Machine Translation

And Multilingual NLP

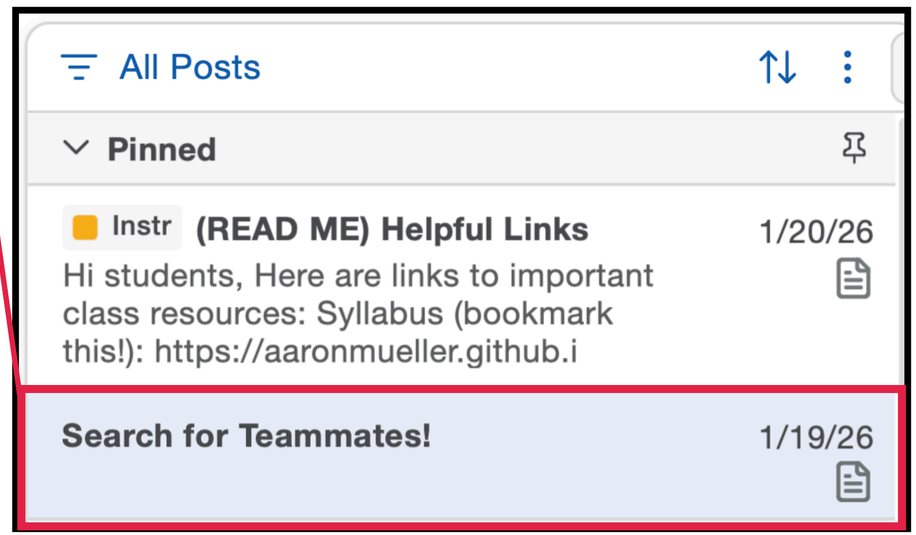
Aaron Mueller
CAS CS 505: Natural Language Processing
Boston University
Spring 2026

Admin

- The **final project specs** have been released!
  - The first deadline is the *final project proposal* (due **Mar. 31**).
  - If you'd like us to help you find group members, post to this Piazza thread before noon on Friday, and we'll randomly assign you to a group of ≈3-4 people.
  - Good news: you'll have BU SCC access!

- **HW2** (Transformers) is due Thursday at 11:59pm.

- HW1 grades will be released this week.



Low-rank Adaptation (LoRA)

LoRA is a way to fine-tune our model to do a new task.

We can use matrices of size $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ to get:

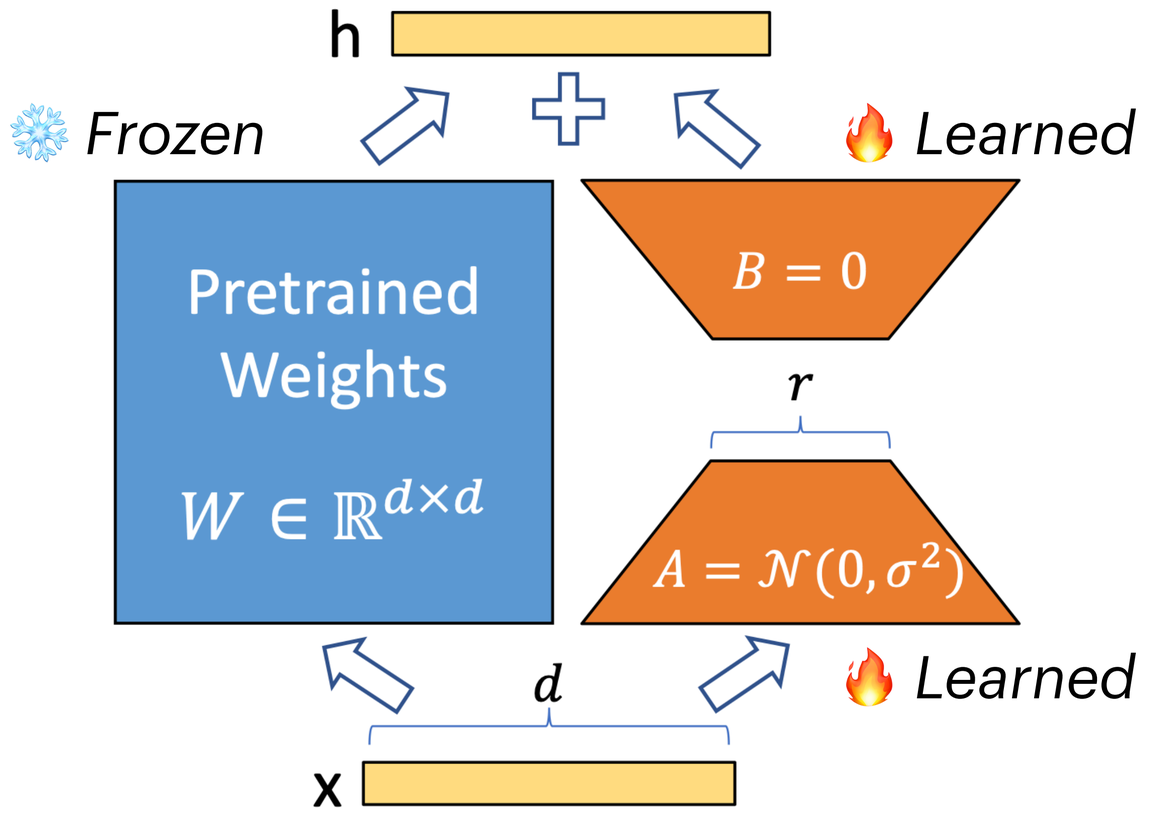
$$\mathbf{h} = \mathbf{x}W + \mathbf{x}AB$$

