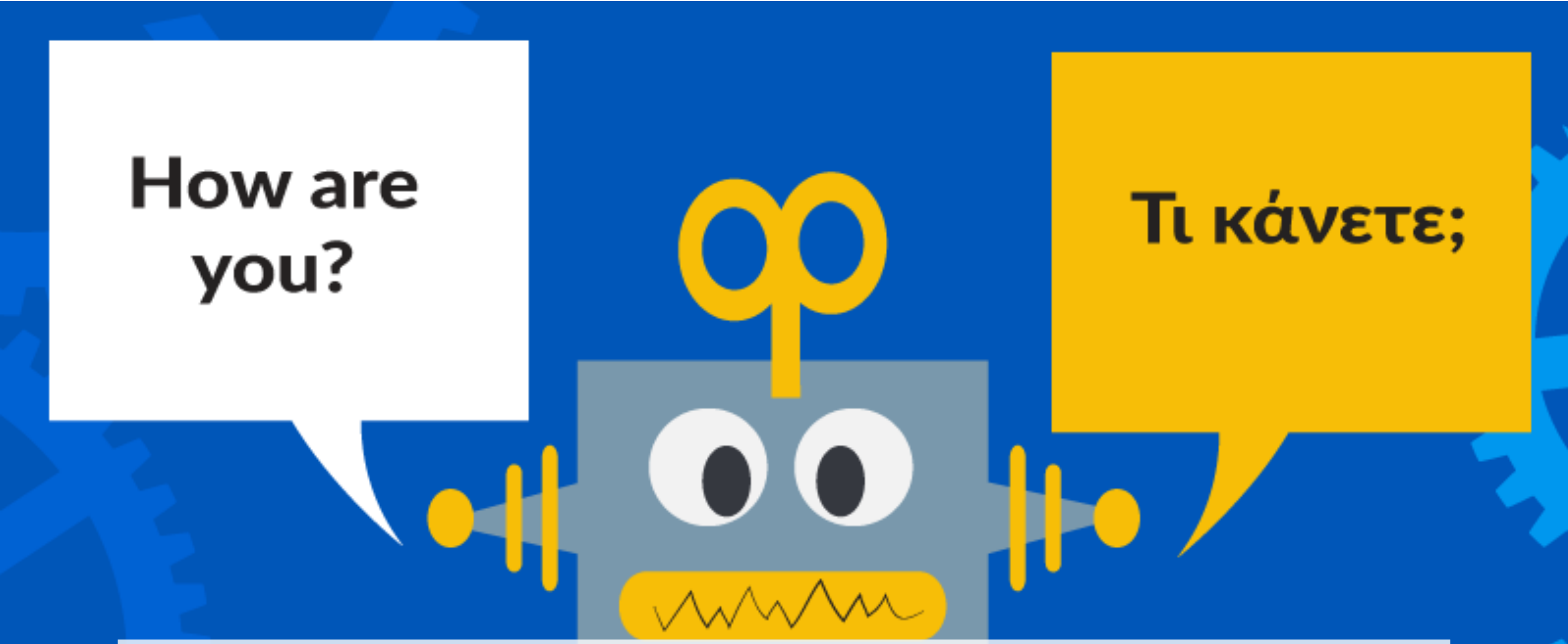


Neural Machine Translation

Instructor: Jackie CK Cheung and David Adelani

COMP-550

J&M Ch. 13 (3rd ed.). Eisenstein, Ch. 18

Outline

Finish: IBM models and phrase-based MT

Neural machine translation

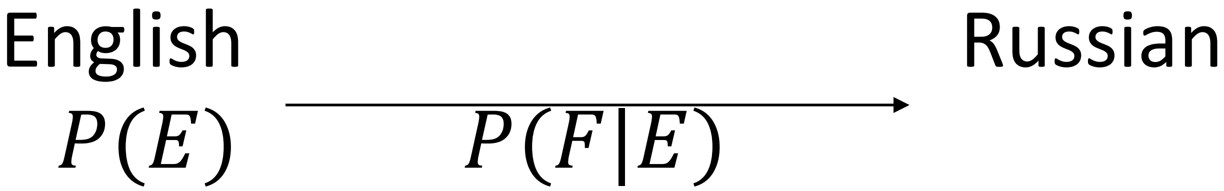
- Attention mechanism

Transformers and large language models

- Prompting

Statistical Machine Translation

Let’s look at a popular direct-transfer approach to statistical machine translation: the **noisy channel model**.



English

$P(E)$

$P(F|E)$

Russian

> *When I look at an article in Russian, I say:*
> *‘This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.’*
>
> Warren Weaver, 1955

IBM Model 1

IBM developed a series of five influential models that make increasingly powerful assumptions.

Model 1 is the most basic:

- Each source word is aligned to **zero or one** target word
- Don't try to model different **distortions** of word order (e.g., completely flipping word order vs. just swapping the orders of one or two words)
- Don't try to model likelihood of **fertility** (some phrases, e.g., *take a walk*, might be translated as one unit)

Expectation-Maximization

1. Initialize the parameters $P(f|e)$ randomly
2. Iterate for a while:
   - **E-step**: Given the current parameters, compute the expected value of $\text{Count}(f,e)$ over the training data
   - **M-step**: Given the current $\text{Count}(f,e)$, compute the new MLE $\theta_k = P(f|e)$

Details, Details

In practice, don’t initialize $t(f|e)$ uniformly:

- Given reasonable sizes of lexicon, too many parameters = too much memory and computation!
- Rather, restrict it to word pairs e’, f’, where e’ and f’ occur is some aligned sentence pair in the training set.

