UNIVERSITY OF HELSINKI

Large Language Models and their interpretation

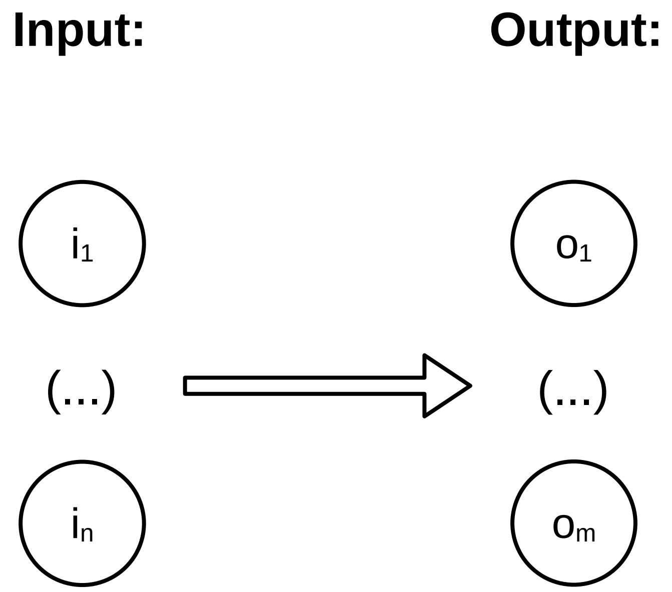
Tommi Buder-Gröndahl

Lecture structure

1. Neural networks: Basics
2. Recurrent neural networks (RNNs)
3. Attention
4. Transformer
5. Large Language Models (LLMs)
6. Interpreting LLMs

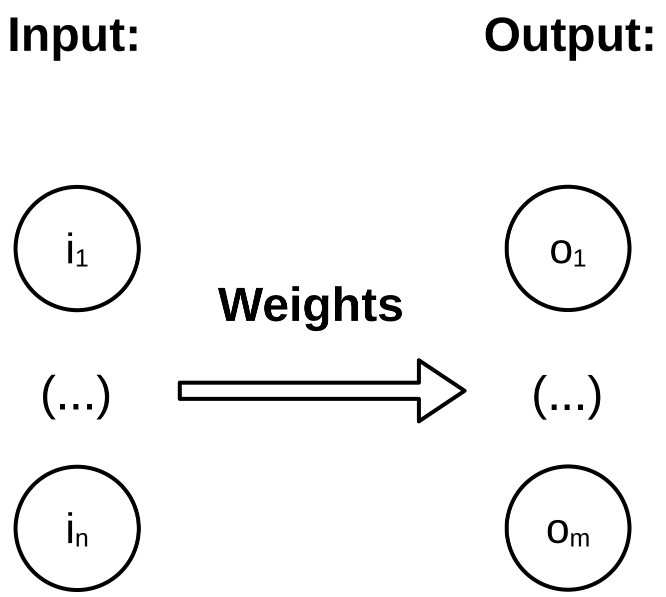
Neural networks: Basics

Neural network



Neural network

Input vector:
$\mathbf{i} = [i_1, \ldots, i_n]$



Output vector:
$\mathbf{o} = [o_1, \ldots, o_m]$

Neural network

Input vector:

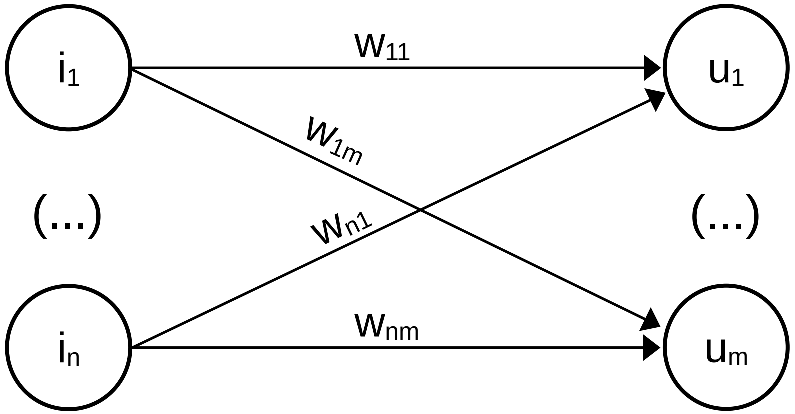
$\mathbf{i} = [i_1, \ldots, i_n]$

Weight matrix:

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \cdots & \cdots & \cdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix}$$

Weighted sums of inputs:

$\mathbf{u} = \mathbf{i}W = [u_1, \ldots, u_m]$



Neural network

Input vector:

$\mathbf{i} = [i_1, \ldots, i_n]$

Weight matrix:

$$W = \begin{bmatrix} w_{11} & \ldots & w_{1m} \\ \ldots & \ldots & \ldots \\ w_{n1} & \ldots & w_{nm} \end{bmatrix}$$

Weighted sums of inputs:

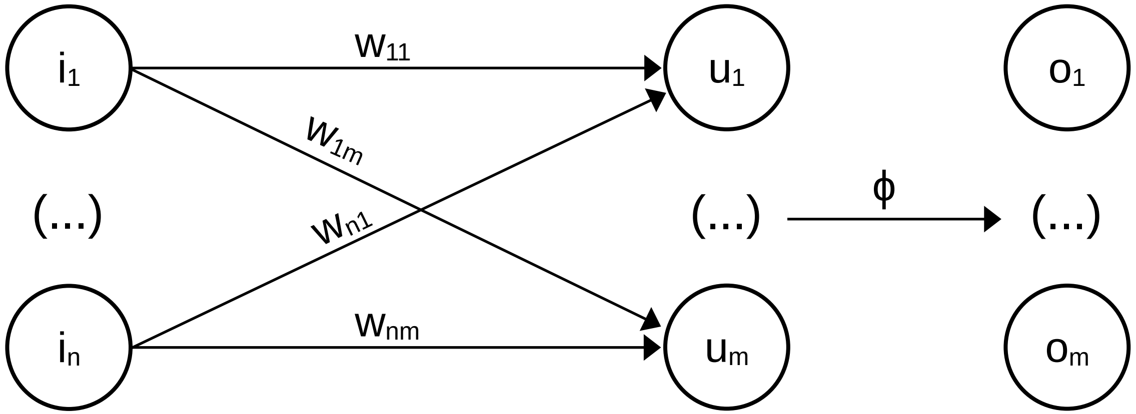
$\mathbf{u} = \mathbf{i}W = [u_1, \ldots, u_m]$

Activation function:

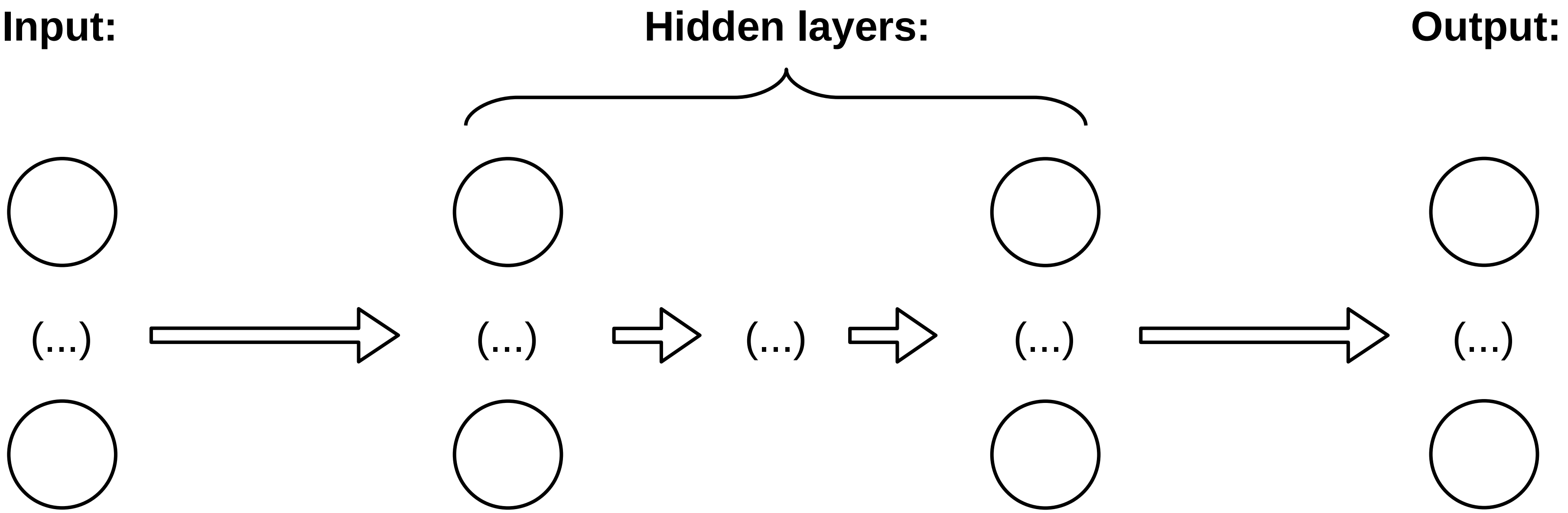
$\phi \in \{\sigma, \tanh, \text{ReLu}, \ldots\}$

Output:

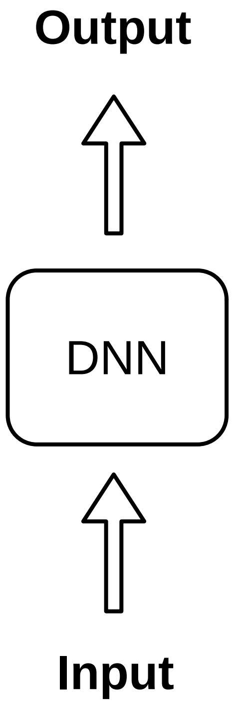
$\mathbf{o} = \phi(\mathbf{u}) = [o_1, \ldots, o_m]$



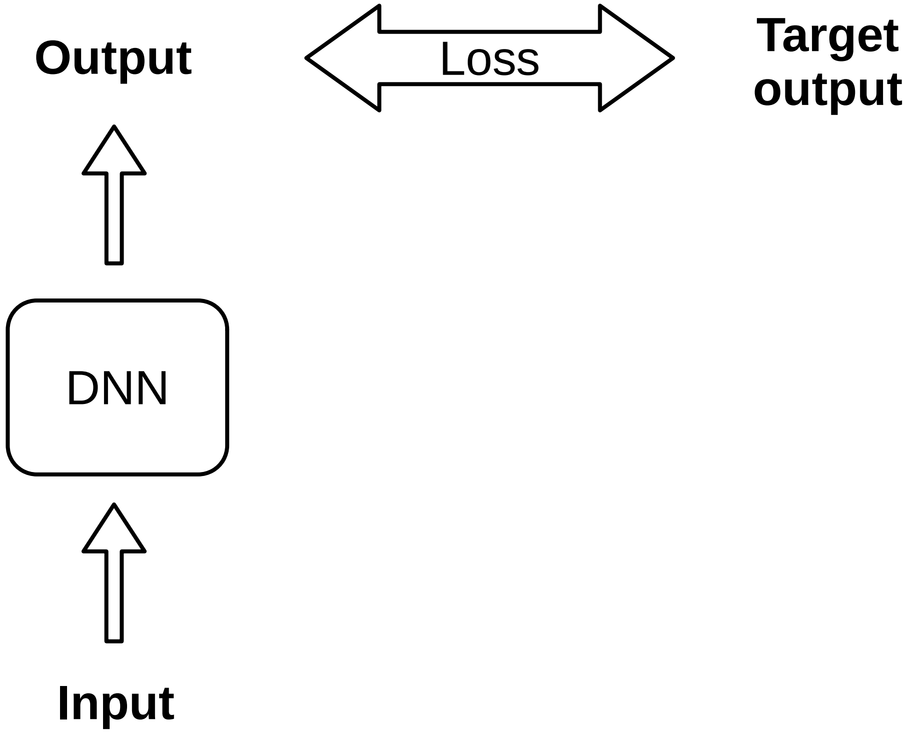
Deep Neural Network (DNN)



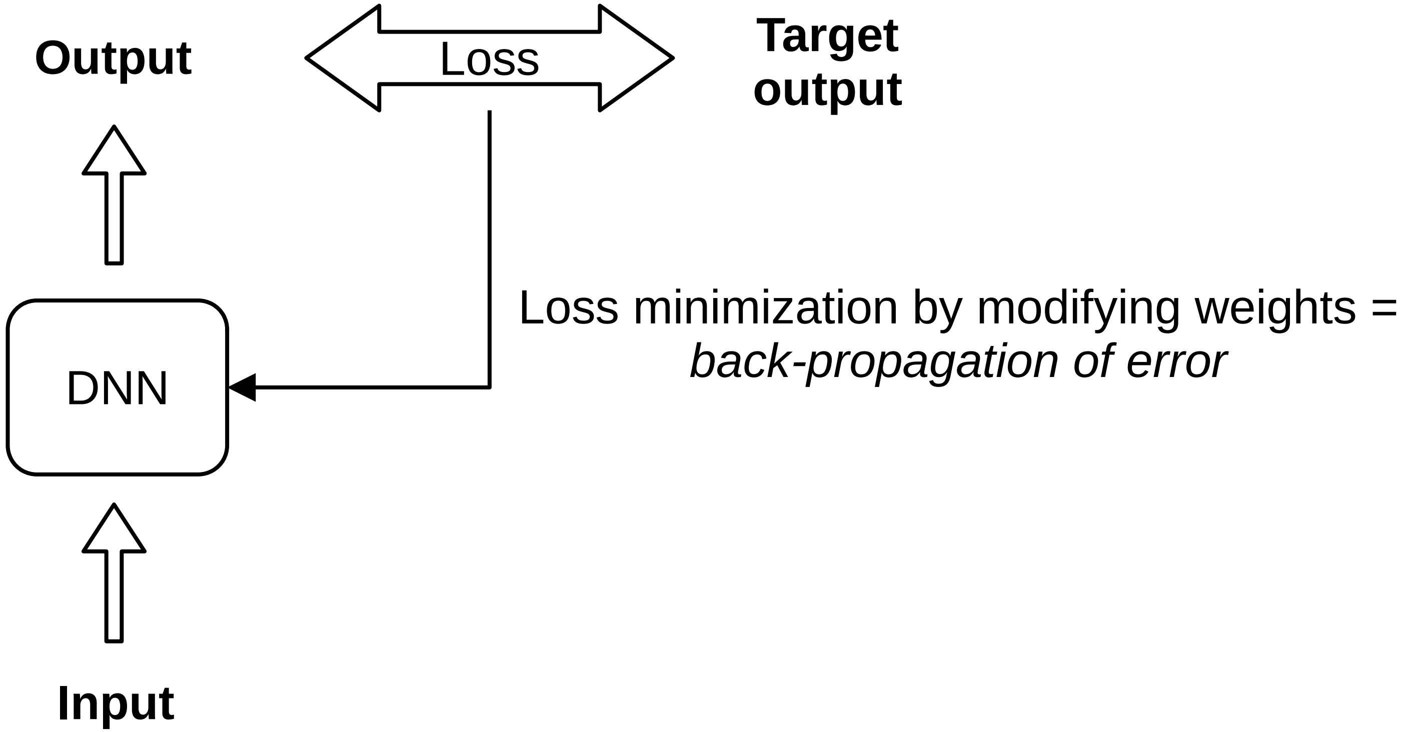
Training a DNN



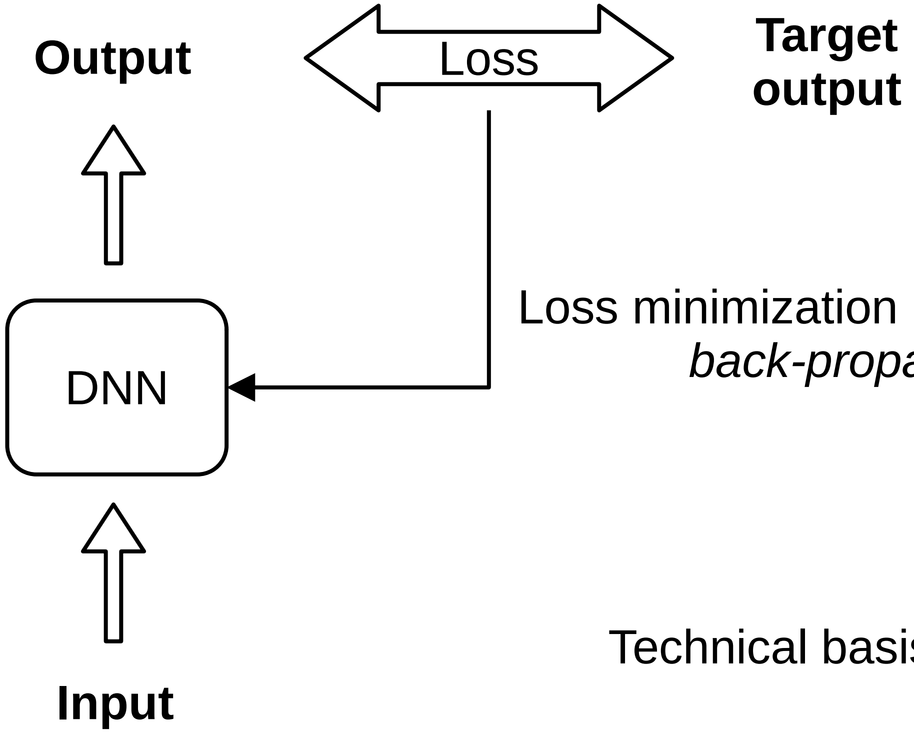
Training a DNN



Training a DNN



Training a DNN



Loss minimization by modifying weights = *back-propagation of error*

Technical basis: *gradient descent*

Recurrent neural networks (RNNs)