+ Don't need gradients for most parameters: way lower memory usage

+ Doesn't add any time during inference

+ Modular: can swap out different LoRAs for different tasks/domains

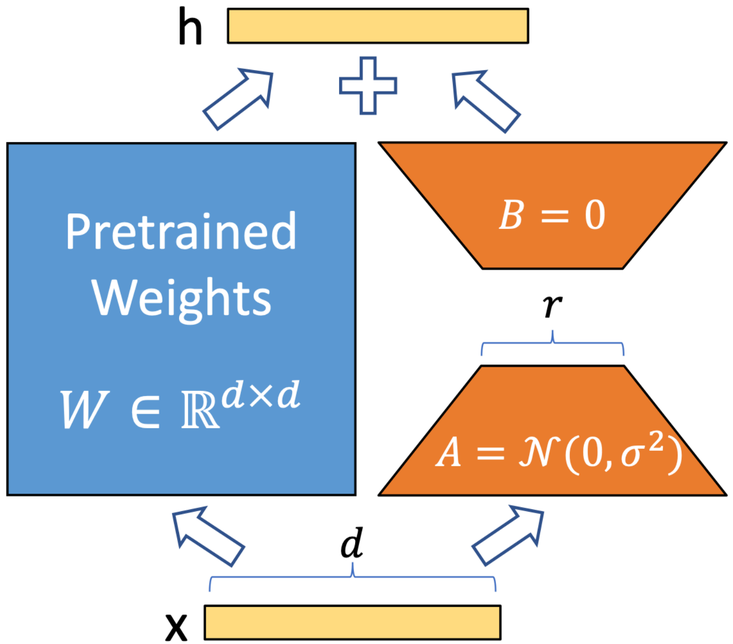
- Quite a few more hyperparameters to fiddle with



[Hu et al., 2021]

Implementing LoRA

```
input_dim = 768
output_dim = 768
rank = 8

W = . . .     # depends on pretrained model shape

W_A = nn.Parameter(torch.empty(input_dim, rank))
W_B = nn.Parameter(torch.empty(rank, output_dim))

nn.init.kaiming_uniform_(W_A, a=math.sqrt(5))
nn.init.zeros_(W_B)

def regular_forward_matmul(x, W):
    h = x @ W
    return h

def lora_forward_matmul(x, W, W_A, W_B):
    h = x @ W
    h += x @ (W_A @ W_B) * alpha
    return h
```