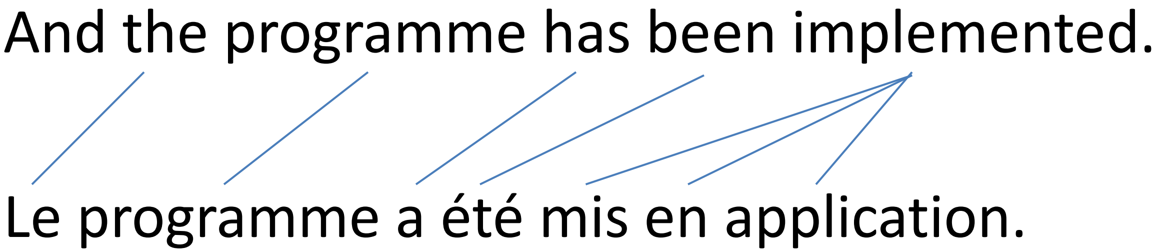
When sentence lengths are high, need to efficiently compute probabilities of all possible alignments.

- Can adapt our algorithm to implicitly sum over all alignments

IBM Model 2

Does not assume that all possible alignment structures are equiprobable.

- For many language pairs, alignment should proceed without much crossing:



And the programme has been implemented.

Le programme a été mis en application.

Can also draw alignment as a table.

Extensions

Higher IBM models

Model 3: model **fertility**—how many words are used to translate a word

Models 4 and 5 changes the dependency structure of the alignments so that they depend on each other

Phrase-Based SMT

What about dealing with phrases that are better translated as a unit?

coup	*blow*
foudre	*lightning*
coup de foudre	*love at first sight*

Non-constituents also benefit:

Spass am	*fun with the*

Phrase-based, rather than word-based SMT can solve this problem by adding a little more context.

Need to learn **phrase table**

A Model of Phrase-based MT

1. Split sentence into phrases

$$E \;=\; e_1 e_2 \ldots e_I = ep_1 ep_2 \ldots ep_N$$

2. Translate each phrase with **phrase translation probability** $P(fp|ep)$
3. Rearrange phrases with some **reordering probability** $d(dist)$
   - e.g., penalty for changing position

$$P(F|E) = \prod_n P(fp_n|ep_n) d(dist_n)$$

MT Decoding

We still need a **decoding algorithm** to *search* for the best possible translation predicted by a given model.

Many search algorithms can be used:

A* search

Greedy hill-climbing

Beam search

…

Let’s briefly describe a greedy hill-climbing method (Germann et al., 2001)

Greedy Hill-Climbing

Start by creating one complete candidate translation

- e.g., translate each word separately

$$e^* = \mathrm{argmax}_{\mathrm{e}}\, P(f|e)$$

This gives an initial translation:

Diese Woche ist die gruene Hexe zuhause.

↓

This week is the green witch at home.

Hill Climbing

Then, apply change operators:

- Change the translation of a word or phrase
- Combine the translation of two words into a phrase
- Split up the translation of a phrase into two subphrases
- Rearrange parts of the translation

At each point, we evaluate all of the transformations by computing $P(E)P(F,A|E)$, and select the change the maximizes this.

We iteratively run this process until reaching a local optimum.

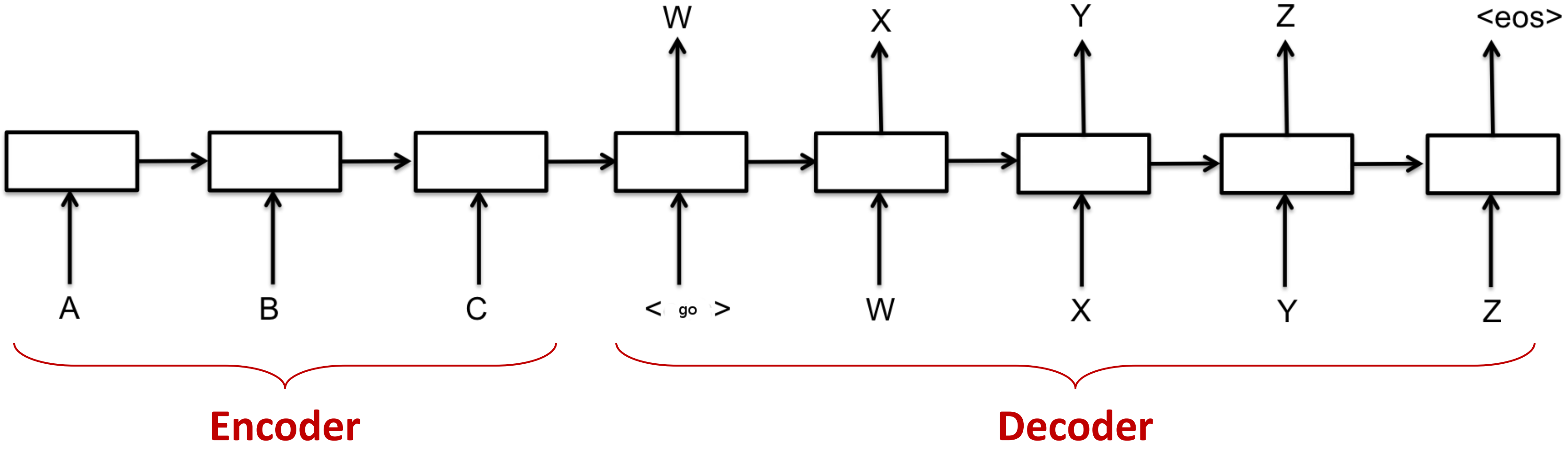
Neural Machine Translation

Neural networks can be applied to machine translation!