Sequential data: influence of context

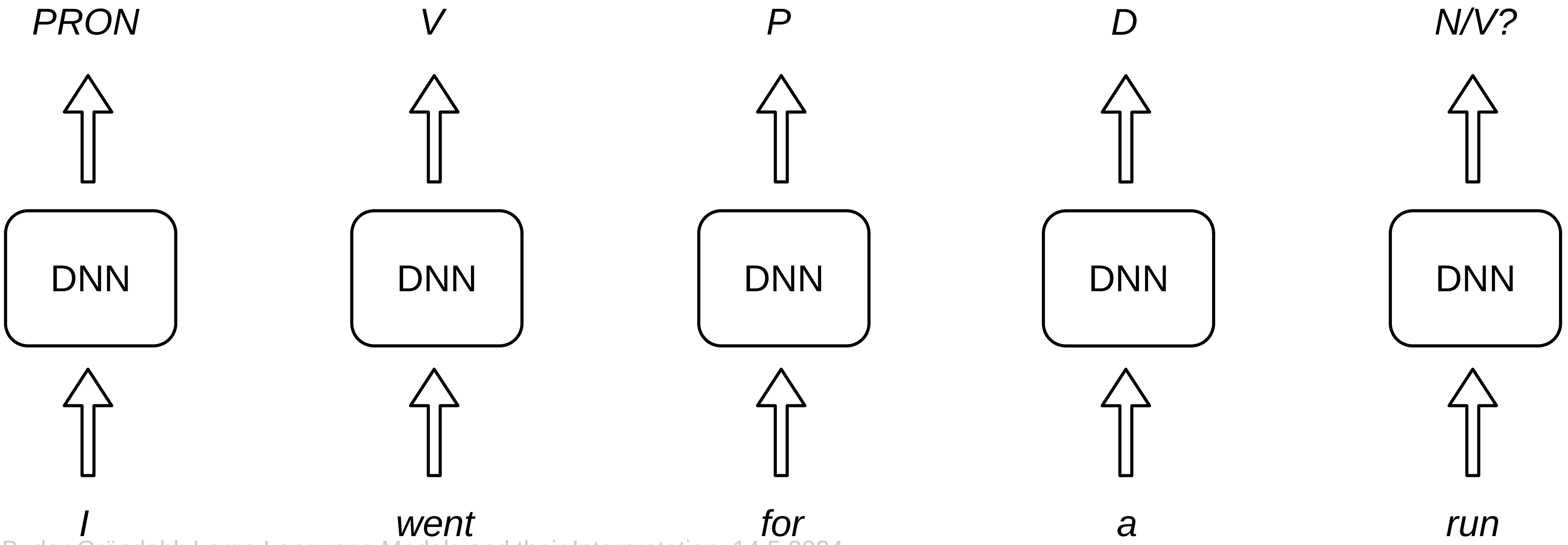
*I **run***

*I went for a **run***

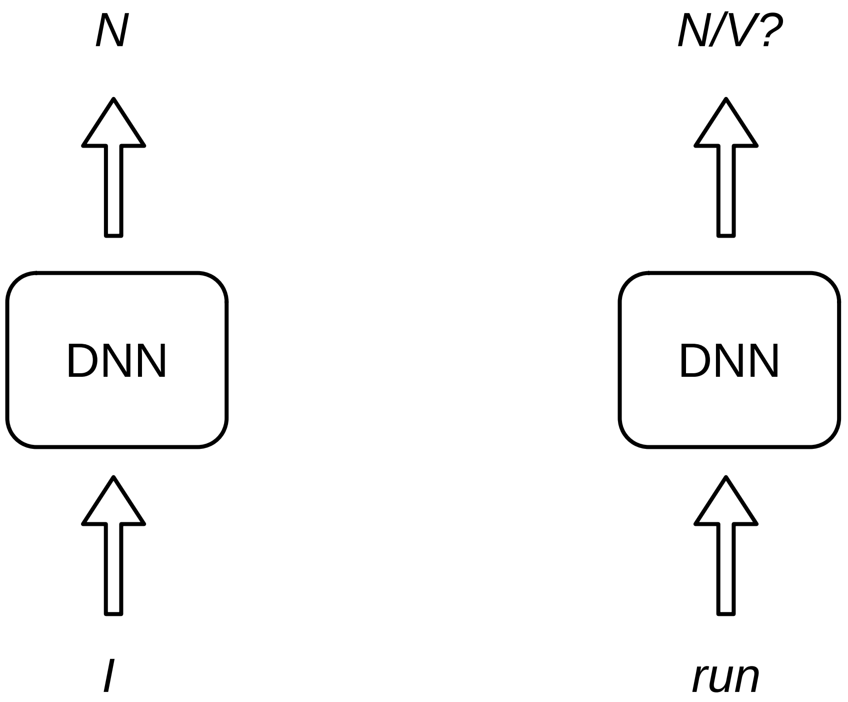
Sequential data: influence of context

I ***run***

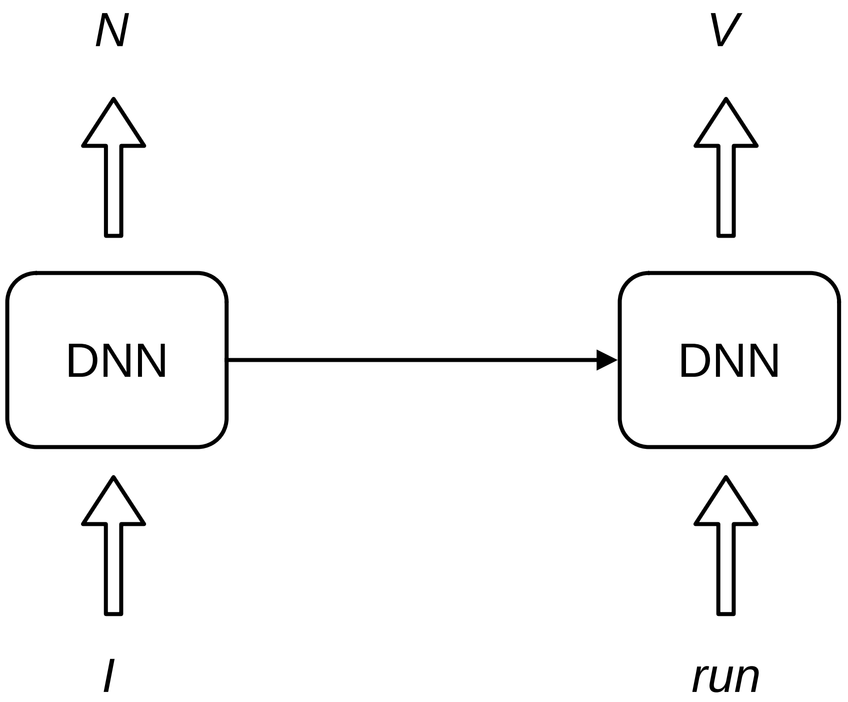
I went for a ***run***



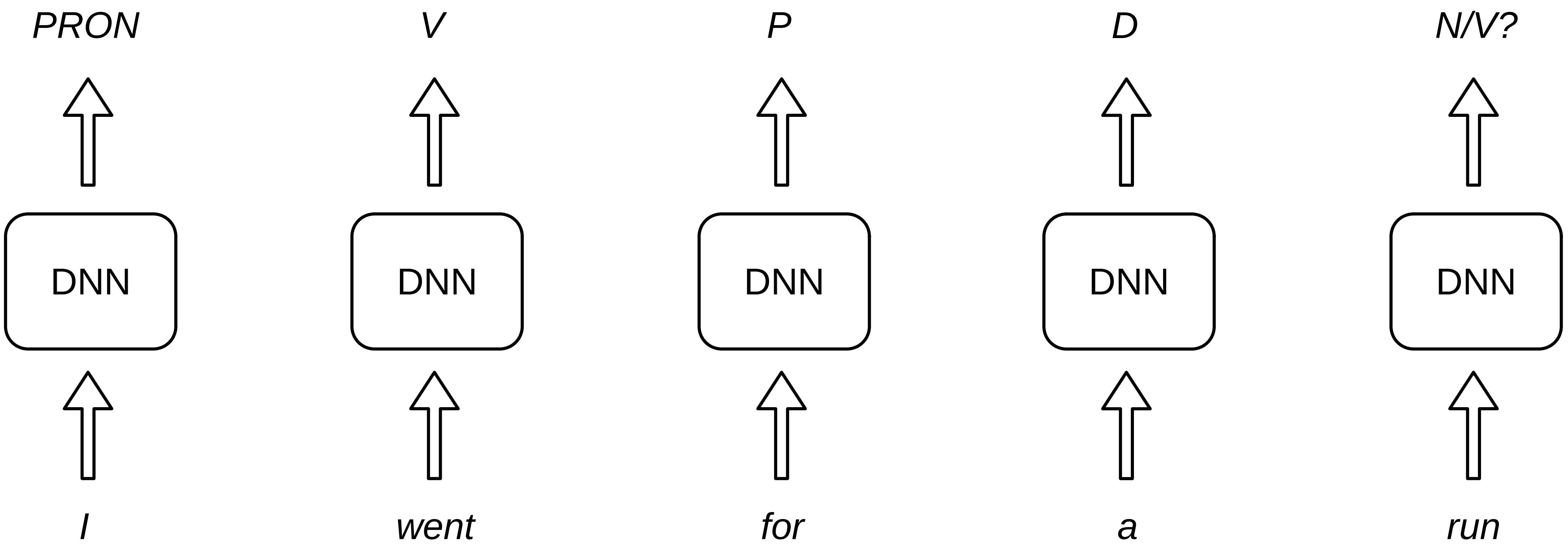
Recurrent connections



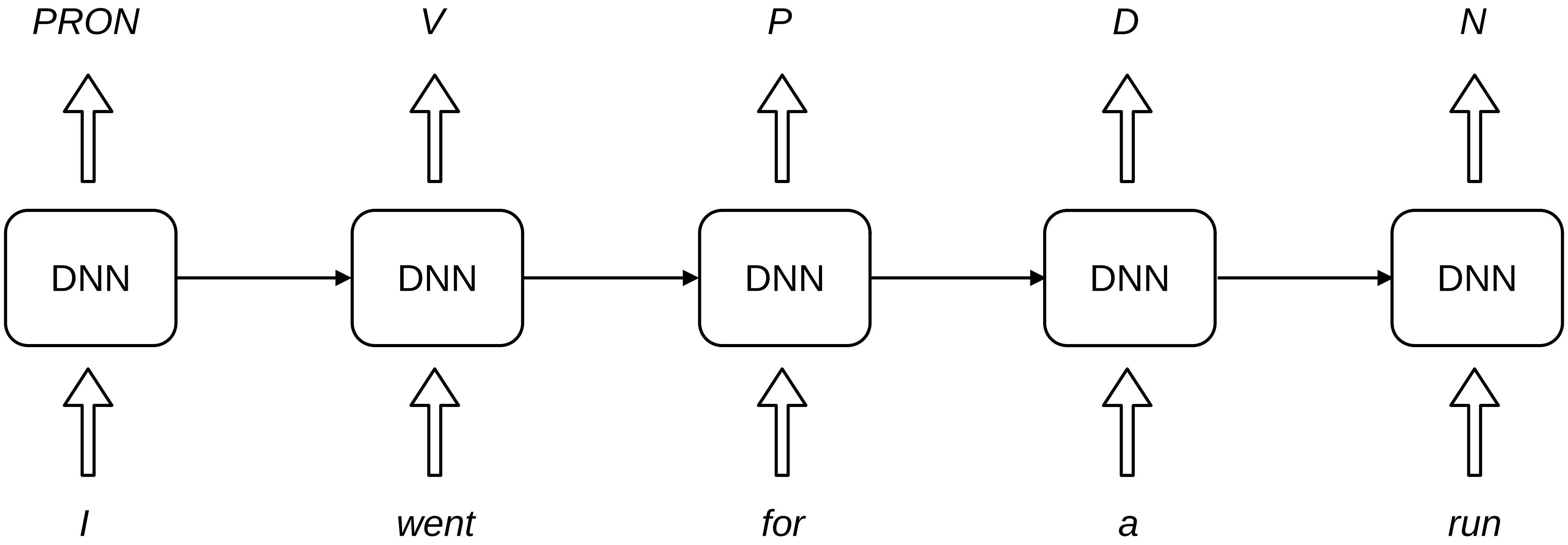
Recurrent connections



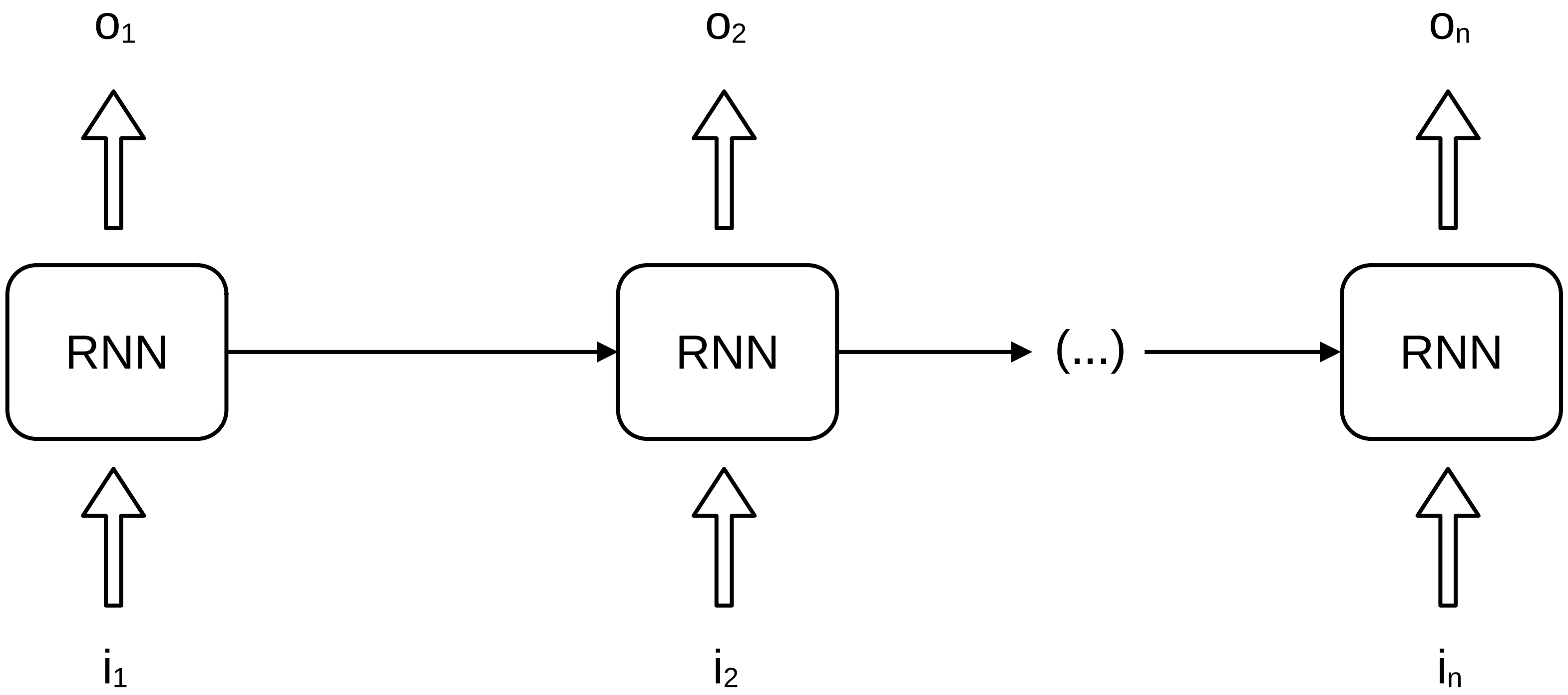
Recurrent connections



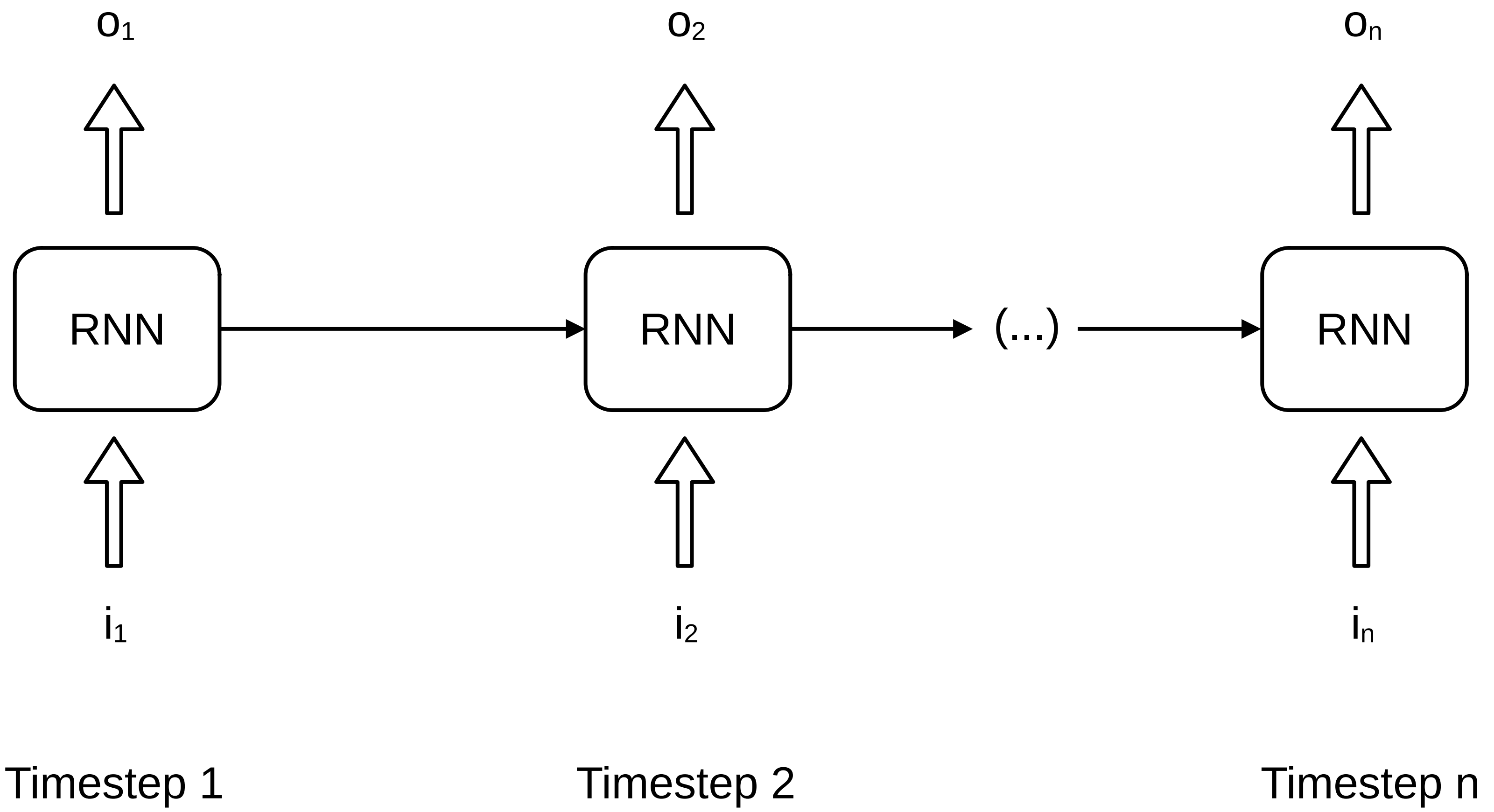
Recurrent connections



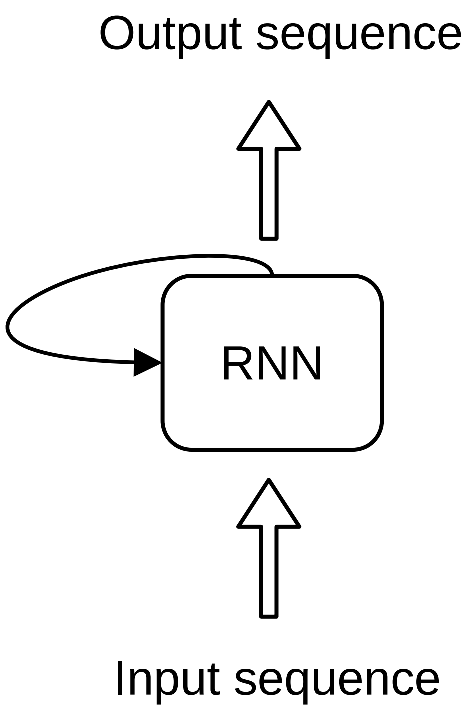
Recurrent Neural Network (RNN)



Recurrent Neural Network (RNN)