$$\mathbf{h} = \mathbf{x}W + \alpha \cdot \mathbf{x}AB$$

Controls strength of LoRA update

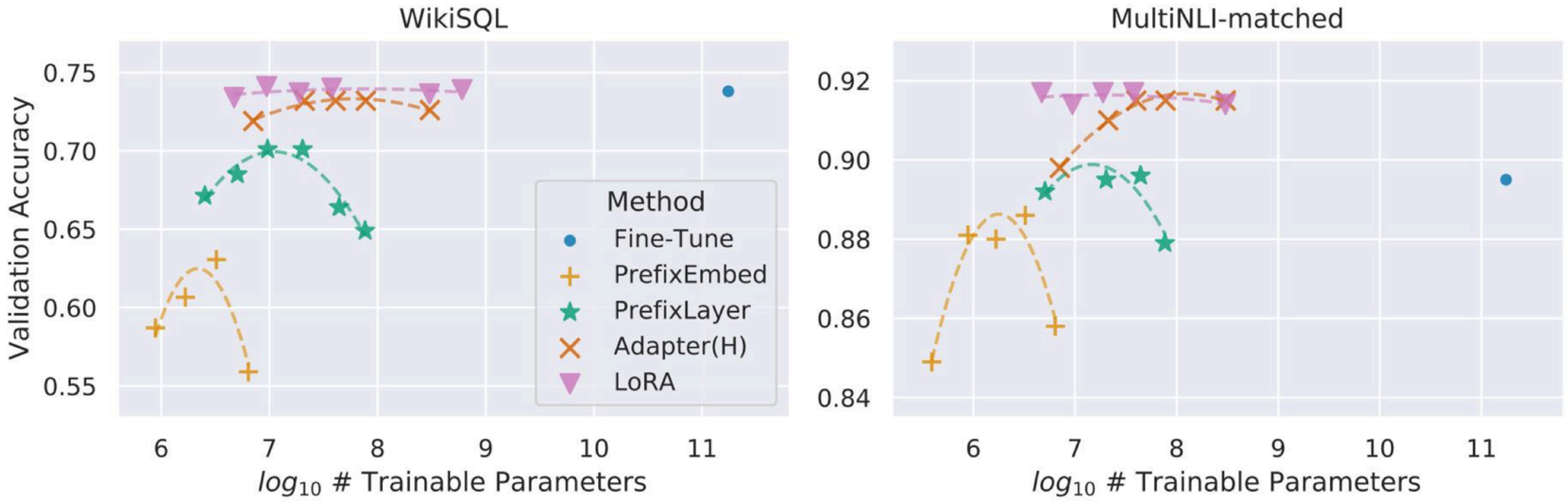
LoRA in Practice

LoRA performs just as well as full fine-tuning on a few NLP tasks.

FT = fine-tune all parameters

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

LoRA gives better performance at the same number of trainable parameters.



Downsides of Fine-tuning Methods Compared to Prompting

1. **Computationally expensive.** Compared to prompting, fine-tuning usually requires a lot more compute and memory.
2. **Time-consuming.** There are a lot of hyperparameters to optimize. Running fine-tuning many times can consume a lot of time and resources.
3. **Risk of catastrophic forgetting.** Once you've updated a model's parameters, the model is probably only good at that task now. It may have forgotten a lot of its general knowledge from pre-training.
4. **The usual issues with optimization.** It's still possible to overfit, underfit, etc.

Takeaways

- Changing the way we ask a language model to do something has *massive* impacts on performance.
    - Chain-of-thought prompting is an easy way to improve performance on formulaic or verifiable tasks—but might not help much in other cases.

- Updating the parameters of a model usually works better—but is also more expensive, and still easy to get wrong.
    - A nice middle ground: parameter-efficient fine-tuning (e.g., LoRA)

- In a few weeks: **post-training** models to follow instructions, and to produce well-structured chain-of-thought-style responses without us asking

Overview of Concepts

Linguistic typology is the study of cross-linguistic similarities and differences.

