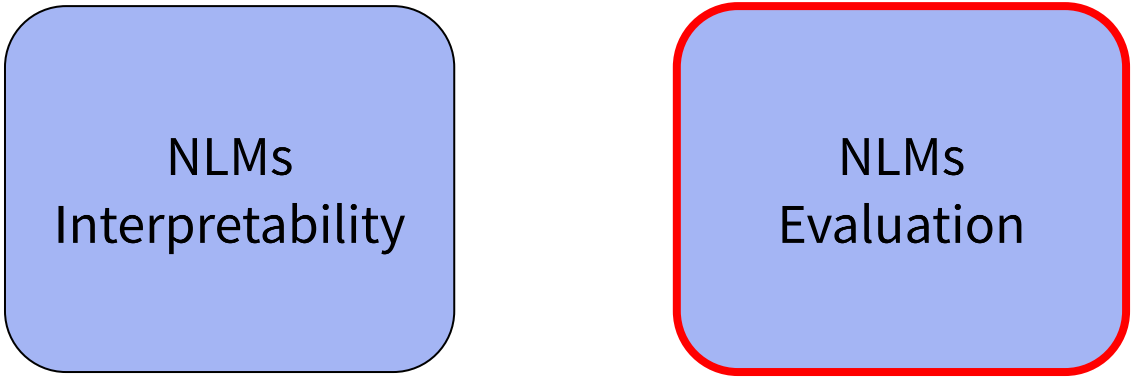


Research questions:

- What happens in an end-to-end neural network model when trained on a language modeling task?
- What kind of linguistic knowledge (i.e. features) is encoded within their representations?
- Is there a relationship between the linguistic knowledge implicitly encoded and the ability to solve a specific task?

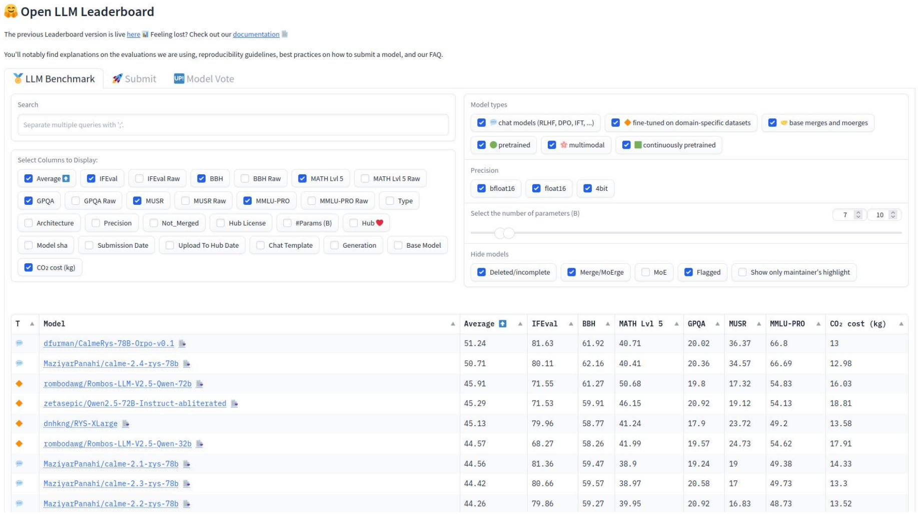
Interpreting and Evaluating NLMs

- The rapid development and widespread adoption of state-of-the-art Neural Language Models (NLMs) have increased the need for studies focused on their **interpretability** and the **evaluation** of their abilities



Evaluation of Neural Language Models

- The evaluation of NLMs has seen significant advancements in the past few years, with the development of dedicated benchmarks and evaluation frameworks
- These benchmarks are designed to assess models' performance on specific tasks and reasoning abilities:
  - OpenLLM Leaderboard
  - BigBench (Srivastava et al., 2023)
  - Holmes (Waldis et al., 2024)



Link: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

The Limits of the Modern Evaluation Landscape

Evaluating LLMs is far from straightforward, since there are several factors that can distort or limit our understanding of their true capabilities:

Data Contamination: Models may have seen parts of the evaluation data during training, leading to inflated performance

Narrow Benchmark: Many benchmarks test specific tasks or surface-level skills, failing to capture general reasoning or robustness

Prompt Sensitivity: Results can vary dramatically depending on prompt wording, context, or even output format

Evaluation Metrics Limitations: Automatic metrics are often unreliable for open-ended generation, while human evaluation remains costly and complex to perform consistently

Evaluating NLMs Linguistic Abilities

- Within the broader context of interpretability and evaluation, one line of research focuses on studying and assessing the linguistic abilities of Neural Language Models

- Such studies aim to uncover:
  - the implicit linguistic competencence encoded within these models
  - evaluate their generalization abilities

Assessing BERT's Syntactic Abilities (Goldberg, 2019)

- Goldberg (2019) proposes a methodology for testing the implicit linguistic competence of BERT

- Specifically, two linguistic phenomena are considered:
  - Subject-Verb Agreement;
  - Reflexive Anaphora.

- **Approach**: masking target words and asking the model to "fill in the gap" with the words with high probability scores

Assessing BERT's Syntactic Abilities (Goldberg, 2019)

the game that the guard hates is bad

Assessing BERT's Syntactic Abilities (Goldberg, 2019)

the game that the guard hates **[MASK]** bad

Assessing BERT's Syntactic Abilities (Goldberg, 2019)

the game that the guard hates **[MASK]** bad

- *p(is)* = ?
- *p(are)* = ?

Assessing BERT's Syntactic Abilities (Goldberg, 2019)

	BERT Base	BERT Large	LSTM (M&L)	Humans (M&L)	# Pairs (# M&L Pairs)
SUBJECT-VERB AGREEMENT:					
Simple	1.00	1.00	0.94	0.96	120 (140)
In a sentential complement	0.83	0.86	0.99	0.93	1440 (1680)
Short VP coordination	0.89	0.86	0.90	0.82	720 (840)
Long VP coordination	0.98	0.97	0.61	0.82	400 (400)
Across a prepositional phrase	0.85	0.85	0.57	0.85	19440 (22400)
Across a subject relative clause	0.84	0.85	0.56	0.88	9600 (11200)
Across an object relative clause	0.89	0.85	0.50	0.85	19680 (22400)
Across an object relative (no *that*)	0.86	0.81	0.52	0.82	19680 (22400)
In an object relative clause	0.95	0.99	0.84	0.78	15960 (22400)
In an object relative (no *that*)	0.79	0.82	0.71	0.79	15960 (22400)
REFLEXIVE ANAPHORA:					
Simple	0.94	0.92	0.83	0.96	280 (280)
In a sentential complement	0.89	0.86	0.86	0.91	3360 (3360)
Across a relative clause	0.80	0.76	0.55	0.87	22400 (22400)

Table 3: Results on the Marvin and Linzen (2018) stimuli. M&L results numbers are taken from Marvin and Linzen (2018). The BERT and M&L numbers are *not* directly comparable, as the experimental setup differs in many ways.

BLiMP (Warstadt A. et al., 2020)

- Evaluate the linguistic competence of language models through controlled, theory-driven tests
- **Key idea**: Use minimal pairs to test whether a model assigns higher probability to the grammatical sentence
- Design:
  - 67 test sets, each targeting a specific phenomenon
  - Around 1000 pairs per phenomenon
  - Automatically generated with linguistically precise templates to ensure syntactic and semantic control