Recurrent Neural Network (RNN)

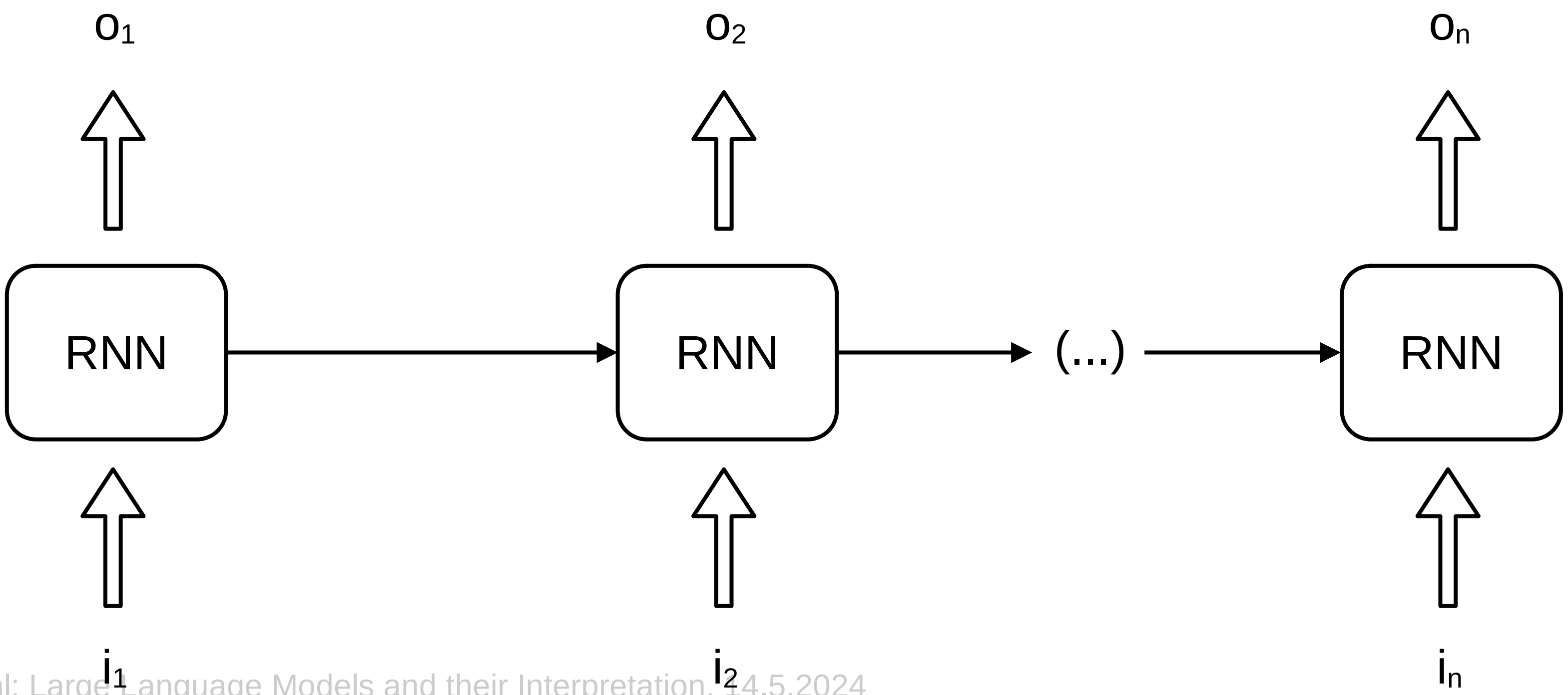


Encoder-decoder RNNs + Attention

Encoder-decoder RNN

Basic RNN maps inputs to outputs 1-1

- Part-of-speech tagging
- Spelling correction
- (...)



Encoder-decoder RNN

Basic RNN maps inputs to outputs 1-1

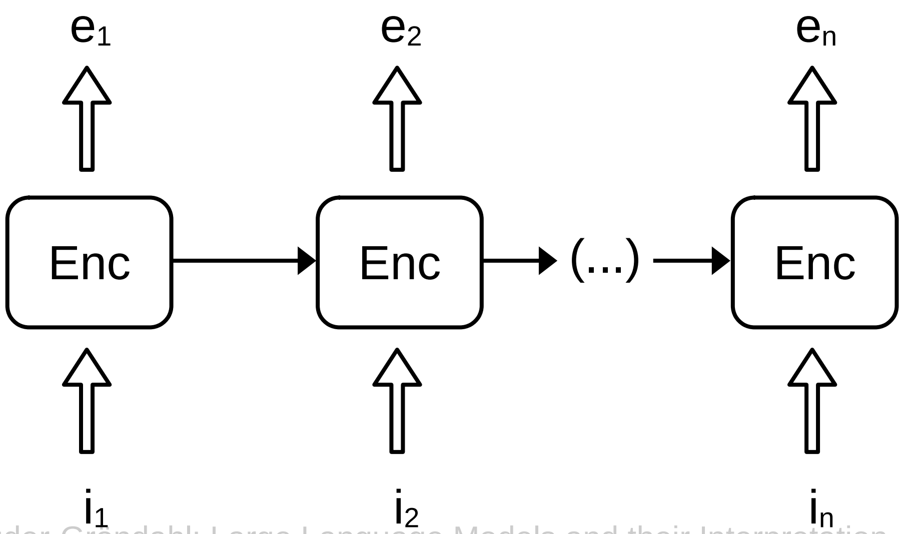
- Part-of-speech tagging
- Spelling correction
- (...)

But we often want more flexible input-output mappings: e.g. machine translation

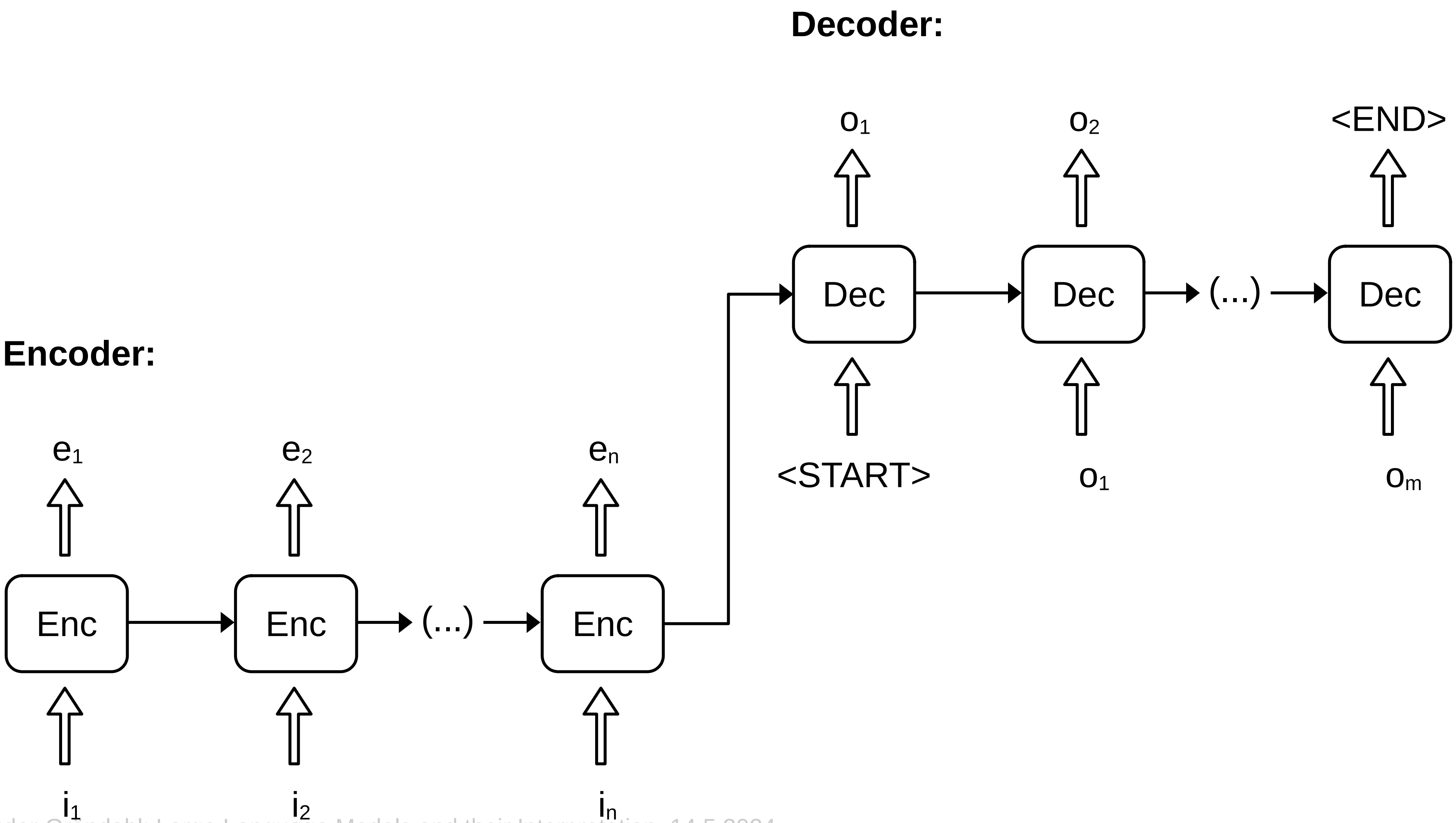
- Grammatical and lexical variation → different number of words between sentences
- Word-order variation
- (...)