Methods developed here **led to current SOTA models** for large-scale language processing.

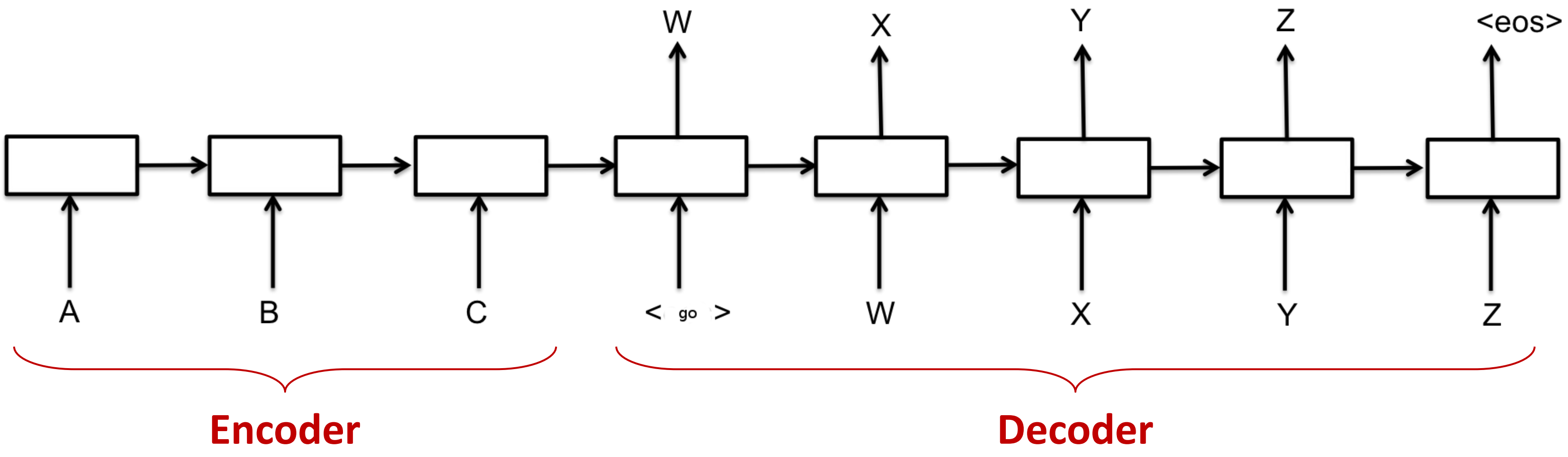
RNNs for MT

Consider a **sequence-to-sequence** recurrent neural network (Cho, 2014):



- Each block above is an RNN cell, such as a **LSTM** block
- Or in practice, a bidirectional deep LSTM block

Encoder-Decoder



$z = ENCODE(w^{(s)})$

$w^{(t)}|w^{(s)} \sim DECODE(z)$, can be interpreted as probability

Some tricks improve performance a lot:

- Reverse order of input sentence
- Train an ensemble of translation models, decode by averaging their output probabilities

Attention Mechanism

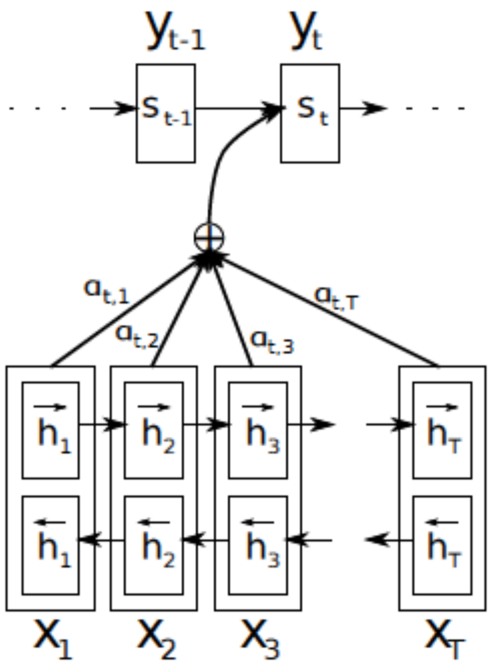
- Translate and align at the same time?
- At decoder step, take a weighted combination of the hidden representations in the encoder for use in predicting next word (Bahdanau et al., 2015):

$$c_i = \sum_j \alpha_{ij} h_j$$ Used in decoding at time i

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k e_{ik}}$$

$$e_{ij} = a(s_{i-1}, h_j)$$

where a is a feed-forward NN



Decoder

Encoder

Attention As Soft Retrieval

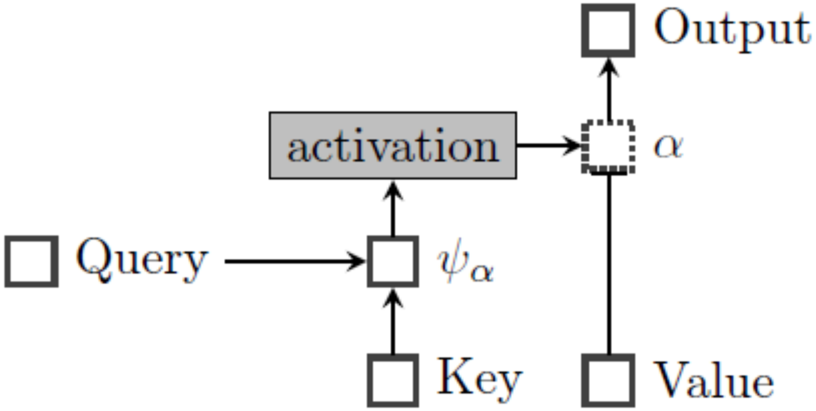


Figure from Eisenstein, Ch. 18

Attention is like a *soft* version of a retrieval system from a memory of (key, value) pairs:

Query – representation of what we're looking for

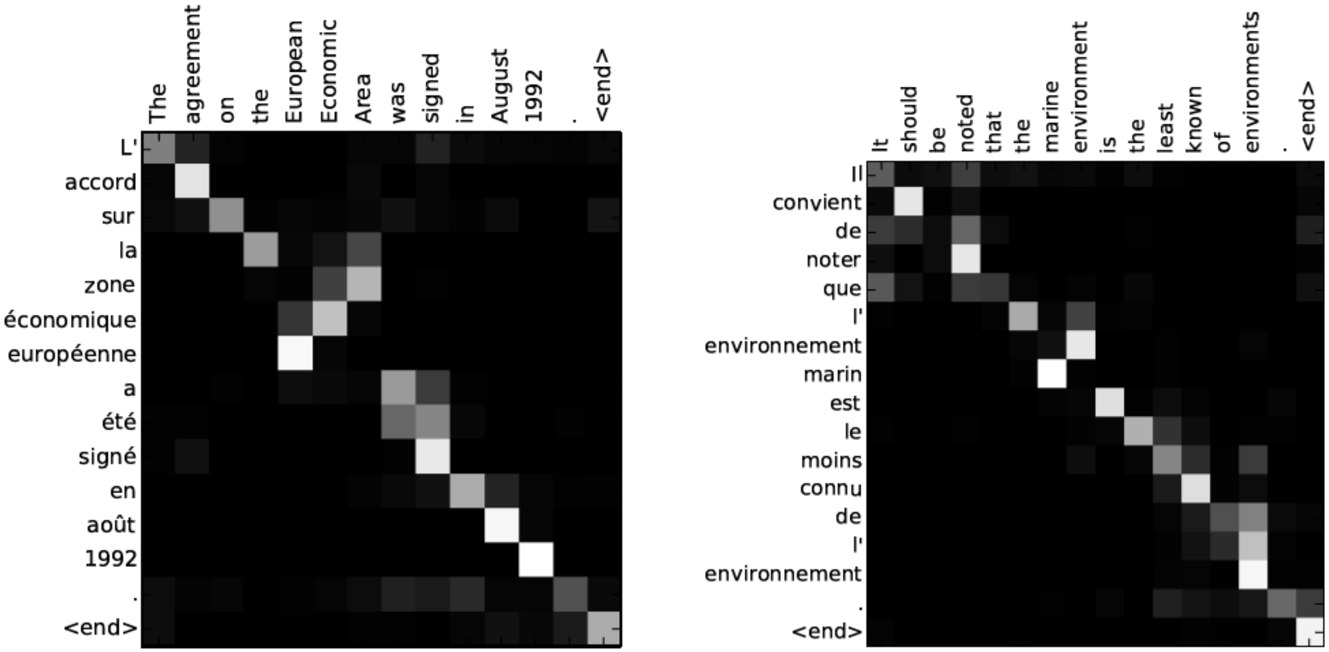
Key – representation of an entry in memory

Value – information that is stored

Soft because all of the above are vectors, and an entry can be partially retrieved, since α ranges between 0 and 1.

Let's identify the query, key, and value in the NMT model.

Visualization of Attention Weights



(a)

(b)

from (Bahdanau et al., 2015)

Use of attention now widespread in NLP!

Transformer Architecture (Vaswani et al., 2017)

What if attention was all we needed?

Motivation to replace RNNs like LSTMs:

- Despite supposedly solving vanishing gradient problem, recurrence in LSTMs still make it difficult to look at patterns and information over long distances.
- Inherent nature of recurrence – need to pass information one step at a time

Idea behind Transformers:

- Allow information flow between any pair of words!

Attention in Transformers

Sentence: $w_1 \;\; w_2 \;\; \dots \;\; w_n$

Embeddings: $x_1 \;\; x_2 \;\; \dots \;\; x_n$

Goal is to compute next layer of word representations at layer l:

$$z_1^l \;\; z_2^l \;\; \dots z_n^l$$

Attention learn a distribution over words to decide how important each word is in order to compute the representations at the next layer

Values, Keys, and Queries

Three views of a word:

query use of this word as a query, because we want to compute its representation at the next layer