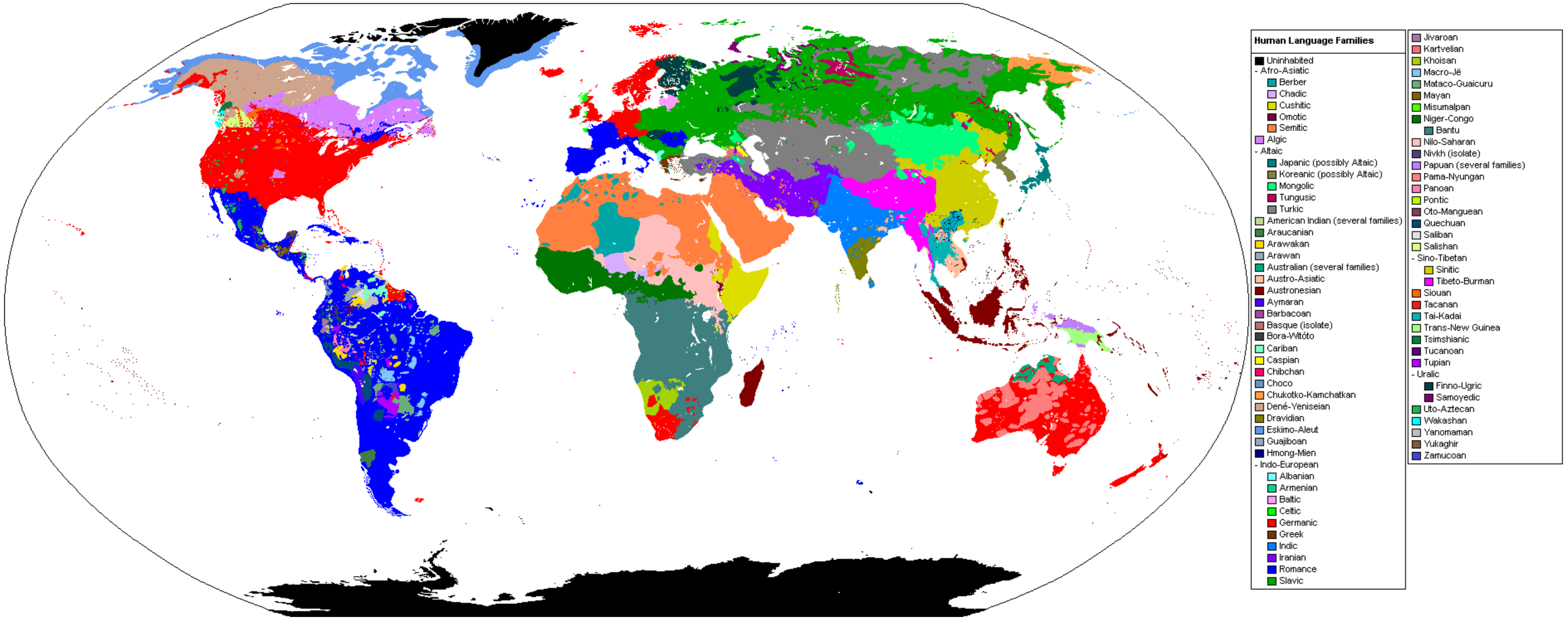
Machine translation (MT) is the task of automatically mapping texts across languages.

Encoder-decoder models are the state of the art for machine translation.

A **parellel corpus** is a corpus where every document appears in multiple languages.

BLEU is a metric for evaluating MT systems.

Beam search is a decoding method that maintains multiple good options until the end.