Encoder-decoder RNN

Encoder:



Encoder-decoder RNN



Vanishing gradient

Problem

- Older encoder inputs have less effect than more recent ones
- Harder to find long-distance dependencies

*The **dog** that chased two cats **is** brown*

Vanishing gradient

Problem

- Older encoder inputs have less effect than more recent ones
- Harder to find long-distance dependencies

*The **dog** that chased two cats **is** brown*

Long short-term memory (LSTM)

- More complex RNN to alleviate the vanishing gradient problem
- Two distinct hidden states updated differently, allowing better retention of information

Vanishing gradient

Problem

- Older encoder inputs have less effect than more recent ones
- Harder to find long-distance dependencies

*The **dog** that chased two cats **is** brown*

Long short-term memory (LSTM)

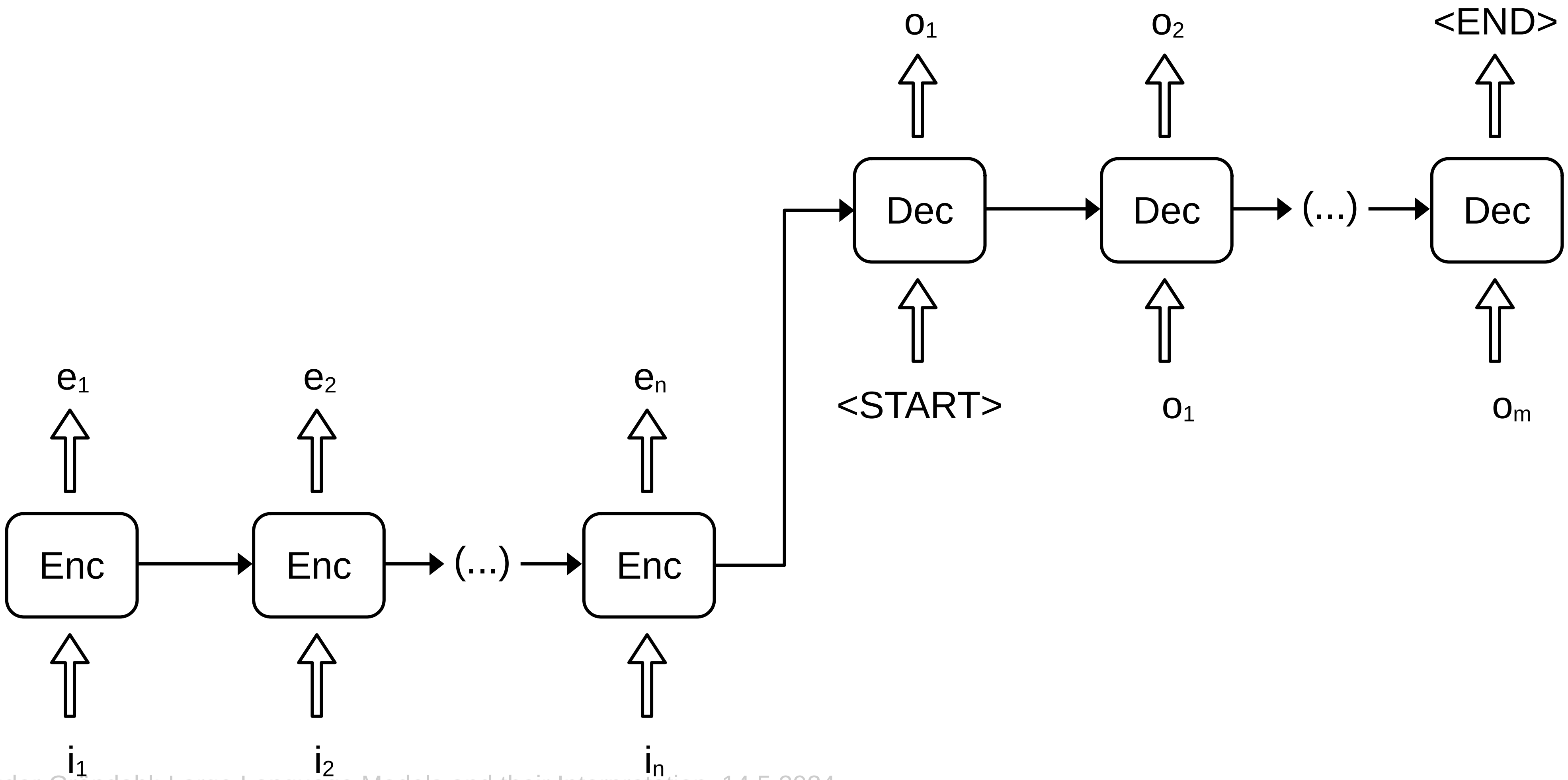
- More complex RNN to alleviate the vanishing gradient problem
- Two distinct hidden states updated differently, allowing better retention of information
- *Bidirectional* LSTMs*:* reading input from front-to-back and back-to-front, combining results
- Gated recurrent unit (GRU): similar to LSTM but simpler

Vanishing gradient

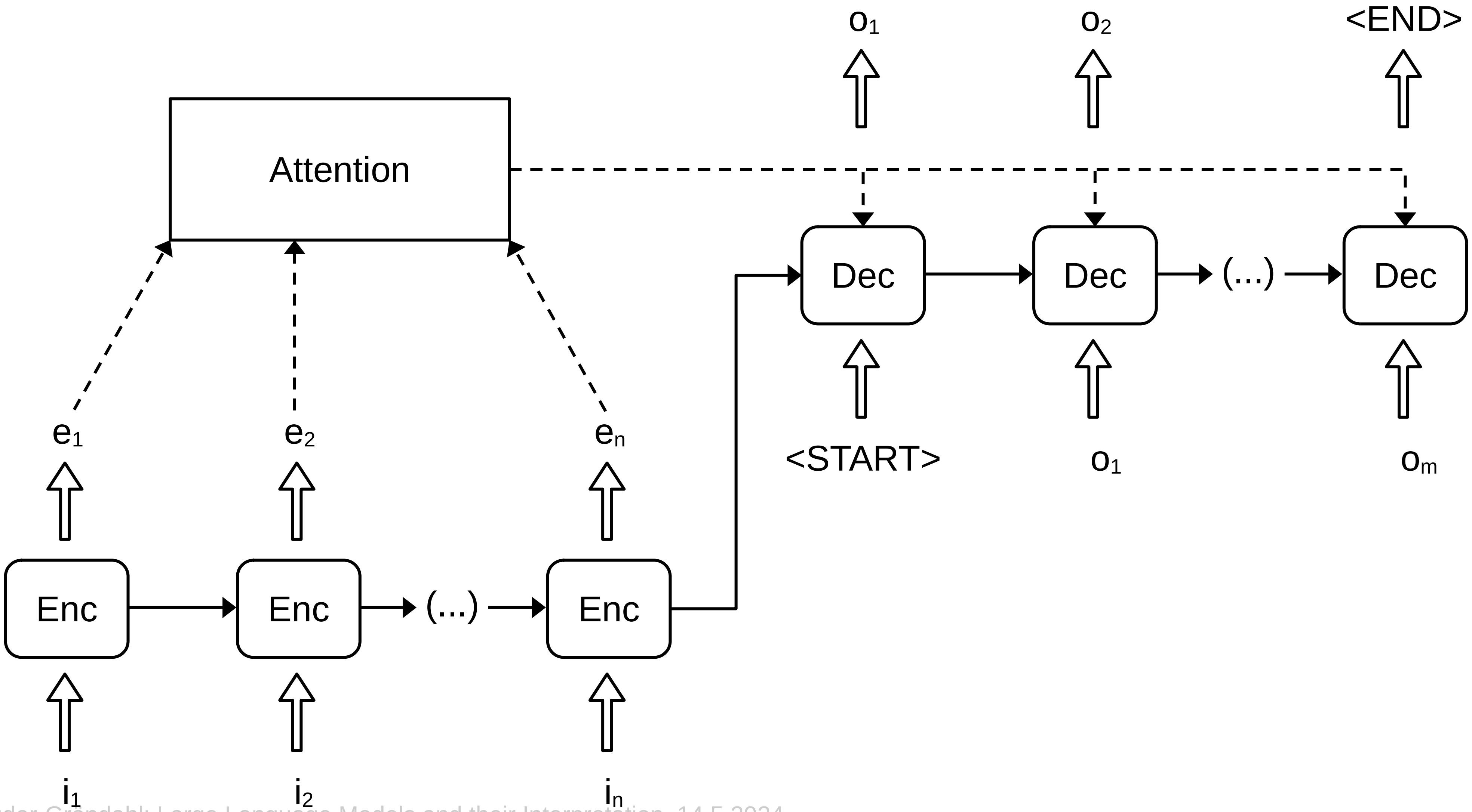
Attention

- Calculates a probability distribution across all encoding steps
- Combines all encoder outputs weighted by the probability distribution
- Using the result as additional decoder input

Encoder-decoder RNN



Encoder-decoder RNN + Attention



Transformer

Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

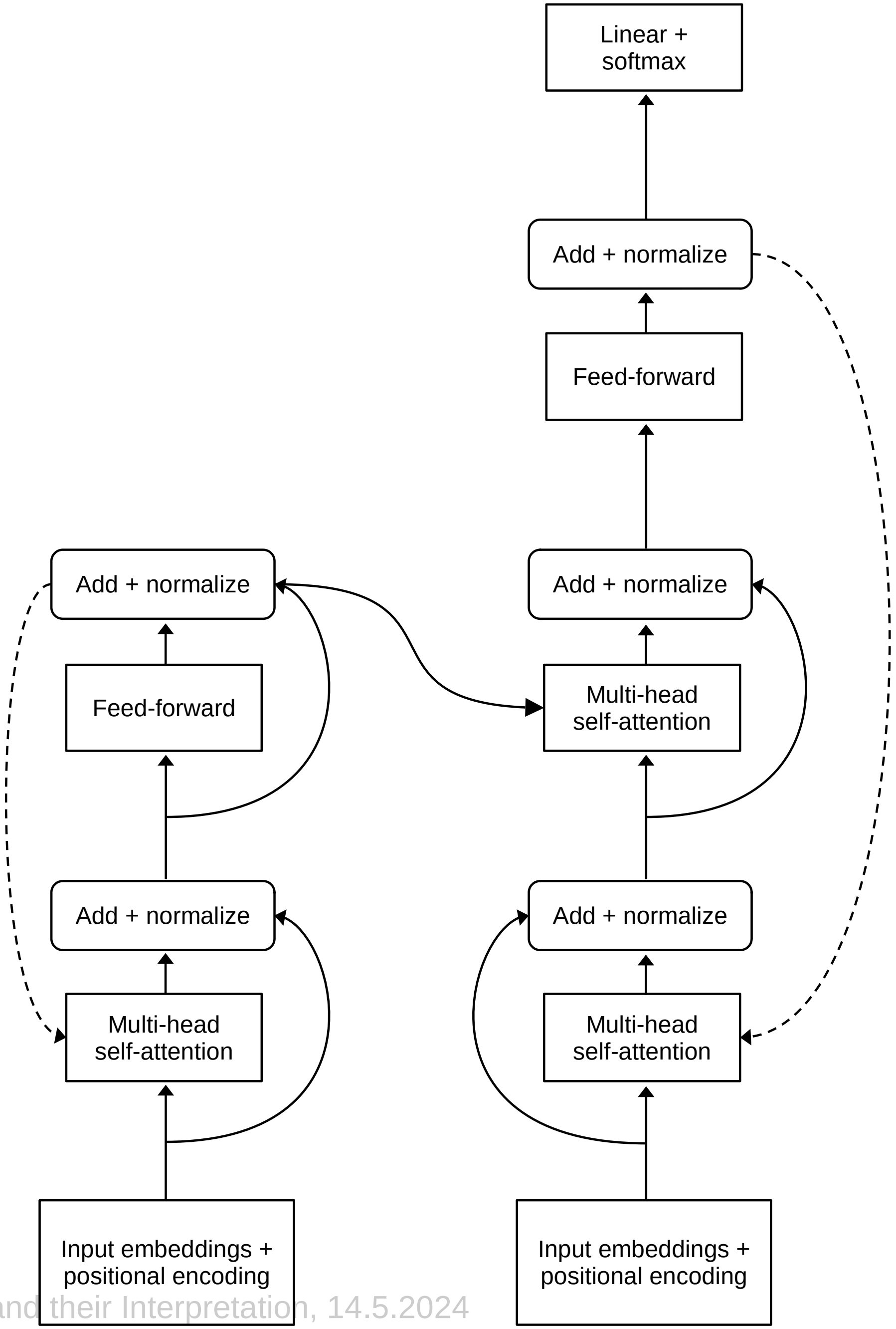
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

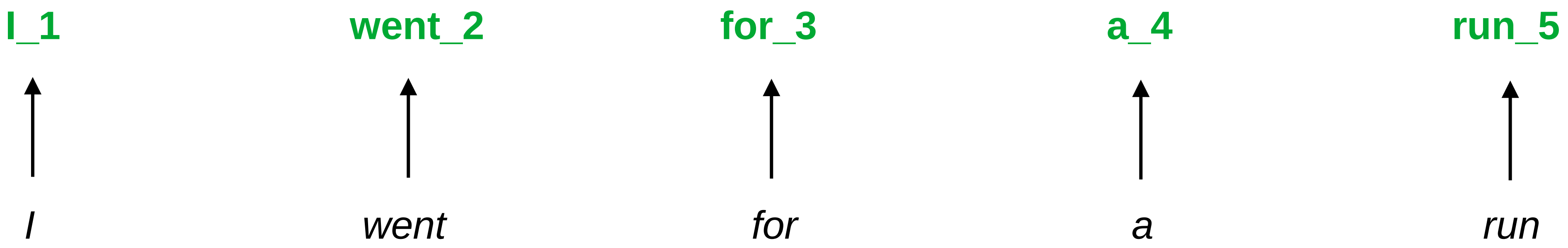
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Transformer



Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

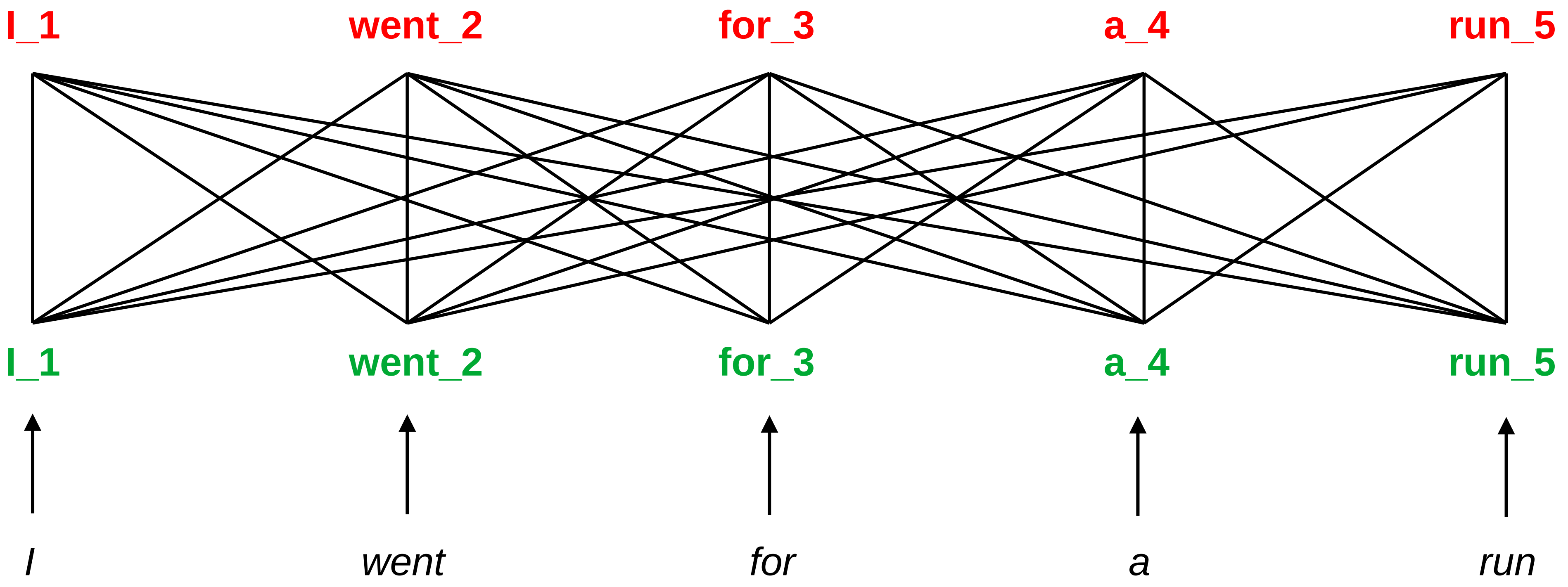


Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Input goes through **multi-head self-attention**, creating new **contextual encodings** for each token.