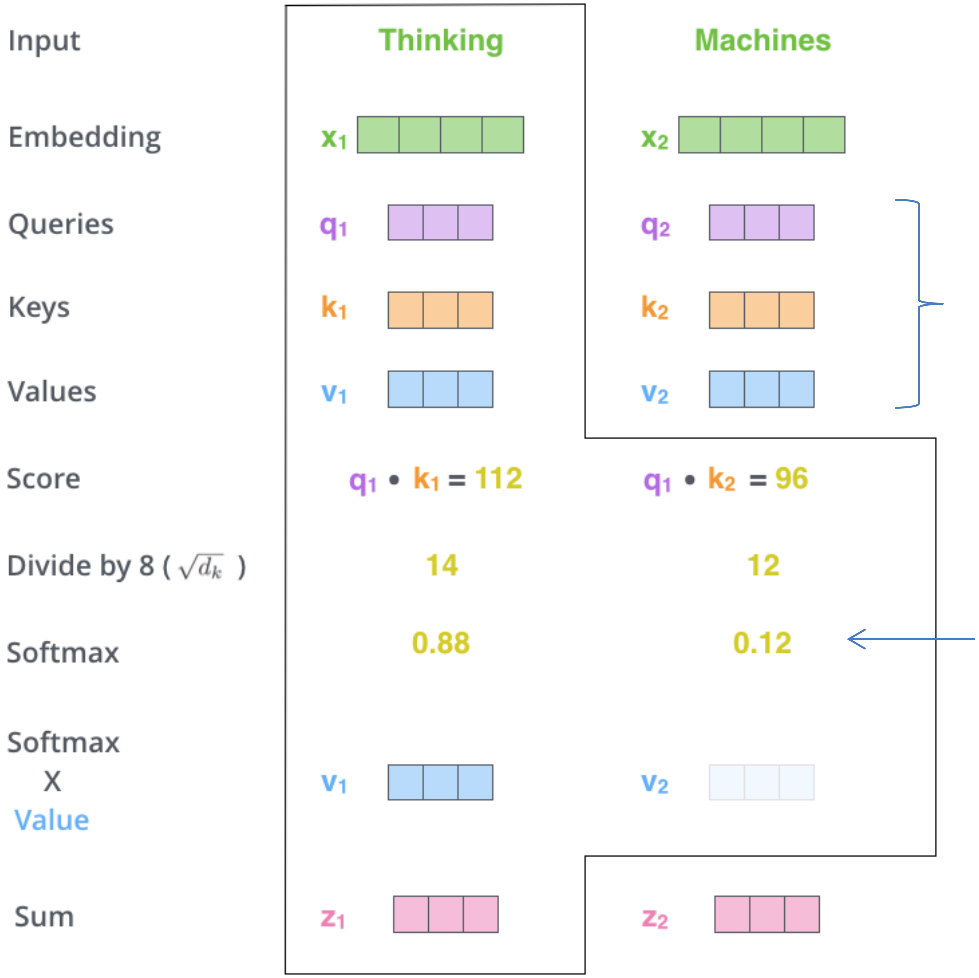
key use of this word as a key; we use this vector to decide how important the word is to another word as part of the attention computation

value this vector stores the value associated with the key, once you've done the attention computation

Each view is associated with its own vector

Example: Two word sentence

Computing the representation of the first word at the next layer:



Done by projecting x_i into subspaces

These are the attention scores

$O(N^2)$ connections!

Source: http://jalammar.github.io/illustrated-transformer/

Transformer Architecture

There are a number of other bells and whistles.