Phenomenon	N	Acceptable Example	Unacceptable Example
ANAPHOR AGR.	2	*Many girls insulted themselves.*	*Many girls insulted herself.*
ARG. STRUCTURE	9	*Rose wasn't disturbing Mark.*	*Rose wasn't boasting Mark.*
BINDING	7	*Carlos said that Lori helped him.*	*Carlos said that Lori helped himself.*
CONTROL/RAISING	5	*There was bound to be a fish escaping.*	*There was unable to be a fish escaping.*
DET.-NOUN AGR.	8	*Rachelle had bought that chair.*	*Rachelle had bought that chairs.*
ELLIPSIS	2	*Anne's doctor cleans one important book and Stacey cleans a few.*	*Anne's doctor cleans one book and Stacey cleans a few important.*
FILLER-GAP	7	*Brett knew what many waiters find.*	*Brett knew that many waiters find.*
IRREGULAR FORMS	2	*Aaron broke the unicycle.*	*Aaron broken the unicycle.*
ISLAND EFFECTS	8	*Whose hat should Tonya wear?*	*Whose should Tonya wear hat?*
NPI LICENSING	7	*The truck has clearly tipped over.*	*The truck has ever tipped over.*
QUANTIFIERS	4	*No boy knew fewer than six guys.*	*No boy knew at most six guys.*
SUBJECT-VERB AGR.	6	*These casseroles disgust Kayla.*	*These casseroles disgusts Kayla.*

Table 2: Minimal pairs from each of the twelve linguistic phenomenon categories covered by BLiMP. Differences are underlined. N is the number of 1,000-example minimal pair paradigms within each broad category.

BLiMP (Warstadt A. et al., 2020)

- Evaluate the linguistic competence of language models through controlled, theo
- **Key** whet prob
- Desi
  - ○
  - ○
  - ○ precise templates to ensure syntactic and semantic control

Model	Overall	Ana. Agr	Arg. Str	Binding	Ctrl. Rais.	D-N Agr	Ellipsis	Filler. Gap	Irregular	Island	NPI	Quantifiers	S-V Agr
5-gram	60.5	47.9	71.9	64.4	68.5	70.0	36.9	58.1	79.5	53.7	45.5	53.5	60.3
LSTM	68.9	91.7	73.2	73.5	67.0	85.4	67.6	72.5	89.1	42.9	51.7	64.5	80.1
TXL	68.7	94.1	69.5	74.7	71.5	83.0	77.2	64.9	78.2	45.8	55.2	69.3	76.0
GPT-2	80.1	99.6	78.3	80.1	80.5	93.3	86.6	79.0	84.1	63.1	78.9	71.3	89.0
Human	88.6	97.5	90.0	87.3	83.9	92.2	85.0	86.9	97.0	84.9	88.1	86.6	90.9

Table 3: Percentage accuracy of four baseline models and raw human performance on BLiMP using a forced-choice task. A random guessing baseline would achieve an accuracy of 50%.

Example

ulted herself.
pasting Mark.
it Lori helped himself.
ble to be a fish escaping.
ought that chairs.
cleans one book and
ns a few important.
t many waiters find.
the unicycle.
Tonya wear hat?
ever tipped over.
t most six guys.
es disgusts Kayla.

ories covered by BLiMP. r paradigms within each broad category.

Evaluating Lexical Proficiency in Neural Language Models

- Few works focused on investigating and evaluating NLMs' abilities in tasks related to lexical proficiency

- Almost no study that goes beyond commonly lexicalized words

Ciaccio C., Miaschi A., Dell'Orletta F. (2025). Evaluating Lexical Proficiency in Neural Language Models. In *Proceedings of ACL 2025*, July 27 - August 1, Vienna

Evaluating Lexical Proficiency in Neural Language Models

- Few works focused on investigating and evaluating NLMs' abilities in tasks related to lexical proficiency