Contextual encoding for each token is calculated from previous embeddings of each token.



Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Input embeddings +
positional encoding

Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Each Transformer layer contains (several) **attention heads.**

An attention head contains three weight matrices:

query weights: W_q

key weigths: W_k

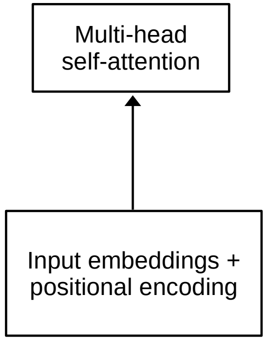
value weights: W_v

Input embedding x_i is multiplied by each matrix, which yields:

query-vector: $q_i = x_i W_q$

key-vector: $k_i = x_i W_k$

value-vector: $v_i = x_i W_v$



Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Each Transformer layer contains (several) **attention heads.**

An attention head contains three weight matrices:

query weights: W_q
key weigths: W_k
value weights: W_v

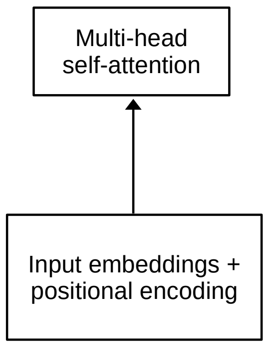
Input embedding x_i is multiplied by each matrix, which yields:

query-vector: $q_i = x_i W_q$
key-vector: $k_i = x_i W_k$
value-vector: $v_i = x_i W_v$

Attention between inputs *i* and *j*:

$$a_{ij} = softmax\left(\frac{q_i . k_j}{\sqrt{d_k}}\right)$$

(d_k = dimensionality of k_j)



Output for input *i* = sum of all v_j weighted with a_{ij} (**contextual encoding**)

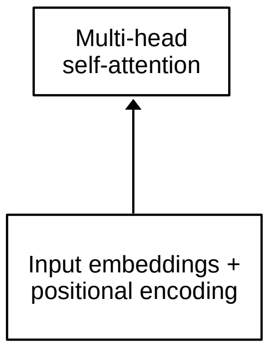
Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Multi-head self-attention:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q: query matrix
K: key matrix
V: value matrix

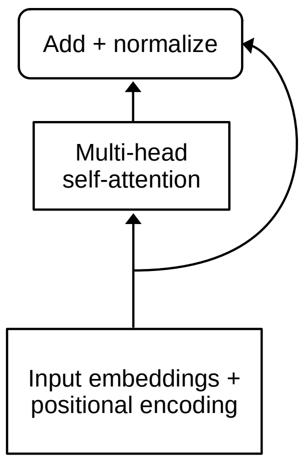


Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Input goes through **multi-head self-attention**.

Outputs of attention heads are combined (+ **residual connections**).



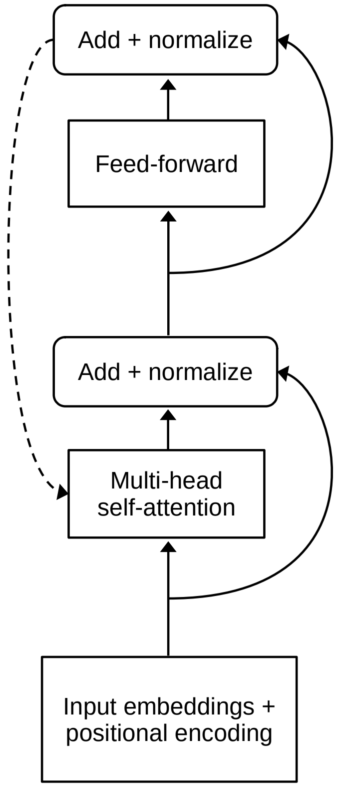
Transformer

Each input word has an **embedding**, which is combined with **positional encoding**.

Input goes through **multi-head self-attention**.

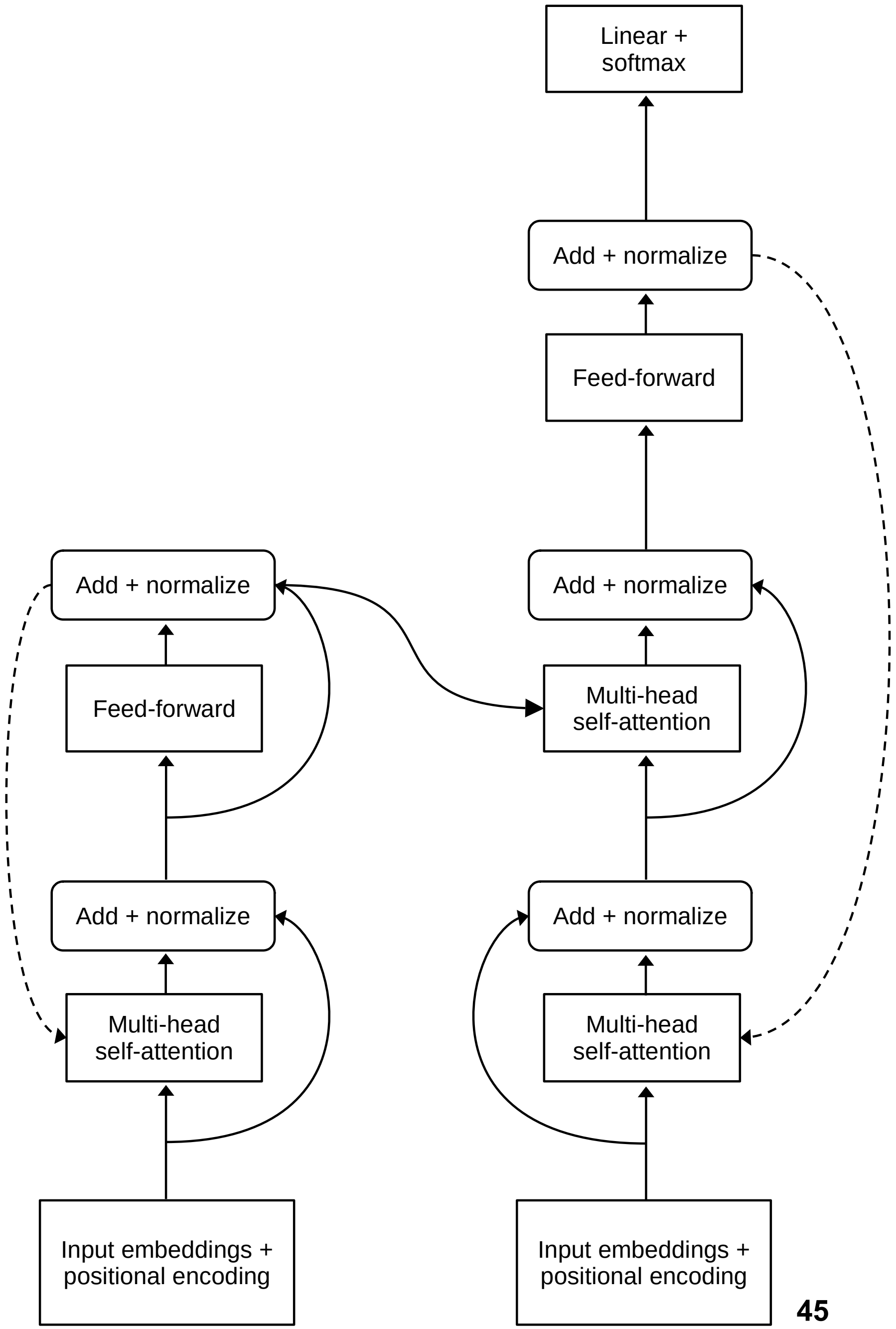
Outputs of attention heads are combined
(+ **residual connections**).

Output functions as input to a **feed-forward** network
(+ **residual connections**).

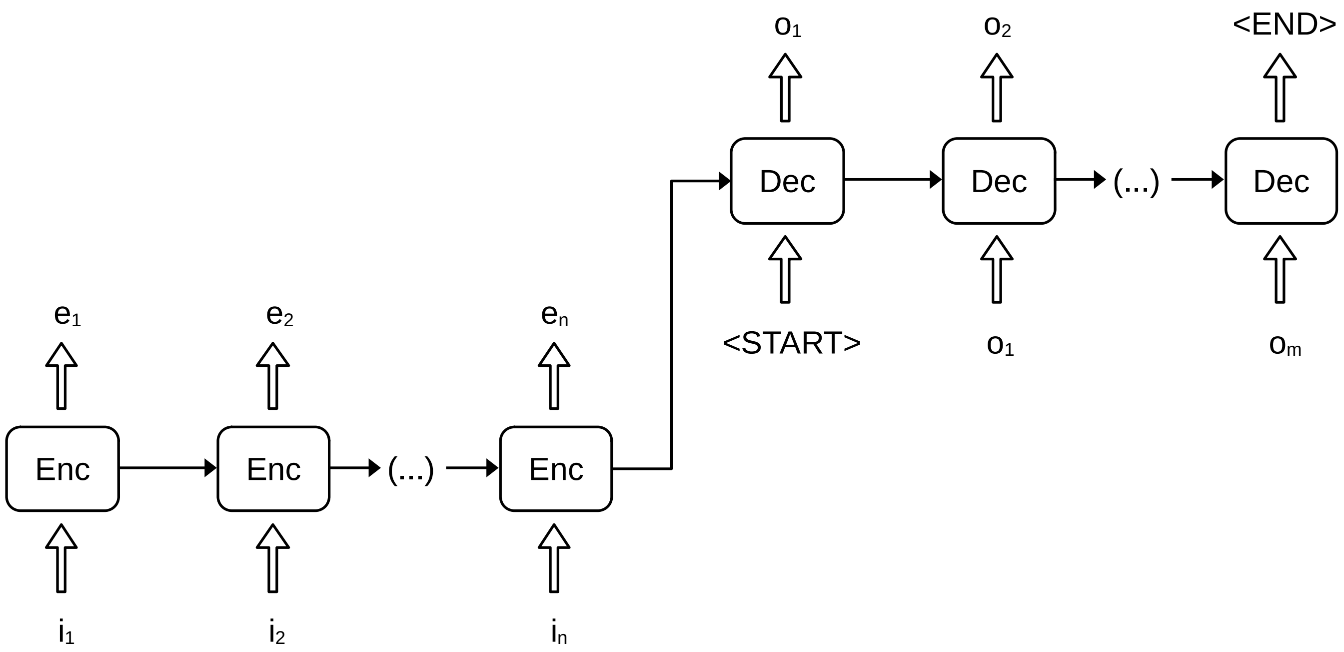


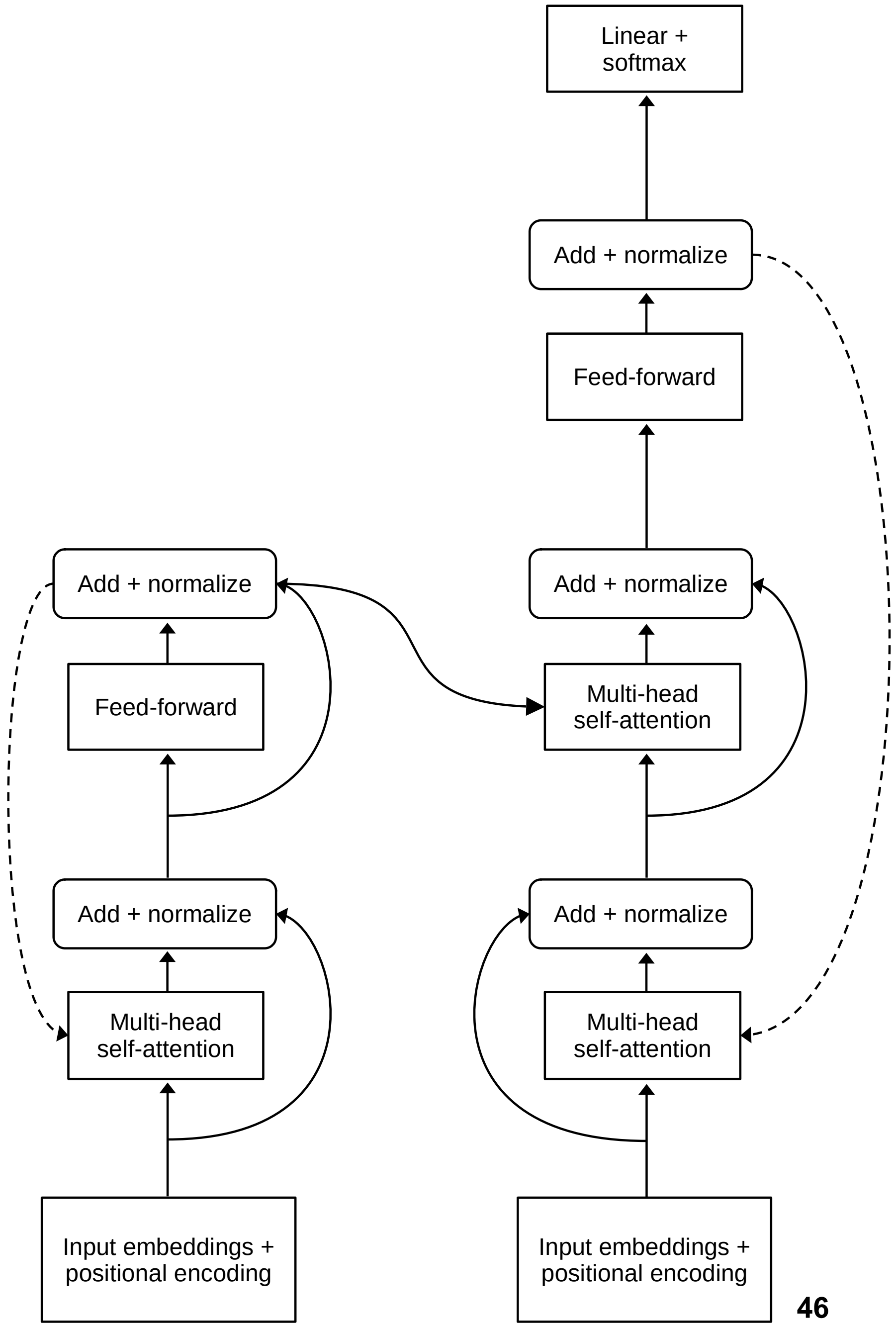
Transformer

Encoder-decoder Transformer: the decoder is like the encoder, but gets additional input via **encoder-decoder attention**



RNN vs. Transformer?