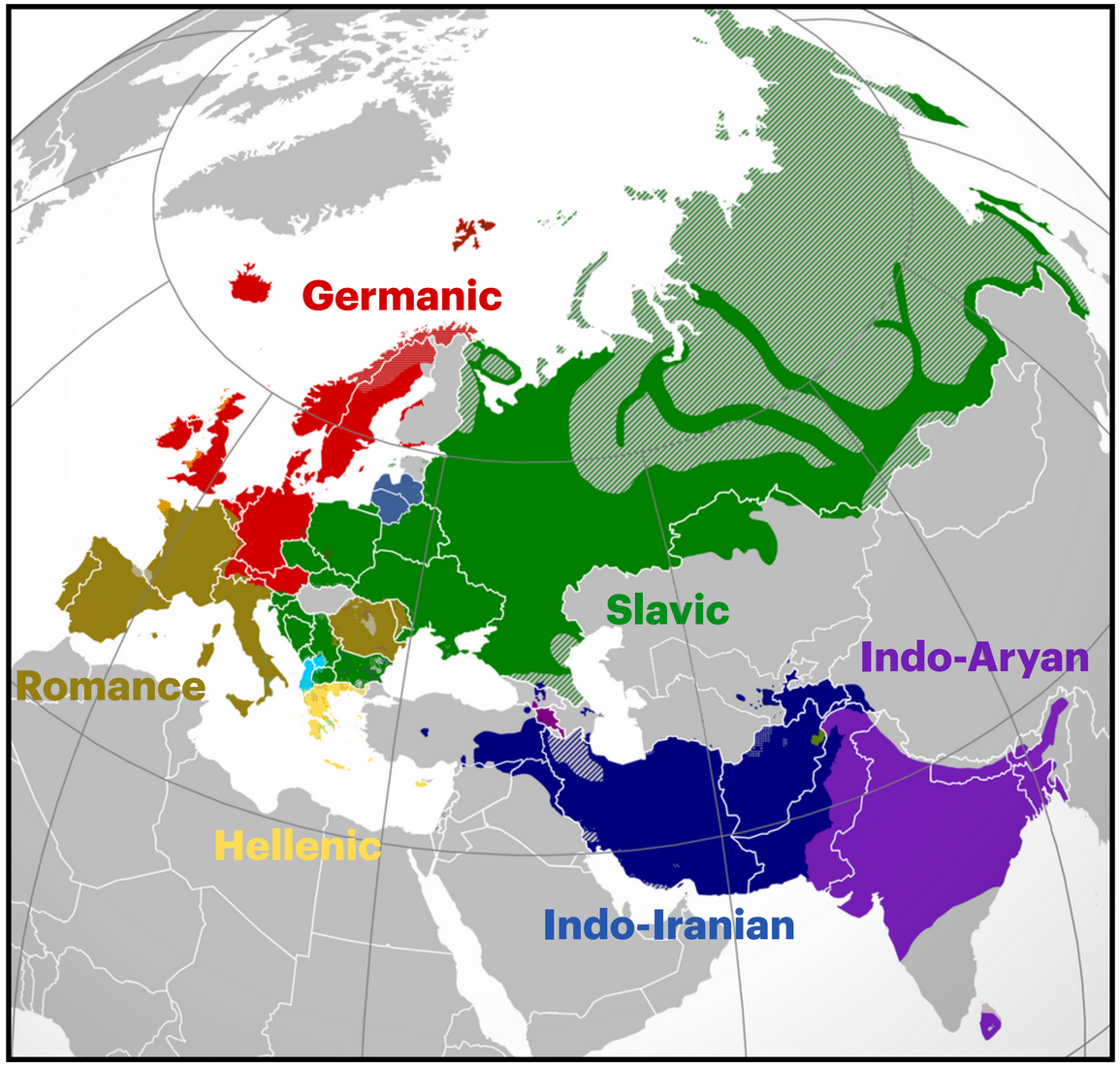


Over 7,000 languages in the world (that we know of)!