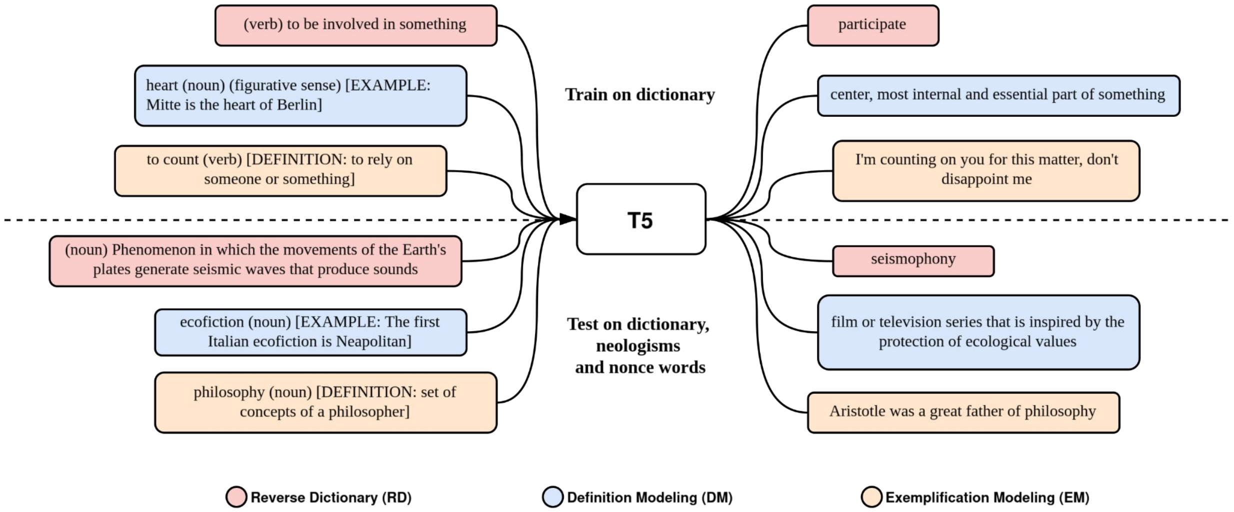
- Almost no study that goes beyond commonly lexicalized words

- We propose an evaluation framework for testing the lexical proficiency of LMs on different linguistic settings for the Italian language

Ciaccio C., Miaschi A., Dell'Orletta F. (2025). Evaluating Lexical Proficiency in Neural Language Models. In *Proceedings of ACL 2025*, July 27 - August 1, Vienna

Our Approach

- Evaluation of Encoder-Decoder Models on a mixture of tasks that implicitly exposes the morpho-lexical link that relates lemmas to definitions



- **Reverse Dictionary:** generating a target word given a source definition
- **Definition Modeling:** generating a definition given a word
- **Exemplification Modeling:** generating a usage example given a word paired with a definition

Settings, Data and Models

- We conducted our evaluation across three different settings:
  - **Dictionary setting:** Evaluating against an unseen split of the models training dataset
  - **Neologism setting:** Evaluating against unseen neologisms that have zero to few occurrences in the models' **pretraining data**
  - **Nonce words setting:** assessing the linguistically creative abilities in creating, defining, and using nonce words (i.e. unseen words)

- Three different training/evaluation datasets:
  - **Dictionary dataset:** We developed a new resources starting from the April 2024 Wikizionario Dump + ONLI (*Osservatorio Neologico della Lingua Italiana*) neologism database
  - **Neologism dataset:** We collected a list of neologisms from various online dictionaries (appearing between 2021 to 2024) and kept only those with less then five occurrences in the pretraining dataset of our models
  - **Nonce words dataset:** We used GPT-4o to obtain a list of 100 unattested nonce words

Model	Lang	#P	#T	#T/#P
IT5-small	IT	60M	41B	683.33
IT5-base	IT	220M	41B	186.36
MT5-base	Multi	580M	6.3T	10,862.06
IT5-large	IT	738M	41B	55.55

Table 2: Models used in experiments along with the pre-training languages (*Lang*), number of parameters (*#P*), number of training tokens (*#T*) and the number of tokens per parameter (*#T/#P*).