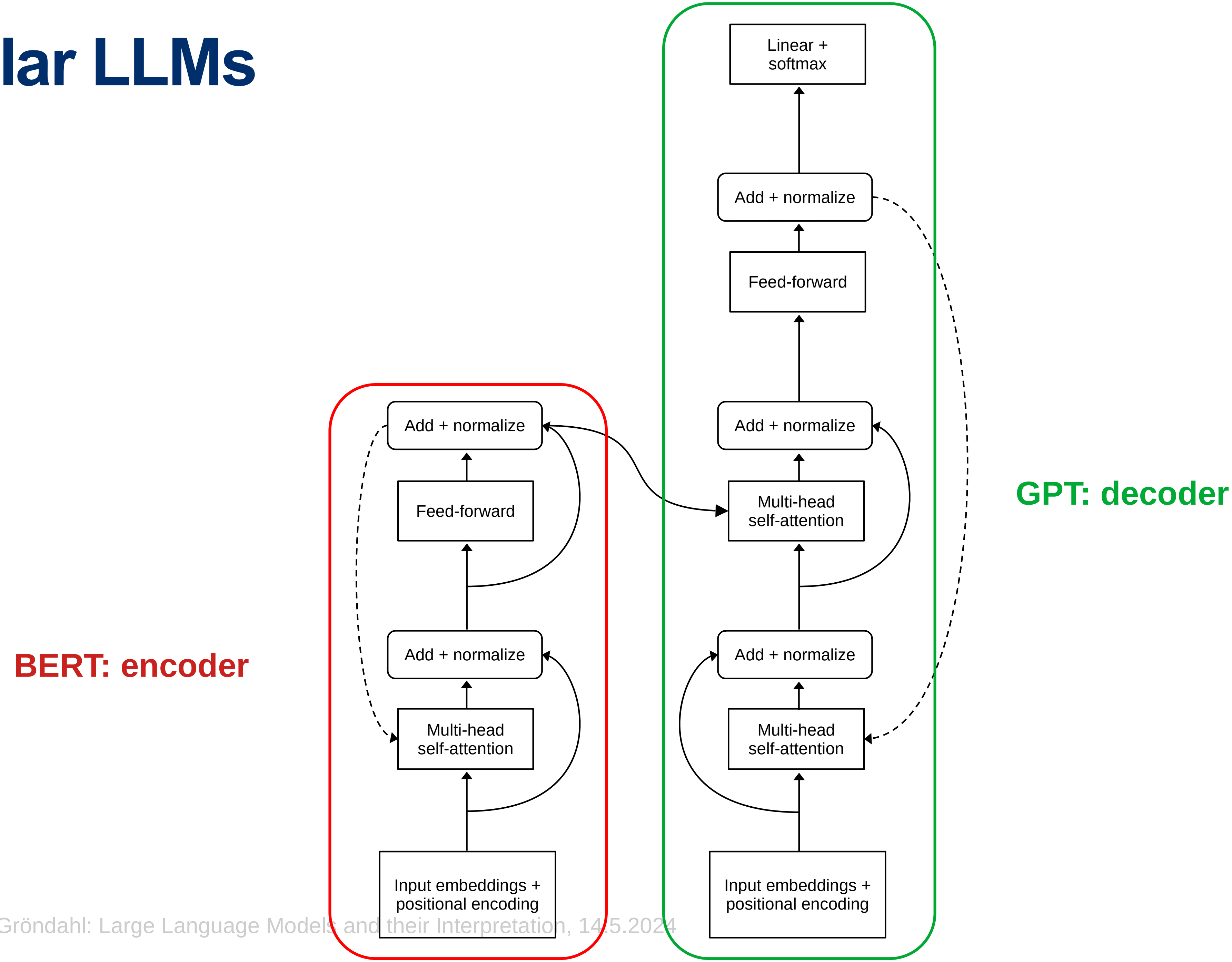
RNN vs. Transformer?

RNN	Transformer
Based on recurrent connections	No recurrent connections
Attention is a useful addition	Fully Attention-based
Goes through the input one token at a time	Goes through all tokens in parallel
Generates one representation of the whole input (last encoding step)	Generates a separate encoding for each input token
Order between tokens arises indirectly via processing steps	Positional encoding added to each input token separately
Long-distance dependencies are especially challenging (vanishing gradient)	Distance between tokens has no direct impact on the strength of their connection

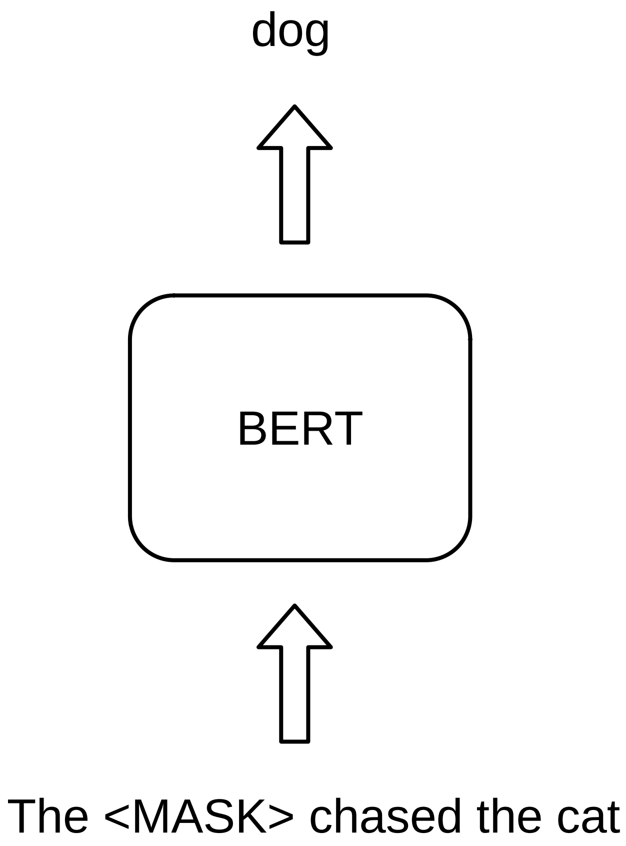
Input goes through multi-head self-attention.

Large Language Models (LLMs)