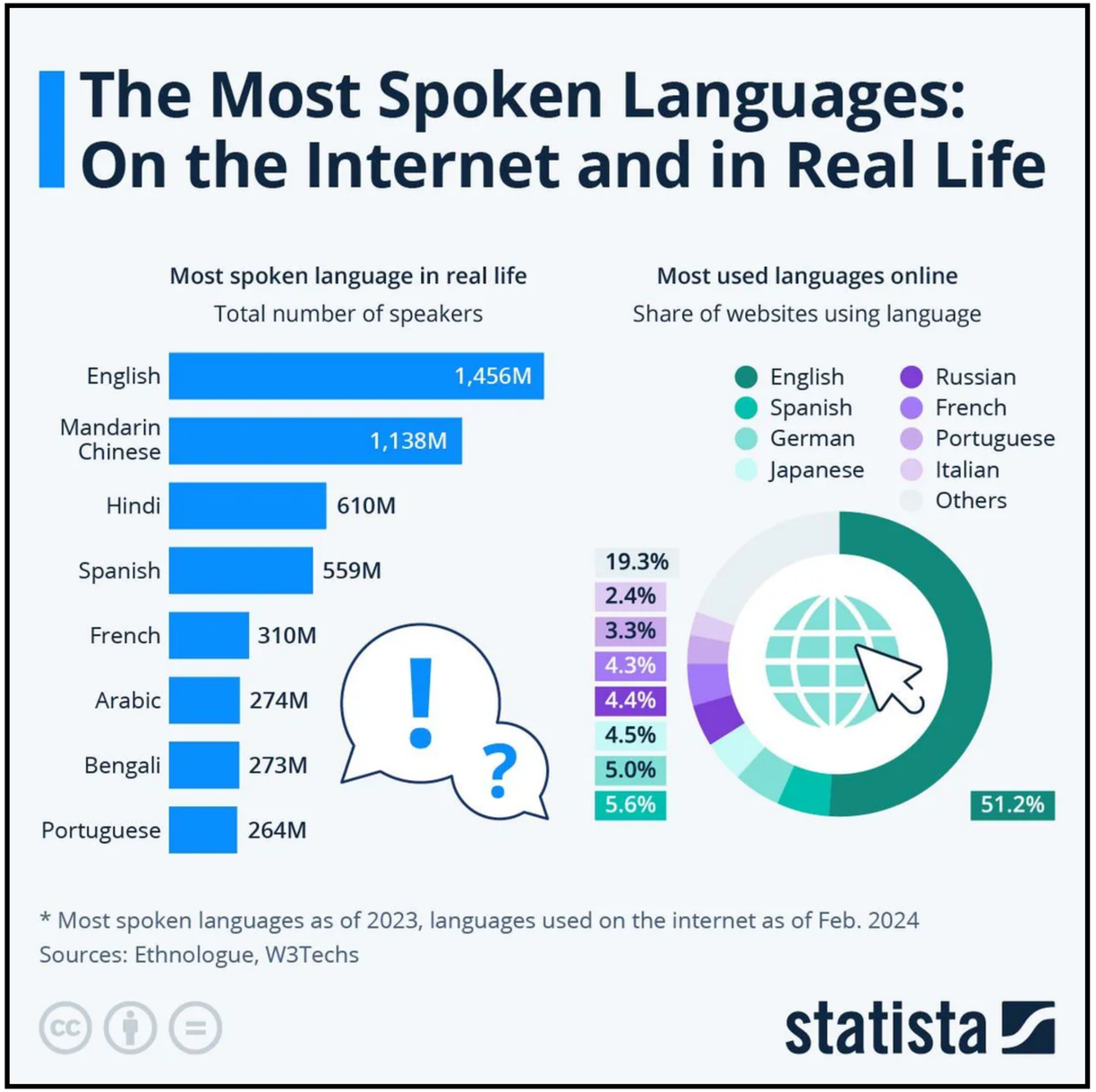
Language Families

- A **language family** is a group of languages descended from a common ancestor (a protolanguage).
- Language families can have subfamilies. English belongs to the Germanic family, alongside Swedish and German.
- Some languages, like Basque, have no known relatives; these are **language isolates**.

The Indo-European language family



- The world has thousands of languages, but the internet is dominated by a few.
- English makes up >50% of internet text!
- But not everyone speaks English—and not everyone *wants* to speak English.



NLP Is Often English-centric

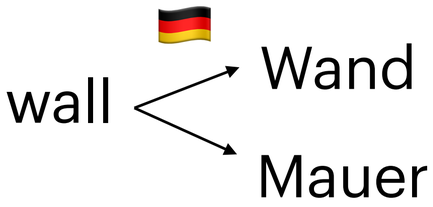
- Almost all of our tasks and examples in this class so far have been in English.