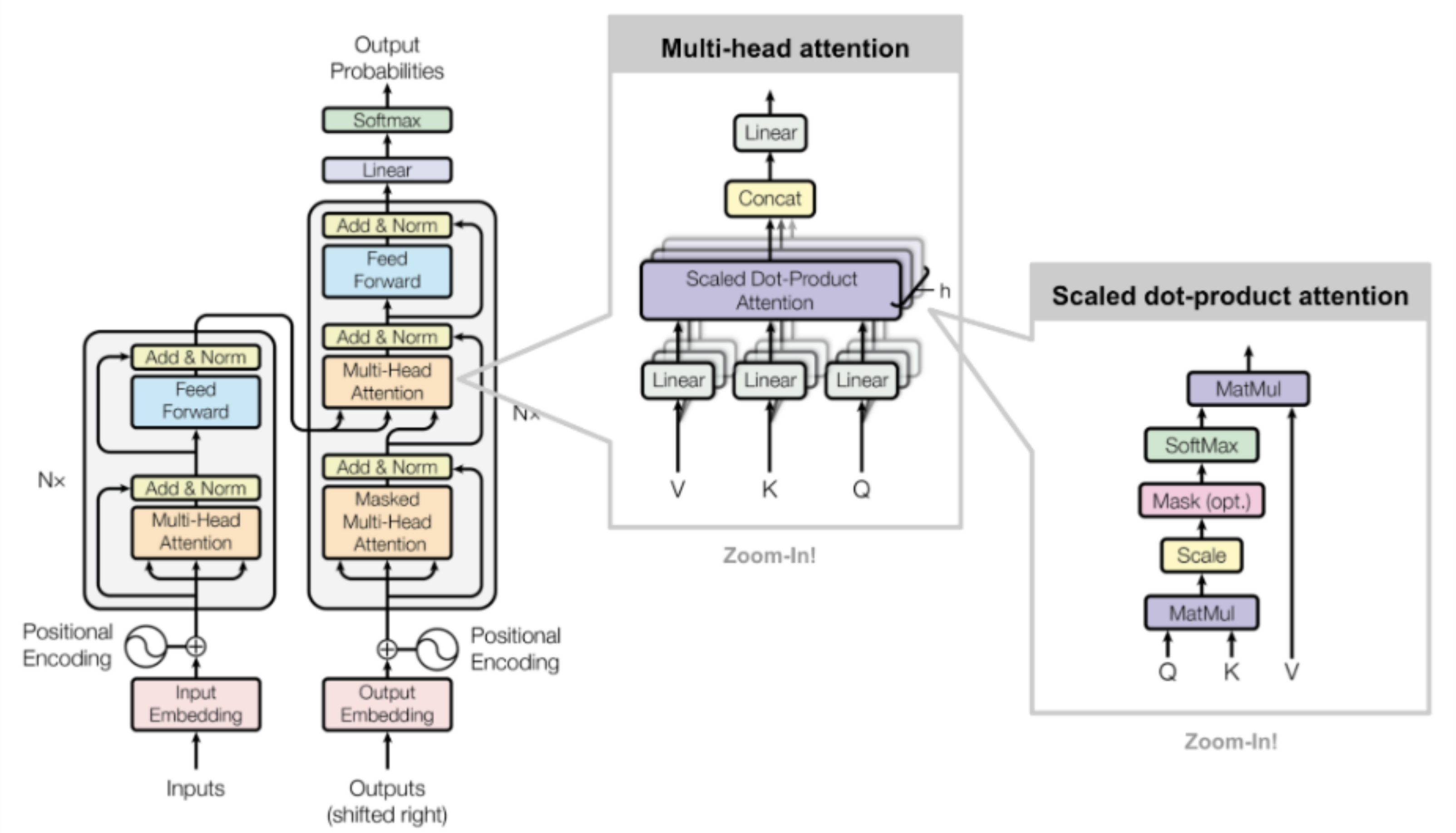


Fig. 17. The full model architecture of the transformer. (Image source: Fig 1 & 2 in Vaswani, et al., 2017.)

Transformers in NMT

Google Translate uses a Transformer encoder, but an RNN-based decoder

https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html

Transformer decoders are very expensive at inference time! Compare:

- **RNN** – feed in one token, predict next token
- **Transformer** – feed in entire context generated so far, predict next token

BERT (Devlin et al., 2018)

A transformer model trained on:

- masked language modelling

e.g., *There is a word [MASKED] in this sentence*.

- next sentence prediction

Given, s1 s2 -> Does s2 follow s1?

Training corpora:

- Books (800M words), English Wikipedia (2500M words)

Up to 340M parameters.

Scaling Up Even More

GPT-3 from OpenAI (Brown et al., 2020):

- Training: ~500B words (web crawled data, books, Wikipedia)
- Up to 175B model parameters
- Many larger models exist now from major tech companies!

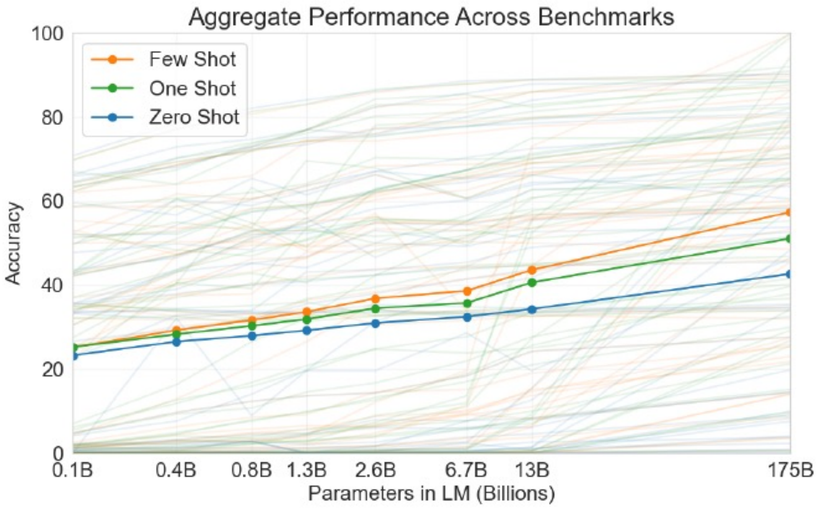


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Successes

BERT + variants are the basis of modern NLP systems.

- Many new SOTA results which start by **fine-tuning** one of these **pre-trained** Transformer models

GPT-3 also shows some success at **few-shot** or **zero-shot** learning:

few-shot give a small number (<100) of examples to finetune on

zero-shot give no new examples; usually need to give some other natural language prompt as side input

T5 (Raffel et al., 2020)

A transformer model trained on:

- Span corruption e.g.

input: Thank you <X> me to your party <Y> week.

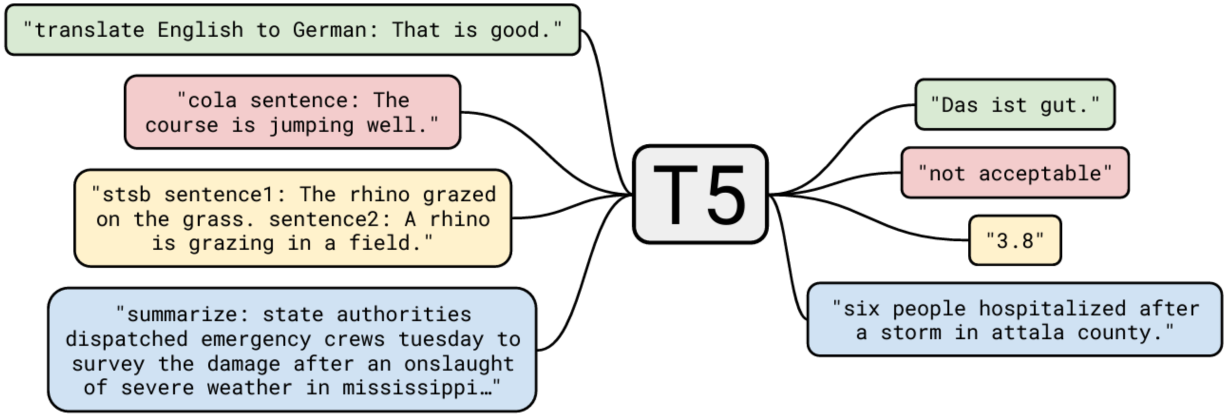
output: <X> for inviting <Y> last <Z>

Training corpora:

- C4 (34B tokens)

Up to 11B parameters.