Results

		Reverse Dictionary					Definition Modeling				Exemplification Modeling	
		Acc@1/10/100	R1	R2	CER↓	SBERT	R1	R2	RL	SBERT	PPL pred. ↓	PPL target
Dict.	**IT5-small**	*.29/.4/.53*	*41.33*	*31.19*	*50.58*	*0.68*	36.85	*23.98*	34.87	*0.61*	144.49	80.26
	IT5-base	.37/.52/.66	48	37.01	46	0.71	**39.58**	**26.54**	**37.42**	**0.65**	118.26	
	MT5-base	.33/.46/.57	43.64	33.73	47.95	0.7	*36.43*	24.58	*34.71*	0.62	*161.8*	
	IT5-large	**.39/.56/.69**	**49.7**	**38.8**	**43.83**	**0.73**	38.97	25.94	36.94	**0.65**	**112.66**	
	Avg	.34/.48/.61	45.67	35.18	47.09	0.7	37.96	25.26	35.98	0.63	134.3	
Neo.	**IT5-small**	*.06/.12/.13*	*25.39*	16.37	*71.95*	*0.55*	*18.36*	*3.44*	14.8	*0.45*	60.6	53.38
	IT5-base	.09/**.16**/.21	**33.06**	19.99	61.47	**0.6**	**21.21**	**5.36**	**16.92**	**0.53**	53.6	
	MT5-base	.08/.15/.18	26.82	*14.23*	**59.98**	0.59	18.43	3.66	*14.4*	0.48	*79.52*	
	IT5-large	**.1/.16/.27**	32.42	**20.64**	63.2	**0.6**	20.69	4.34	16.36	**0.53**	**43.44**	
	Avg	.08/.14/.19	29.4	17.8	64.05	0.58	19.67	4.2	15.62	0.5	59.15	
Nonce	**IT5-small**	—	—	—	—	—	18.91	*2.83*	15.13	*0.49*	68.35	64.28
	IT5-base	—	—	—	—	—	**21.79**	**4.19**	**17.13**	0.56	67.31	
	MT5-base	—	—	—	—	—	*18.1*	2.93	*14.15*	0.51	*84.33*	
	IT5-large	—	—	—	—	—	21.09	3.78	16.6	**0.58**	**48.05**	
	Avg	—	—	—	—	—	19.97	3.42	15.72	0.53	67.01	

Table 3: Results obtained by all the models for all the tasks (RD, DM and EM) and the three linguistically different settings: *Dict.*, *Neo.* and *Nonce*.

Results - Human Evaluation

- We collected human judgments over 100 pairs of definitions (taken from the nonce words dataset) and nonce words (generated by our models)
    - We asked 5 Italian native speakers to read each definition-word pair and express two judgments about the nonce word according to the **perceived novelty** and the **adhesion to the definition**

Results - Human Evaluation

- We collected human judgments over 100 pairs of definitions (taken from the nonce words dataset) and nonce words (generated by our models)
  - We asked 5 Italian native speakers to read each definition-word pair and express two judgments about the nonce word according to the **perceived novelty** and the **adhesion to the definition**
    - **Optimal Innovation Hypothesis** (Giora R. et al., 2004)

	Adhesion	Novelty	α
IT5-small	3.06±1.45	3.11±1.3	.51/.14
IT5-base	3.01±1.32	3.61±1.37	.29/.34
MT5-base	3.37±1.32	2.98±1.31	.37/.15
IT5-large	3.37±1.42	3.11±1.15	.41/.18
GPT-4o	3.86±1.09	3.32±1.15	.17/.07

Table 5: Mean and standard deviation for the adhesion and novelty scores given by human annotators. The column α reports the Krippendorff's Alpha between annotators for adhesion/novelty.

Results - Human Evaluation

- We collected human judgments over 100 pairs of definitions (taken from the nonce words dataset) and nonce words (generated by our models)
  - We asked 5 Italian native speakers to read each definition-word pair and express two judgments about the nonce word according to the **perceived novelty** and the **adhesion to the definition**
    - **Optimal Innovation Hypothesis** (Giora R. et al., 2004)

	Adhesion	Novelty	α
IT5-small	3.06±1.45	3.11±1.3	.51/.14
IT5-base	3.01±1.32	3.61±1.37	.29/.34
MT5-base	3.37±1.32	2.98±1.31	.37/.15
IT5-large	3.37±1.42	3.11±1.15	.41/.18
GPT-4o	3.86±1.09	3.32±1.15	.17/.07

Table 5: Mean and standard deviation for the adhesion and novelty scores given by human annotators. The column α reports the Krippendorff's Alpha between annotators for adhesion/novelty.