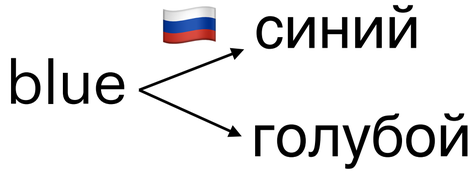
- But there are many other languages out there, many of which are nothing like English!
    - Will the methods we've used so far work well for other languages?

Languages are Diverse

Lexical Diversity

Some words in English have many translations in other languages:



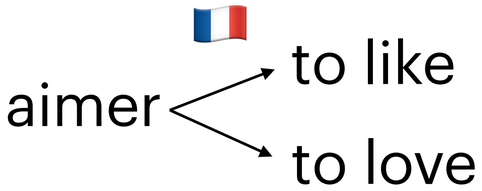


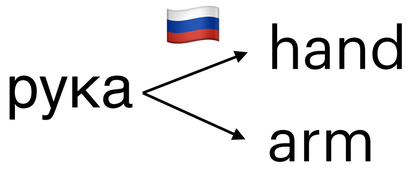
Some ideas that take many words in English take only one in other languages:

Çekoslavakyalılaştıramadıklarımızdan

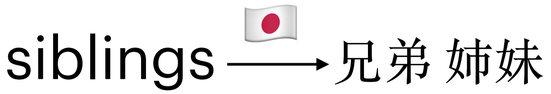
from among those whom we could not make into Czechoslovakians

And vice versa:





And vice versa:



Languages are Diverse

Word Order Diversity

Languages can be grouped by where they put their subjects (S), verbs (V), and objects (O):

English:

S V O

He saw a dog

Japanese:

S O V

彼は 犬を 見た

He dog saw

Arabic:

O S V

كلباً آدم رأى

dog Adam Saw

(Note: Arabic is written right-to-left.)

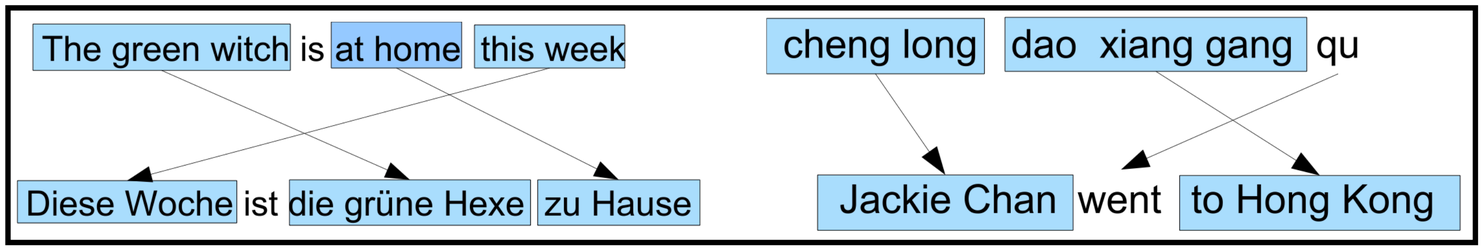
Hixkaryana:

O V S

Kamara y-ahosɨ-ye toto

Dog saw he

Languages also vary in the order of adverbs, prepositions, etc.:



Languages are Diverse

Morphological Diversity

- Some languages are ***isolating***: each word generally has one morpheme

🇻🇳 Anh ấy đọc sách.
He read book

- Some languages are ***agglutinative***: morphemes have easily definable boundaries

🇹🇷 Kitap.lar ilginç.tir
Book.PL interesting.COP

- Some languages (like English) are ***fusional***: morphemes may have many functions and interact with each other

🇬🇧 He cook.s interesting food.s
cook.3PL.SG food.PL

- Languages have widely varying morpheme-to-word ratios.