Cast every task as a text generation task



"translate English to German: That is good."

"cola sentence: The course is jumping well."

"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"

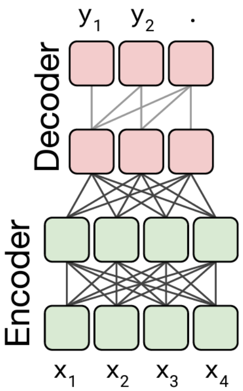
"Das ist gut."

"not acceptable"

"3.8"

"six people hospitalized after a storm in attala county."

Text-to-Text Transfer Transformer



mT5 (Xue et al., 2021)

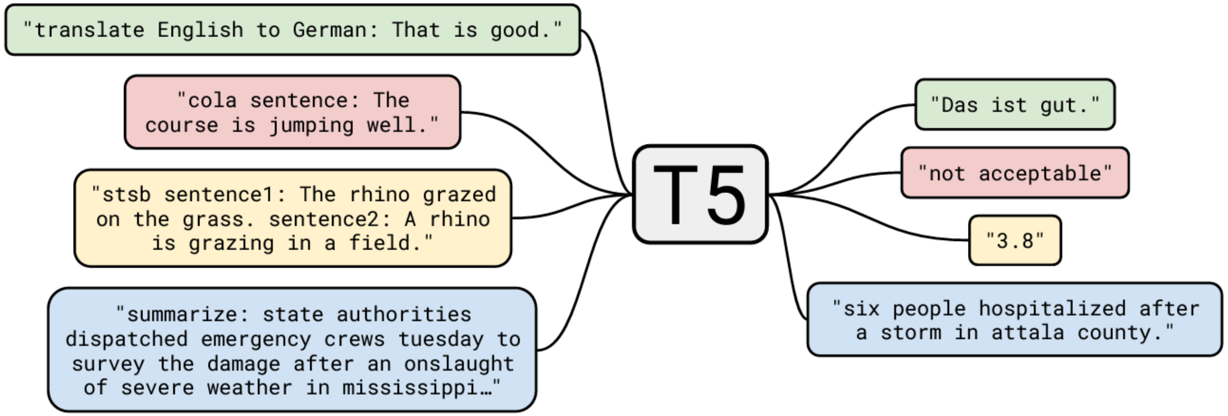
Multilingual T5 trained on the same T5 architecture.

Training corpora:

- mC4 (6.3T tokens): 101 languages

A pre-trained multilingual LM that can be fine-tuned on monolingual and cross-lingual tasks such as:

- Machine translation


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

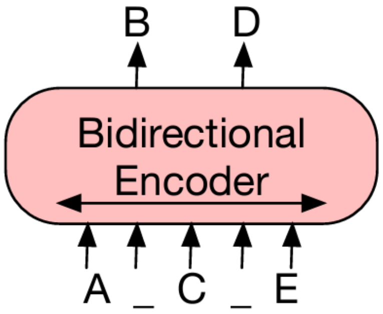
Other Text-to-Text Models

BART (Lewis et al, 2019)

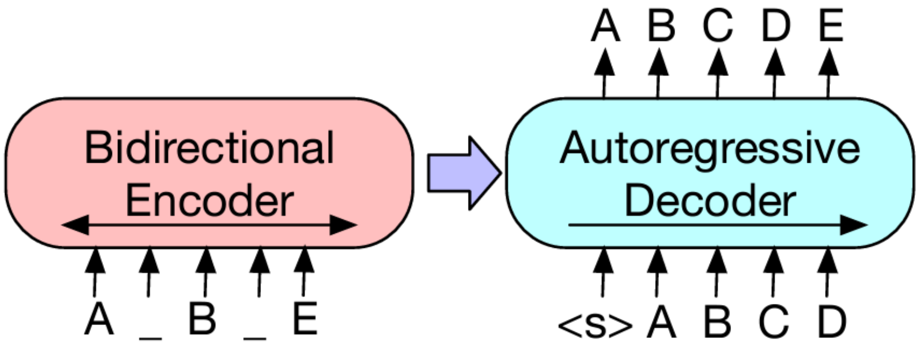
- Similar objective as BERT but with enc-dec transformer architecture.

mBART (Liu et al., 2020) & mBART50 (Tang et al., 2020)

- Multilingual variant with 25 & 50 languages.