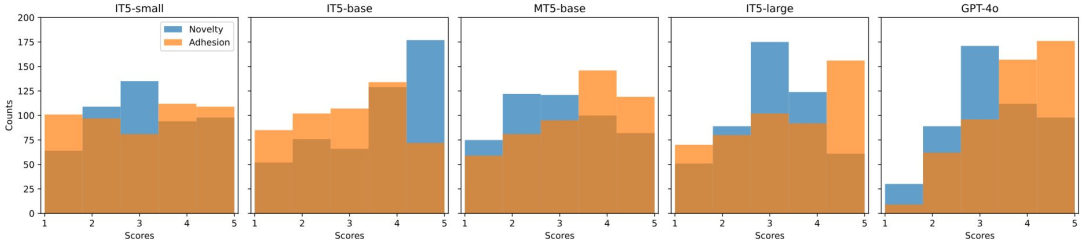


Figure 1: Distribution of novelty and adhesion human scores across the 5 values of the Likert scale for all models.

Results

Definitions	Model	Predicted Word	Adhesion	Novelty
Veicolo progettato per esplorazioni su superfici planetarie, adatto a terreni extraterrestri. [trad. *Vehicle designed for exploration on planetary surfaces, suitable for extraterrestrial terrain.*]	IT5-small	planetaro	3.0	4.2
	IT5-base	elioplano [trad. *helioplane*]	2.2	4.6
	MT5-base	cosmoplano [trad. *cosmoplane*]	3.2	4.0
	IT5-large	astroveicolo [trad. *astrovehicle*]	4.6	3.2
	GPT-4o	roverastro [trad. *astrorover*]	3.6	3.4
Vela navigabile che raccoglie dati geologici mentre si sposta su laghi o mari, utilizzata in esplorazioni scientifiche. [trad. *Navigable sail that collects geological data as it moves across lakes or seas, used in scientific exploration.*]	IT5-small	geonauta [trad. *geonaut*]	4.6	2.4
	IT5-base	ecovela [trad. *ecosail*]	4.4	1.8
	MT5-base	vettolaghiera	2.0	4.4
	IT5-large	idrovedetta [trad. *hydropatrol*]	4.6	2.8
	GPT-4o	geonave [trad. *geoship*]	4.0	3.2
Una tavola o superficie capace di mostrare visivamente il passare del tempo, evidenziando i cambiamenti avvenuti su di essa. [trad. *A table or surface capable of visually showing the passage of time, highlighting the changes that have occurred on it.*]	IT5-small	cromatopompa	1.2	3.8
	IT5-base	cronopalestra [trad. *chronogym*]	2.0	5.0
	MT5-base	retrotavola [trad. *retrotable*]	2.2	3.0
	IT5-large	cronotavolo [trad. *chronotable*]	4.4	3.0
	GPT-4o	cronotavola [trad. *chronotable*]	3.6	3.6
Forma d'arte che utilizza nebbie artificiali e giochi di luce per creare installazioni immersive. [trad. *An art form that uses artificial fog and light effects to create immersive installations.*]	IT5-small	immersivismo [trad. *immersivism*]	3.8	2.4
	IT5-base	metacaduta [trad. *metafall*]	2.0	4.6
	MT5-base	fotoart [trad. *photoart*]	3.4	2.6
	IT5-large	nebbiografia [trad. *foggraphy*]	4.4	3.0
	GPT-4o	nebbioarte [trad. *fogart*]	3.6	3.6
Fenomeno in cui i movimenti delle placche terrestri generano onde sismiche che producono suoni dissonanti, studiato in geologia e acustica. [trad. *Phenomenon in which the movements of the earth's plates generate seismic waves that produce dissonant sounds, studied in geology and acoustics.*]	IT5-small	biogeoacustica [trad. *biogeoacoustics*]	4.4	3.4
	IT5-base	sismofonia [trad. *seismophony*]	3.0	4.0
	MT5-base	sismismo [trad. *seismism*]	3.0	4.0
	IT5-large	sismofonia [trad. *seismophony*]	4.2	3.2
	GPT-4o	sismofonia [trad. *seismophony*]	4.2	2.0

Table 6: Sample of generated nonce words (we tried to provide a translation when possible), along with adhesion and novelty average scores, for all the models. The definitions are those generated by GPT-4o.