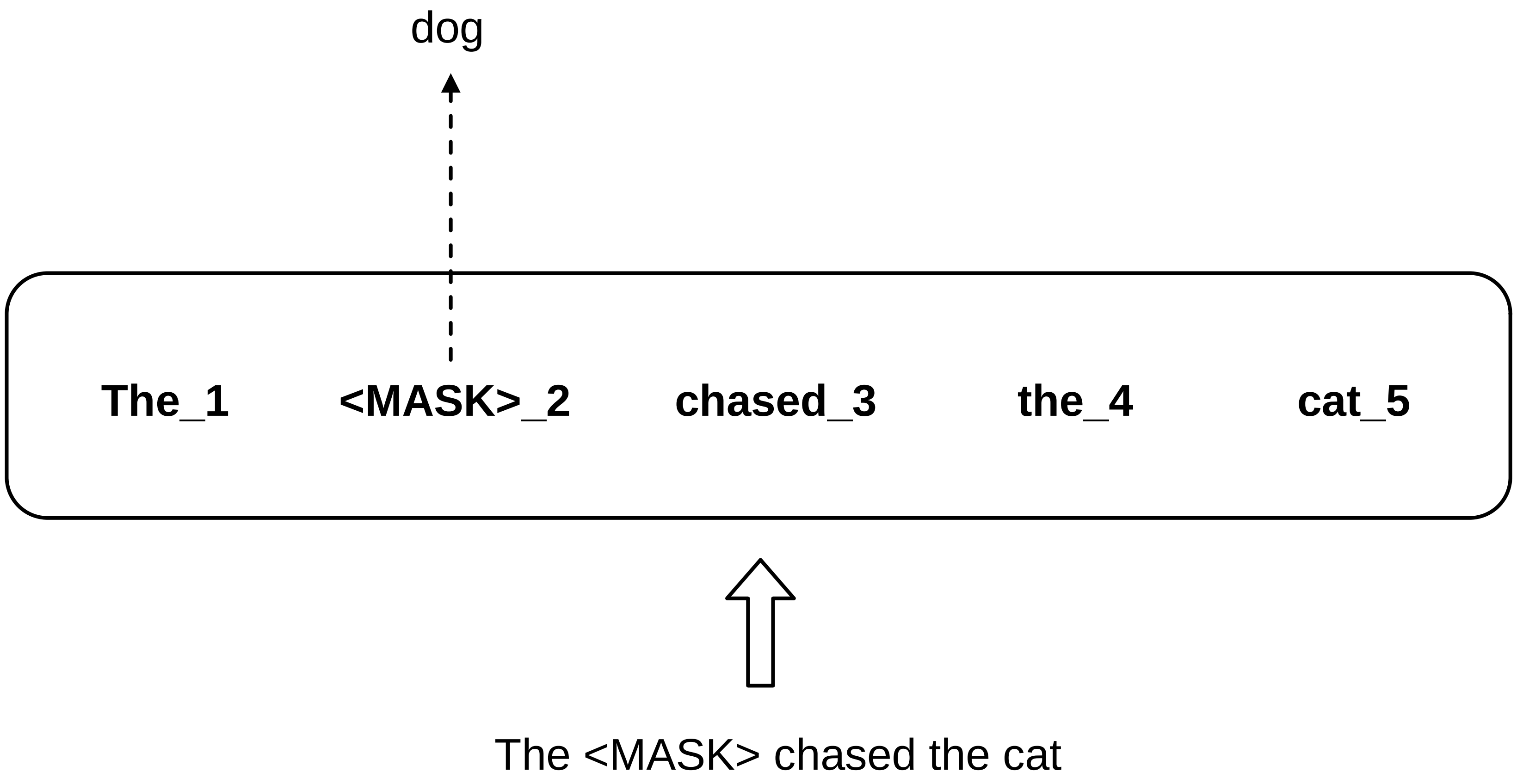
Popular LLMs



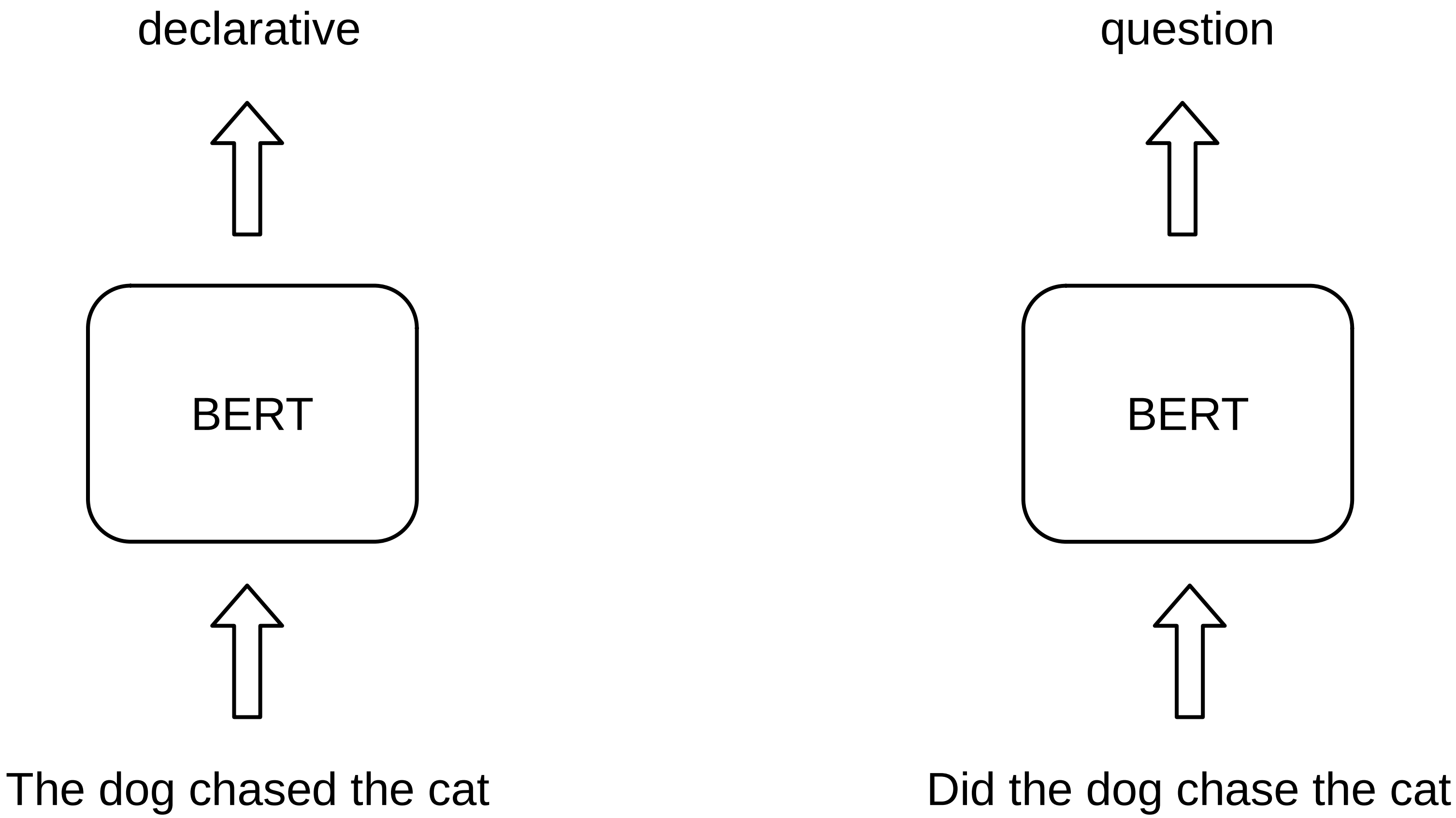
BERT: predicting masked tokens



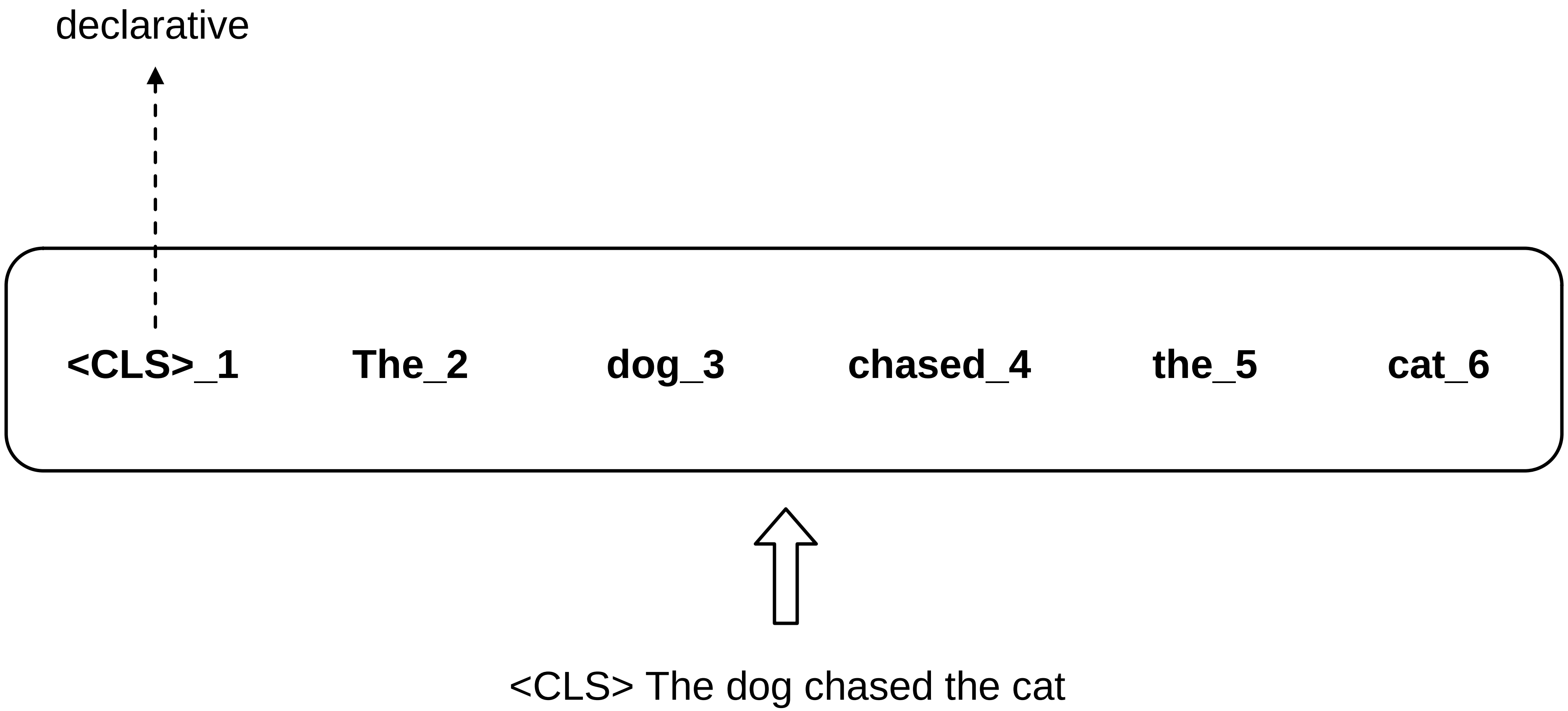
BERT: predicting masked tokens



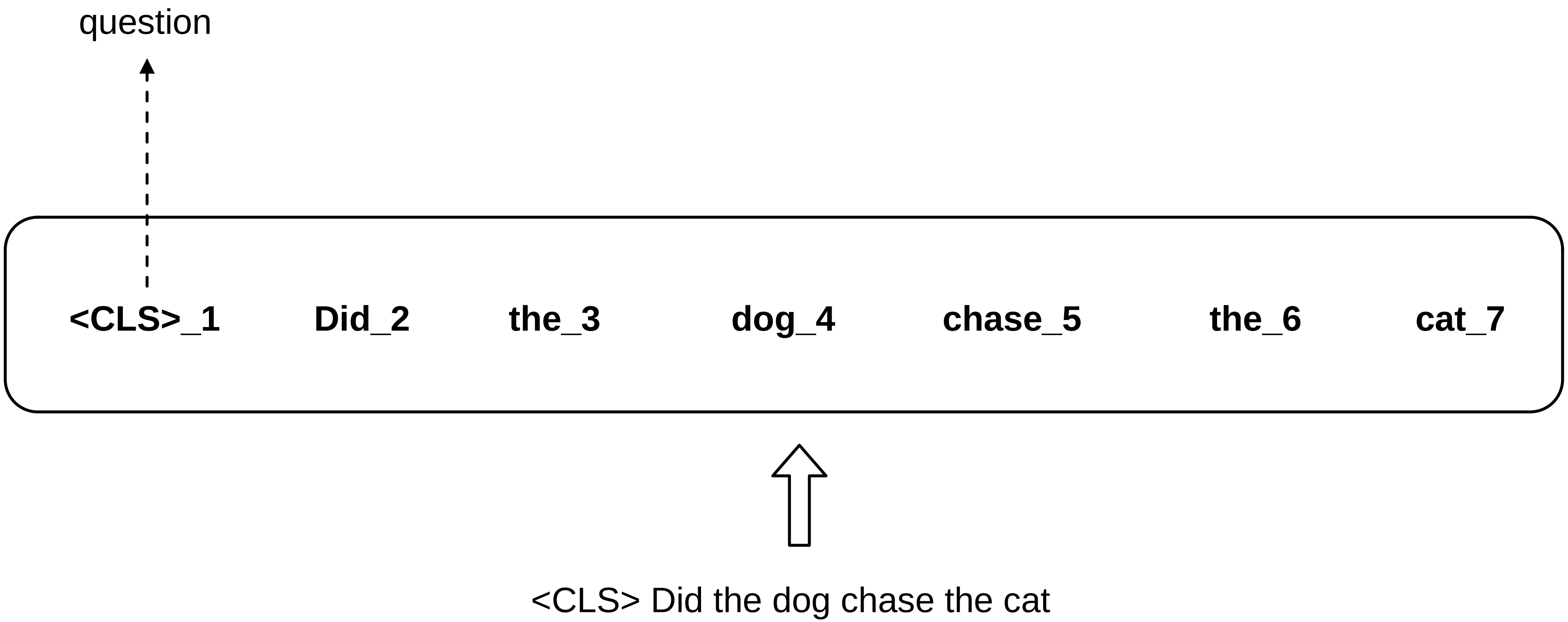
BERT: classifying whole texts



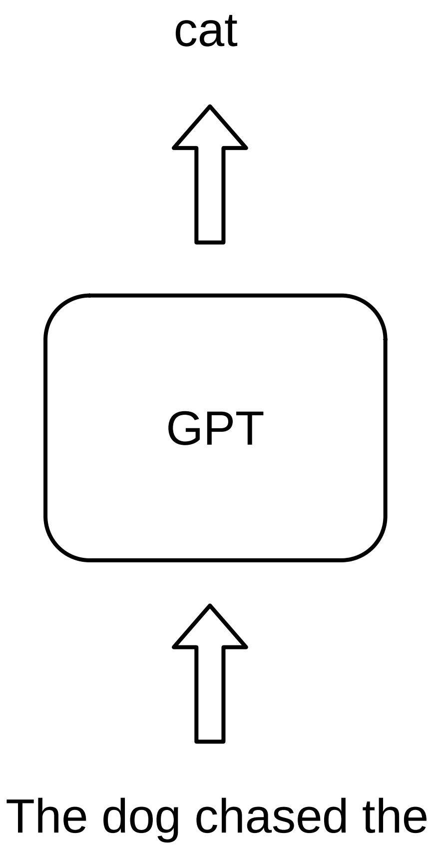
BERT: classifying whole texts



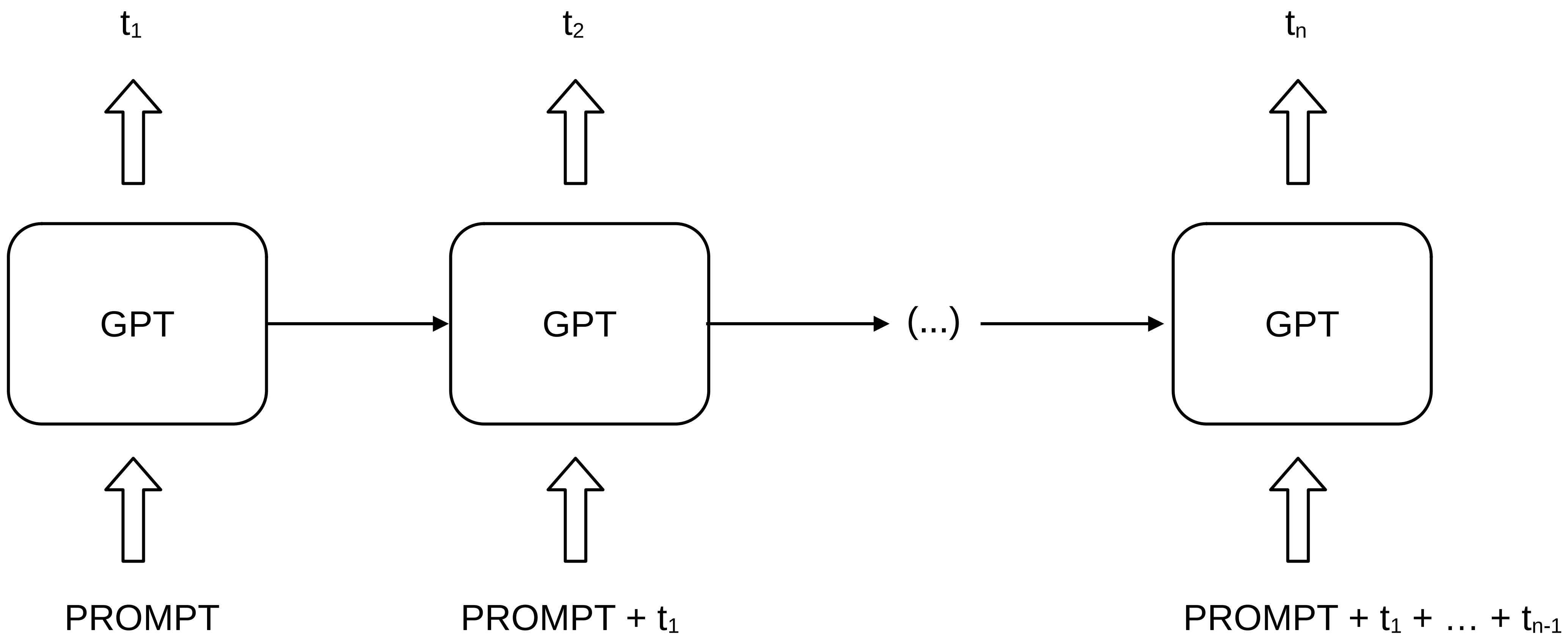
BERT: classifying whole texts



GPT: predicting the next token



GPT: predicting the next token



BERT vs. GPT

	BERT (*Bidirectional Encoder Representations from Transformers*)	**GPT** (*Generative Pre-trained Transformer*)
Architecture	Transformer-encoder	Transformer-decoder
Input	Text	Prompt + prior output
Output	Encoding of each token	Next token
Training	Predicting masked tokens	Predicting upcoming text

LLM variants

Comparison	BERT October 11, 2018	RoBERTa July 26, 2019	DistilBERT October 2, 2019	ALBERT September 26, 2019
Parameters	**Base:** 110M **Large:** 340M	**Base:** 125 **Large:** 355	**Base:** 66	**Base:** 12M **Large:** 18M
Layers / Hidden Dimensions / Self-Attention Heads	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 6 / 768 / 12	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16
Training Time	**Base:** 8 x V100 x 12d **Large:** 280 x V100 x 1d	1024 x V100 x 1 day (4-5x more than BERT)	**Base:** 8 x V100 x 3.5d (4 times less than BERT)	[not given] **Large:** 1.7x faster
Performance	Outperforming SOTA in Oct 2018	88.5 on GLUE	97% of BERT-base's performance on GLUE	89.4 on GLUE
Pre-Training Data	BooksCorpus + English Wikipedia = 16 GB	BERT + CCNews + OpenWebText + Stories = 160 GB	BooksCorpus + English Wikipedia = 16 GB	BooksCorpus + English Wikipedia = 16 GB
Method	Bidirectional Trans-former, MLM & NSP	BERT without NSP, Using Dynamic Masking	BERT Distillation	BERT with reduced para-meters & SOP (not NSP)

https://360digitmg.com/blog/bert-variants-and-their-differences

Model	Launch Date	Training Data	No. of Parameters	Max. Sequence Length
GPT-1	June 2018	Common Crawl, BookCorpus	117 million	1024
GPT-2	February 2019	Common Crawl, BookCorpus, WebText	1.5 billion	2048
GPT-3	June 2020	Common Crawl, BookCorpus, Wikipedia, Books, Articles, and more	175 billion	4096
GPT-4	March 2023	Unknown	Estimated to be in trillions	Unknown

https://www.makeuseof.com/gpt-models-explained-and-compared/

Interpreting LLMs

Methods

Behavioral methods

- Measuring the performance of LLMs on linguistically relevant data

Methods

Behavioral methods

- Measuring the performance of LLMs on linguistically relevant data
- LSTMs and Transformers learn some long-distance dependencies, but commonly rely on linear order rather than hierarchical structure (Linzen et al. 2016, Yedetore et al. 2023)

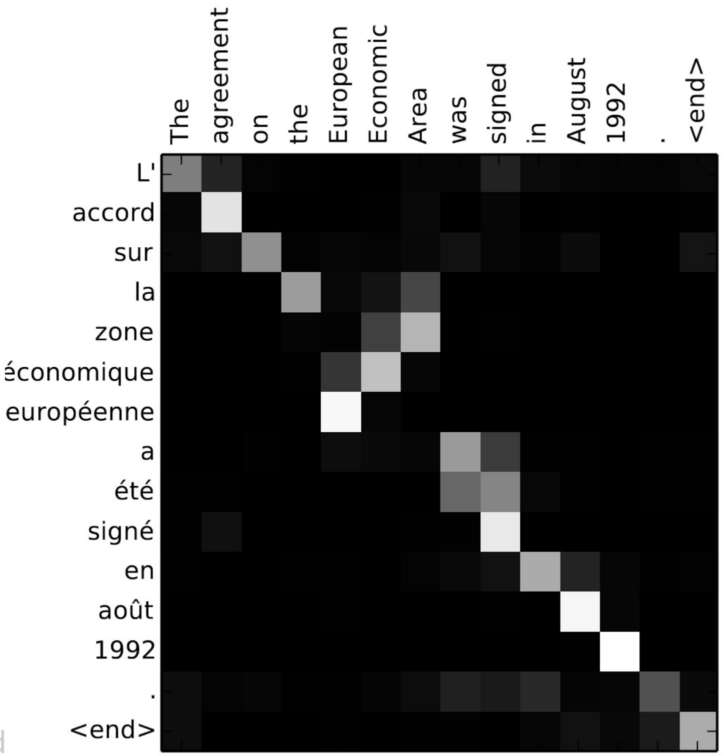
a. The boy who **has** talked **can** read.
b. **Can** the boy who **has** talked ___ read?
c. ***Has** the boy who ___ talked **can** read?