BERT



BART

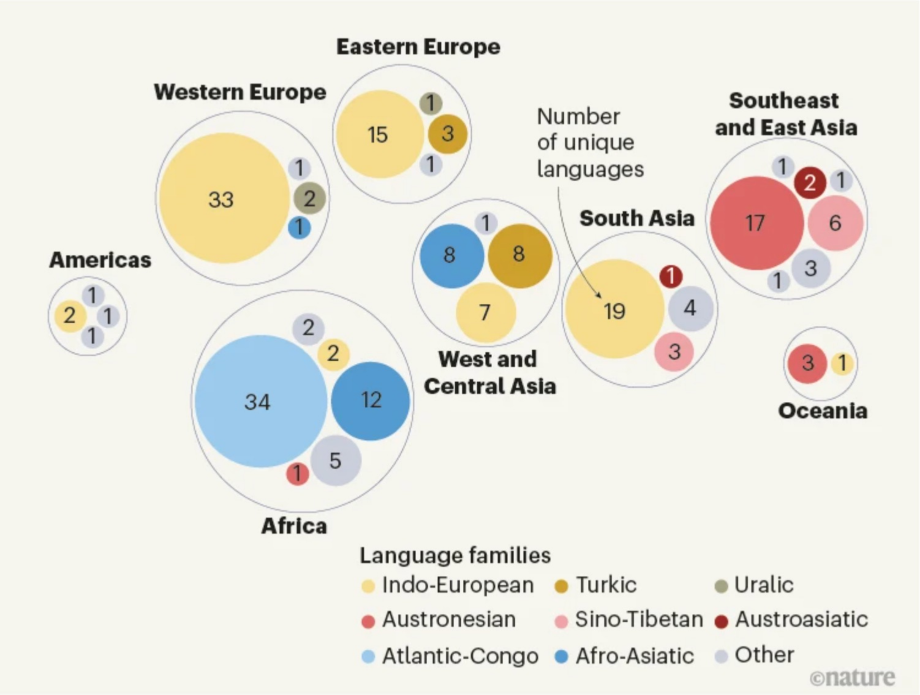
Massively multilingual MT

M2M-100

- Many-to-many MT model for 100 languages

NLLB-200

- Many-to-many MT model for 200+ languages.
- Flores-200 evaluation data



https://www.nature.com/articles/d41586-024-00964-2

Adapting to unseen languages

Difficult to train a massively multilingual MT for all languages

- With a few thousand high quality translation corpus, one can **adapt** M2M-100 & NLLB-200 to new languages.
- Few epochs
- **Example**: Consider a language unseen by any LM such as Luo

	English-Swahili	Swahili-English	English-Luo	Luo-English
M2M-100 (PT)	20.1	25.2	—	—
MT5 (FT)	25.1	29.5	3.1	6.4
mBART50 (FT)	25.8	29.0	10.0	12.1
M2M-100 (FT)	**26.7**	**29.8**	**11.5**	**13.0**

Adelani et al. 2022. A Few Thousand Translations Go a Long Way! Leveraging Pre-trained Models for African News Translation, *In NAACL*.

Current Trends

Given a good enough model, can give **instructions** in the context (a.k.a. **prompt**).

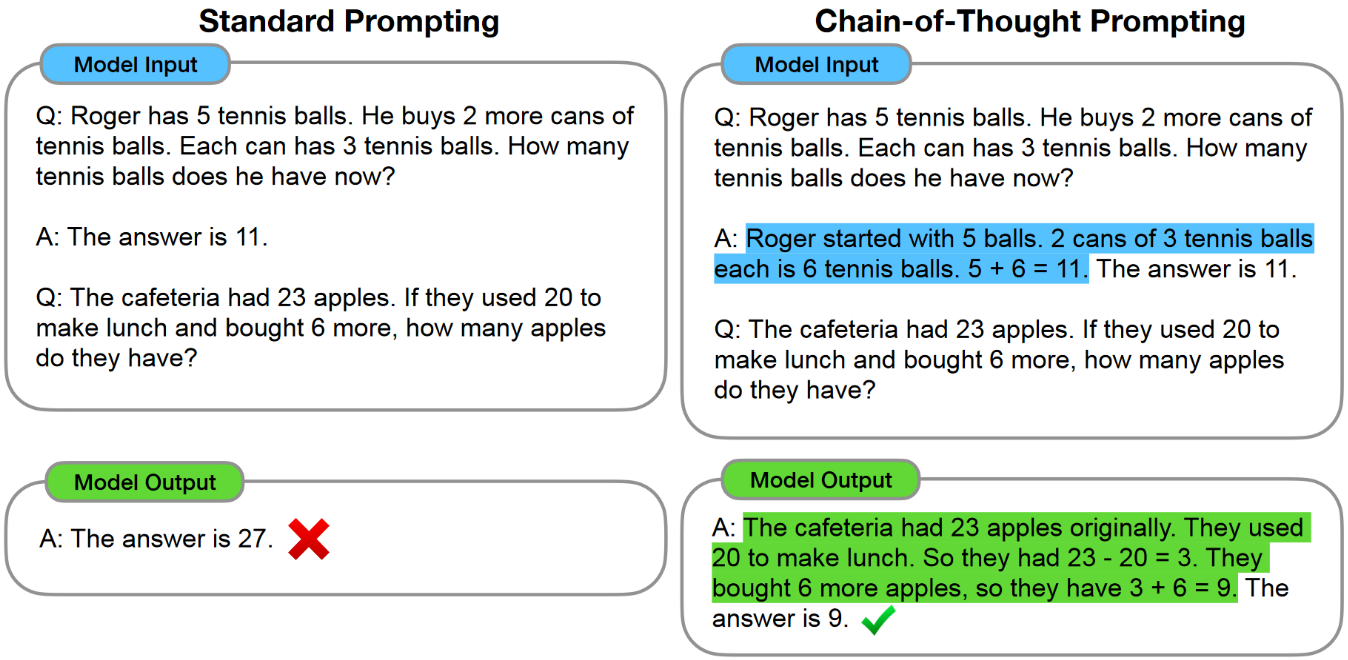
Complete the sentence as if we're in a dark comedy murder mystery. Mary had a little ______

accident	GOOD
lamb	GOOD?
very	BAD
up	BAD

Typical setup: Instructions + demonstrations + query

Chain-of-“Thought” Prompting

Include intermediate reasoning steps:



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Wei et al., 2022

Reference

- Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015.
- Devlin et al. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. ACL 2014.
- Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast Decoding and Optimal Decoding for Machine Translation. ACL 2001.
- Vaswani, Ashish, et al. Attention is all you need. *Advances in neural information processing systems*. 2017.
- Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022.