"Astroveicolo"

Selected Findings

- Larger, monolingual models generally outperformed their multilingual counterparts

- Despite the drop in performance with low-frequency neologisms and nonce words, the rank between models remained consistent

- The models' ability to generate novel and coherent nonce words further indicates LMs are capable of **learning approximations of word formation rules**, rather than relying solely on memorization

Istituto di Linguistica Computazionale "Antonio Zampolli"
Consiglio Nazionale delle Ricerche

Thanks for the attention!

https://alemiaschi.github.io/

@AlessioMiaschi

http://www.italianlp.it/

@ItaliaNLP_Lab

References

- Bengio, Yoshua, et al. (2003). "A neural probabilistic language model." *The journal of machine learning research* 3, pages 1137-1155
- Vaswani, Ashish, et al. (2017). "Attention is all you need." *Advances in Neural Information Processing Systems* (NEURIPS)
- Radford, Alec. "Improving language understanding by generative pre-training." (2018)
- Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.
- Devlin, Jacob, et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*
- Belinkov and Glass (2019) "Analysis Methods in Neural Language Processing: A Survey". In *Transactions of ACL*, Volume 7, pages 49-72
- Srivastava, Aarohi, et al. (2023) "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models." In *Transactions on machine learning research*
- Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych (2024). "Holmes ⌕ A Benchmark to Assess the Linguistic Competence of Language Models". In *Transactions of the Association for Computational Linguistics*
- Goldberg, Yoav (2019). "Assessing BERT's syntactic abilities." arXiv preprint arXiv:1901.05287
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman (2020). "BLiMP: The Benchmark of Linguistic Minimal Pairs for English". In *Transactions of the Association for Computational Linguistics*
- Ciaccio C., Miaschi A., Dell'Orletta F. (2025). Evaluating Lexical Proficiency in Neural Language Models. In *Proceedings of ACL 2025*, July 27 - August 1, Vienna
- Giora, Rachel, et al. (2004) "Weapons of mass distraction: Optimal innovation and pleasure ratings." In *Metaphor and symbol* 19.2, 115-141.

Loss function

- Neural networks are trained to predict probability distributions on classes
- Intuitively, at each step, the probability that the model predicts the correct class is maximized
- The standard loss function is the **cross-entropy loss**
- Given:

$$p^* = (0, \ldots, 0, 1, 0, \ldots)$$ target distribution

$$p = (p_1, \ldots, p_K)$$ model's distribution

Loss function

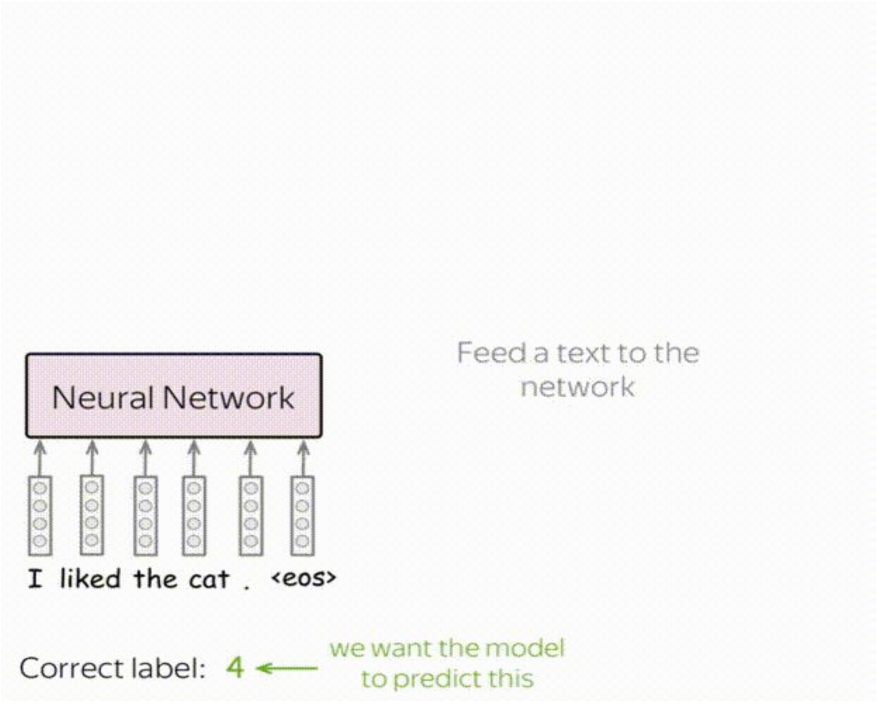
- Neural networks are trained to predict probability distributions on classes
- Intuitively, at each step, the probability that the model predicts the correct class is maximized
- The standard loss function is the **cross-entropy loss**
- Given:

$p^* = (0, \ldots, 0, 1, 0, \ldots)$ target distribution

$p = (p_1, \ldots, p_K)$ model's distribution

$$Loss(p^*, p) = -p^* \log(p) = -\sum_{i=1}^{K} p_i^* \log(p_i)$$

Loss function



Text Classification: https://lena-voita.github.io/nlp_course/text_classification.html

Neural Language Model (NLM)

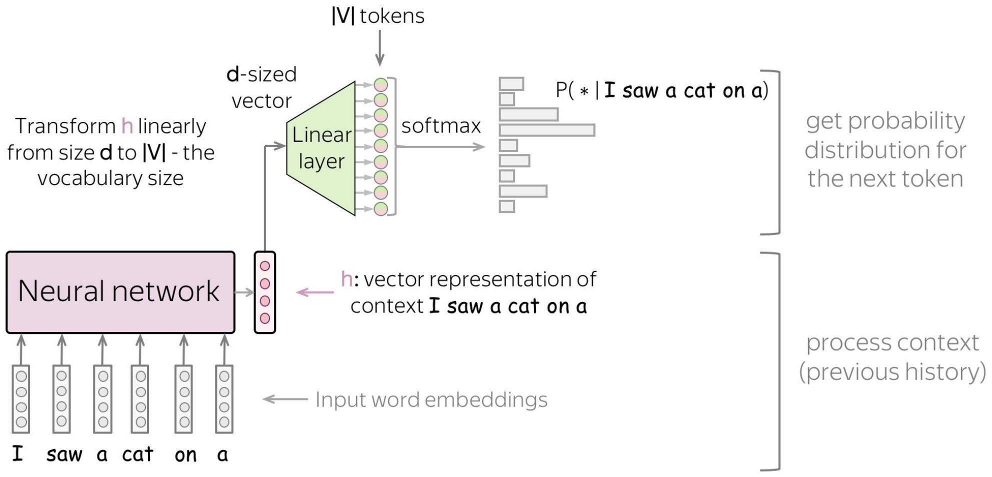
$$Loss(p^*, p) = -p^* \log(p) = -\sum_{i=1}^{K} p_i^* \log(p_i)$$

Neural Language Model (NLM)

$$Loss(p^*, p) = -p^* \log(p) = -\sum_{i=1}^{\cancel{K}\ |V|} p_i^* \log(p_i)$$

Neural Language Model (NLM)

$$Loss(p^*, p) = -p^* \log(p) = -\sum_{i=1}^{\cancel{K}\ |V|} p_i^* \log(p_i)$$



Neural Language Model (NLM)

$$Loss(p^*, p) = -p^* \log(p) = -\sum_{i=1}^{\cancel{K}\ |V|} p_i^* \log(p_i)$$