(Yedetore et al. 2023)

Methods

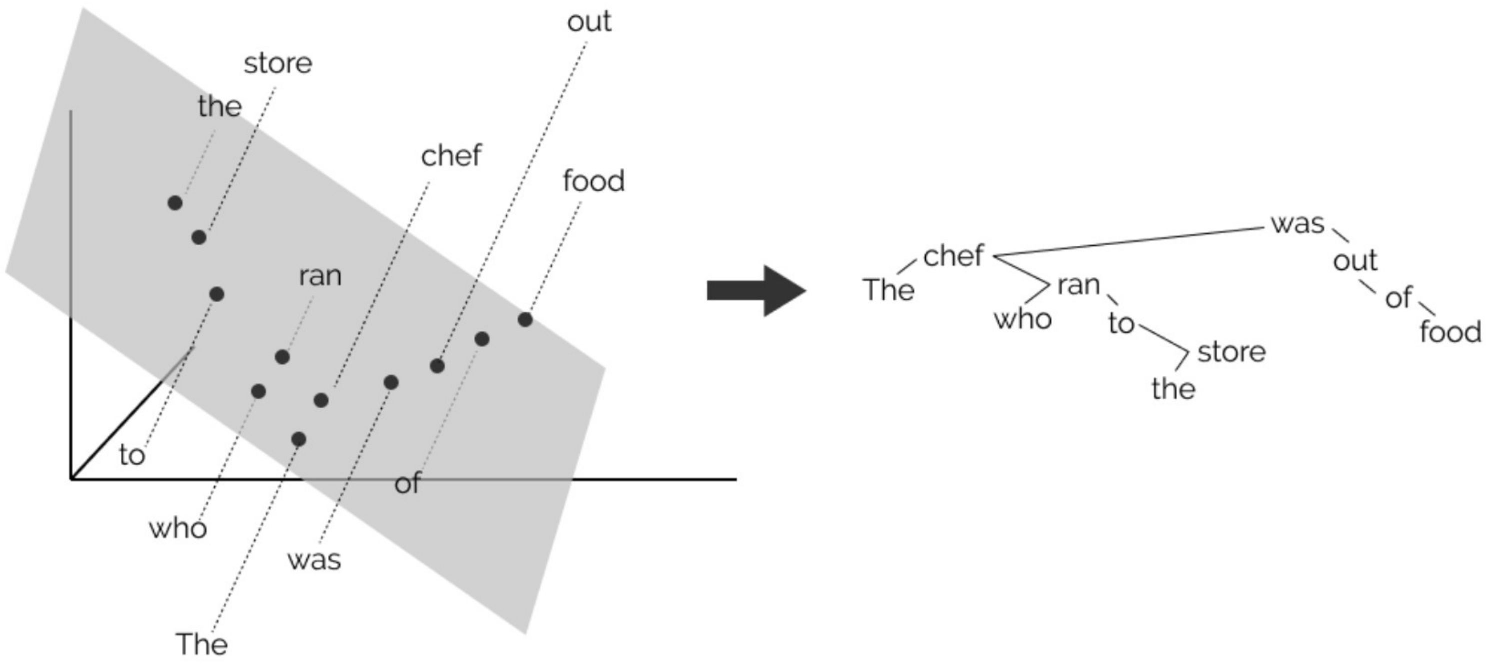
Attention visualization

- Displaying the allocation of attention for each contextual encoding (Bahdanau et al. 2015)
- Challenge: only concerns the input, not the hidden layers



Probing

- Mapping embeddings of pre-trained LLMs to linguistic labels

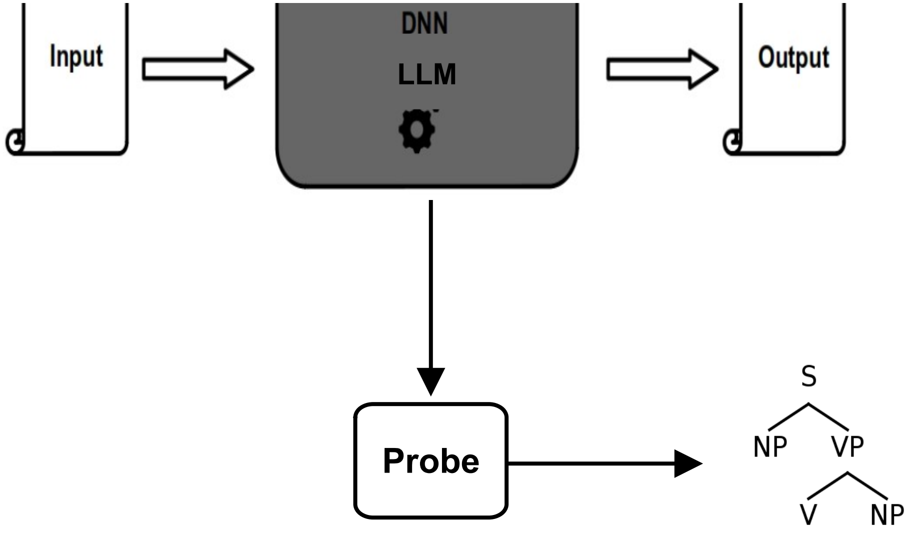


https://nlp.stanford.edu/~johnhew/structural-probe.html

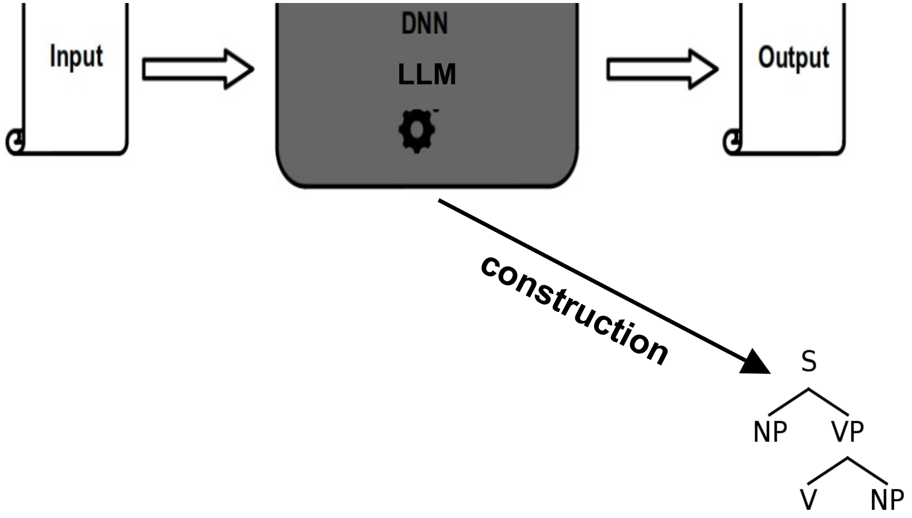
Probing

- Mapping embeddings of pre-trained LLMs to linguistic labels
- Typically *supervised*: labels obtained from human-made (or rule-based) annotations
- *Parameter-free probing*: unsupervised “bottom-up” alternative (Wu et al. 2020)

Supervised:



Parameter-free:



“BERTology” (Rogers et al. 2020)

Grammatical specialization of layers (Tenney et al. 2019, Manning et al. 2020)

- Early layers: superficial information (e.g. part-of-speech, word-order)
- Middle layers: syntactic structure
- Late layers: abstract semantics (e.g. argument structure)

What does BERT learn about the structure of language?

Ganesh Jawahar **Benoît Sagot** **Djamé Seddah**
Inria, France
{firstname.lastname}@inria.fr

BERT Rediscovers the Classical NLP Pipeline

Ian Tenney[1] **Dipanjan Das**[1] **Ellie Pavlick**[1,2]
[1]Google Research [2]Brown University
{iftenney,dipanjand,epavlick}@google.com

Finding Universal Grammatical Relations in Multilingual BERT

Ethan A. Chi, John Hewitt, and **Christopher D. Manning**
Department of Computer Science
Stanford University
{ethanchi,johnhew,manning}@cs.stanford.edu

“BERTology” (Rogers et al. 2020)

Grammatical specialization of layers (Tenney et al. 2019, Manning et al. 2020)

- Early layers: superficial information (e.g. part-of-speech, word-order)
- Middle layers: syntactic structure
- Late layers: abstract semantics (e.g. argument structure)
    - Semantics can also be distributed across layers (Tenney et al. 2019)

What does BERT learn about the structure of language?

Ganesh Jawahar **Benoît Sagot** **Djamé Seddah**
Inria, France
{firstname.lastname}@inria.fr

BERT Rediscovers the Classical NLP Pipeline

Ian Tenney[1] **Dipanjan Das**[1] **Ellie Pavlick**[1,2]
[1]Google Research [2]Brown University
{iftenney,dipanjand,epavlick}@google.com

Finding Universal Grammatical Relations in Multilingual BERT

Ethan A. Chi, John Hewitt, and **Christopher D. Manning**
Department of Computer Science
Stanford University
{ethanchi,johnhew,manning}@cs.stanford.edu

“BERTology” (Rogers et al. 2020)

Syntactic structure

- Probably not directly in attention heads (Htut et al. 2019)
- But encodings can be used to construct syntax (Hewitt & Manning 2019, Wu et al. 2020)
- BERT is sensitive to grammatical relations such as agreement (Goldberg 2019)
- But changing word-order or removing arguments doesn’t always have an effect (Ettinger 2019)

Semantic information

- Thematic roles partly reconstructable via probing (Tenney et al. 2019)
- Challenges with e.g. names and numbers (Wallace et al. 2019, Balasubramanian et al. 2020)

“World knowledge”

- LLMs succeed at certain pragmatic resoning tasks (Petroni et al. 2019)
- Difficulties with tasks that require multi-step reasoning (Forbes et al. 2019)

Challenges (Kulmizev & Nivre 2022)

Grammar vs. “coding properties”

- Syntactic relations (e.g. “subject”) can be coded by word-order, agreement, etc.

Assumptions about grammatical formalism

- Choice of formalism impacts probing results (Kulmizev et al. 2020)

Separating variables

- Data, model architecture, task, linguistic phenomenon

Specifying research questions

- What *does* the model learn?
- What *could* the model learn?
- What *must* the model learn?

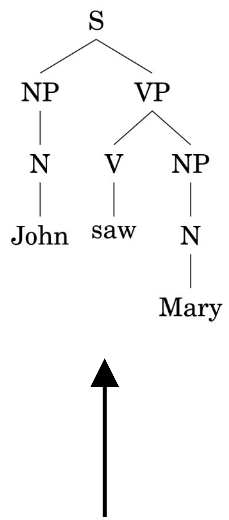
Challenges (Kulmizev & Nivre 2022)

“(…) hypotheses, methodologies, and conclusions comprise many conflicting insights, giving rise to a paradoxical picture reminiscent of Schrödinger's cat – where syntax appears to be simultaneously dead and alive inside the black box models.”

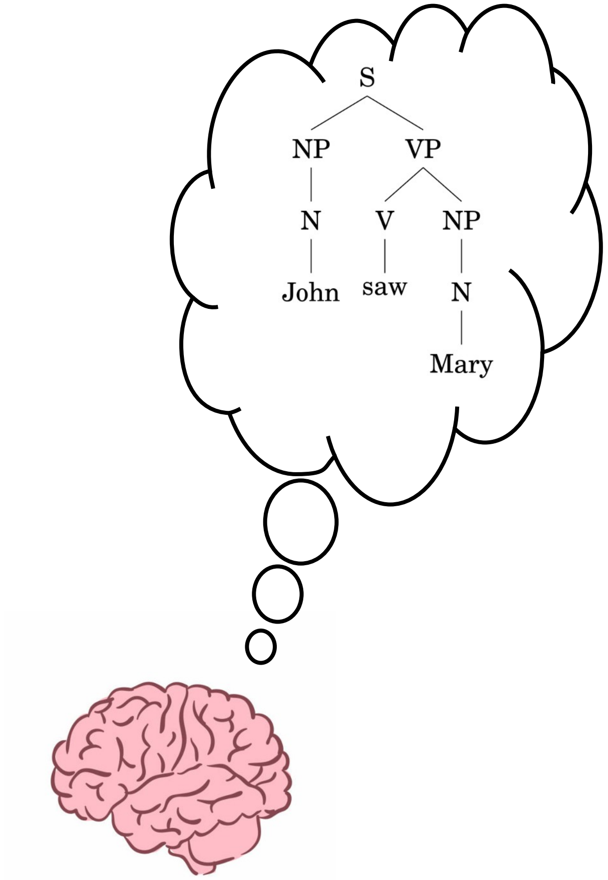
Challenges (my work)

Ambiguity of "linguistic representation" (Buder-Gröndahl 2023)

- Are linguistic properties in the *data* or in *cognition?*



"John saw Mary"



Challenges (my work)

Ambiguity of "linguistic representation" (Buder-Gröndahl 2023)

- Are linguistic properties in the *data* or in *cognition?*

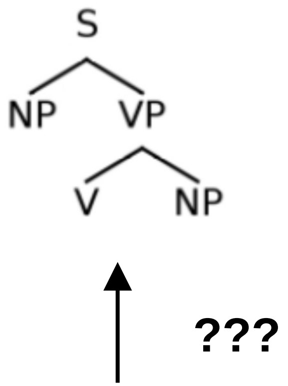
"It remains for linguists to show, in detail, that the speaker has no 'ideas', and that the noise is sufficient."

(Bloomfield, 1936: 93)

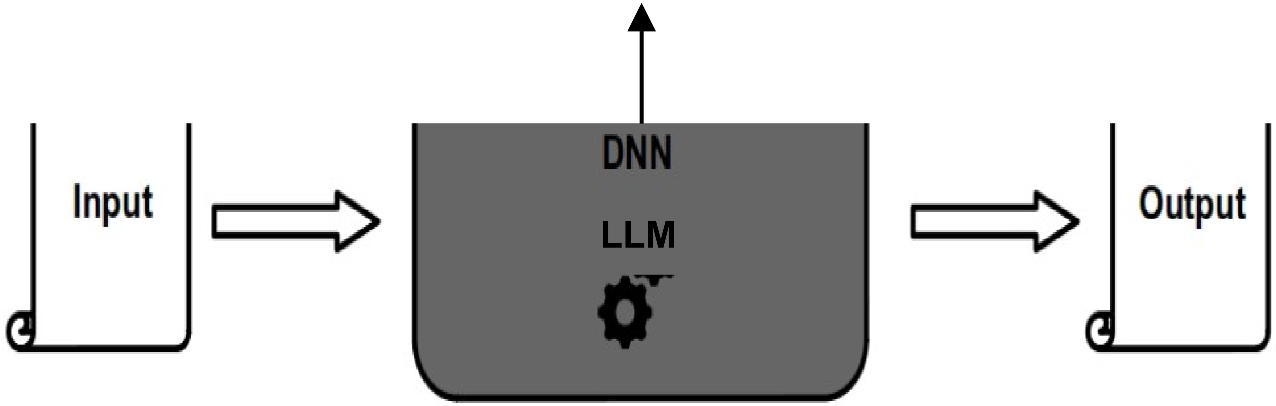
It is appropriate, in my opinion, to regard the grammar of L as a representation of fundamental aspects of the knowledge of L possessed by the speaker-hearer who has mastered L."

(Chomsky, 1975: 5)

Challenges



$$\begin{bmatrix} 0.13452 & 0.34562 & 0.65387 & 0.07501 & 0.10053 \\ 0.06753 & 0.28746 & 0.98463 & 0.44763 & 0.00562 \\ 0.34672 & 0.20056 & 0.94461 & 0.37004 & 0.02204 \end{bmatrix}$$



References

Adger, D. (2022). What are linguistic representations? Mind & Language, 37 (2), 248–260.

Belinkov, Y., & Glass, J. (2019). Analysis methods in neural language processing: A survey. Transactions of the Association for Computational Linguistics, 7 , 49–72.

Chomsky, N. (2012). The science of language. Cambridge: Cambridge University Press.

Coenen, A., Reif, E., Yuan, A., Kim, B., Pearce, A., Viégas, F., Wattenberg, M. (2019). Visualizing and measuring the geometry of BERT. Proceedings of the 33rd Conference on Neural Information Processing Systems (p. 8592-8600).

Collins, J. (2023). Internalist priorities in a philosophy of words. Synthese, 201 (3), 110.

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P. (2020). A survey of the state of explainable AI for natural language processing. Proceedings of the 1st conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing (pp. 447–459).

Egan, F. (2010). Computation models: a modest role for content. Studies in History and Philosophy of Science, 41 (3), 253–259.

Hewitt, J., & Manning, C.D. (2019). A structural probe for finding syntax in word representations. 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) (pp. 4129–4138).

Jawahar, G., Sagot, B., Seddah, D. (2019). What does BERT learn about the structure of language? Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3651–3657).

Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A. (2019). Revealing the dark secrets of BERT. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (pp. 4365–4374).

References

Kulmizev, A., & Nivre, J. (2022). Schrödinger's tree—on syntax and neural language models. Frontiers in Artificial Intelligence, 5.

Kulmizev, A., Ravishankar, V., Abdou, M., Nivre, J. (2020). Do neural language models show preferences for syntactic formalisms? Proceedings of the 58th annual meeting of the association for computational linguistics (pp. 4077–4091).

Pinker, S., & Price, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. Cognition, 28 (1–2), 73–193.

Putnam, H. (1988). Representation and reality. Cambridge: MIT Press.

Rumelhart, D.E., & McClelland, J.L. (1986). On learning the past tenses of English verbs. J.L. McClelland, D.E. Rumelhart, & T.P.R. Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. psychological and biological models (pp. 216–271). Cambridge: MIT Press.

Searle, J. (1992). The Rediscovery of the Mind. Cambridge: MIT Press.

Sprevak, M. (2018). Triviality arguments about computational implementation. M. Sprevak & M. Colombo (Eds.), Routledge handbook of the computational mind (pp. 175–191). London: Routledge.

Swoyer, C. (1991). Structural representation and surrogative reasoning. Synthese, 87 (3), 449–508,

Tenney, I., Das, D., Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 4593–4601).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., . . . Polosukhins, I. (2017). Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing (pp. 6000–6010).

Wu, Z., Chen, Y, Kao, B. and Liu, Q. (2020). Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4166–4176.