Linguistic Universals

Some features are **universal** to all languages (that we know of):

- All languages have nouns and verbs
- All languages have ways to ask questions, issue commands, or agree/disagree

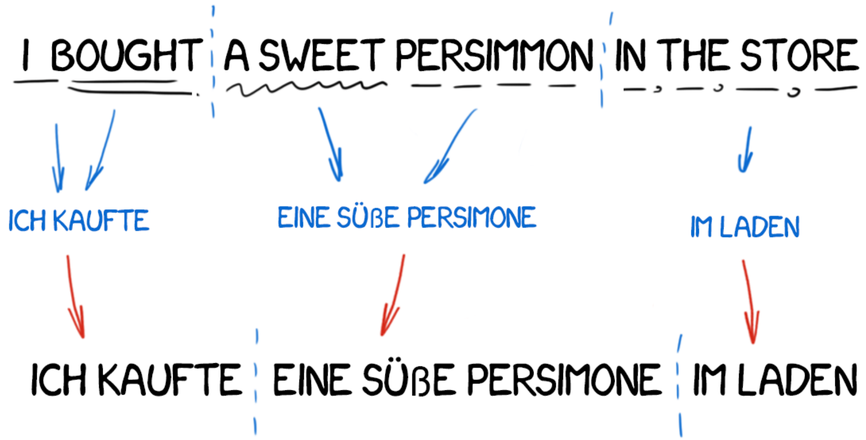
It is believed that universals arise due to human cognitive or biological constraints.

- You may encounter a (controversial) concept known as Universal Grammar: humans may have a biological predisposition for language, and for language to be structured a certain way.

Machine Translation

How do we get computers to translate effectively when languages differ so greatly?

- Can't translate word-by-word
- Could write a bunch of rules, but this would be expensive and slow



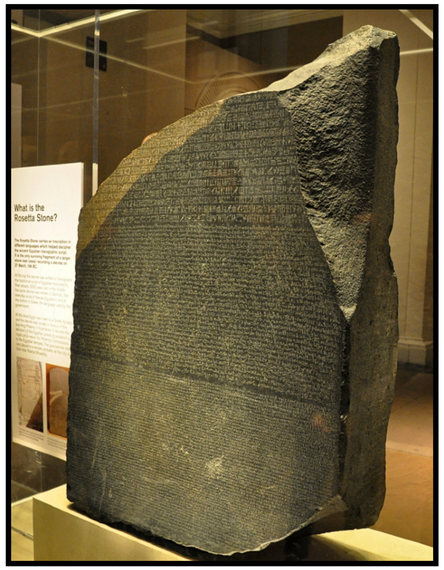
Many huge innovations in NLP came from machine translation: attention, Transformers, encoder-decoder models, among others.

Today, we will walk through a **neural machine translation** (NMT) model using an **encoder-decoder** architecture. We'll also look at a high-quality decoding algorithm, **beam search**, and use **BLEU scores** to evaluate.

Data

Parallel Corpora

Machine translation is a supervised machine learning problem. We make use of a **parallel corpus**, which contains aligned documents or sentences:



The Rosetta Stone, a parallel text (Hieroglyphics, Ancient Greek, Demotic) which helped scholars decipher Egyptian hieroglyphics.

```
{ "de": "Wiederaufnahme der Sitzungsperiode", "en": "Resumption of the session" }

{ "de": "Ich erkläre die am Freitag, dem 15. Dezember 2000, unterbrochene Sitzungsperiode des Europäischen Parlaments für wieder aufgenommen.", "en": "I declare resumed the session of the…

{ "de": "Erklärungen der Präsidentin", "en": "Statements by the President" }

{ "de": "Werte Kolleginnen und Kollegen, wie Sie wissen, hat ein weiteres Erdbeben in Mittelamerika in dieser bereits mehrfach seit Beginn des zwanzigsten Jahrhunderts schwer…

{ "de": "Die vorläufige, schreckliche Bilanz in El Salvador lautet zurzeit bereits: 350 Tote, 1 200 Vermisste, eine vollständig verwüstete Region und Tausende zerstörter Häuser im gesamten…
```

German-English examples from the WMT'19 dataset.

Texts can be aligned at the document level (relatively common), sentence level (common), or word/phrase level (rare).

Data

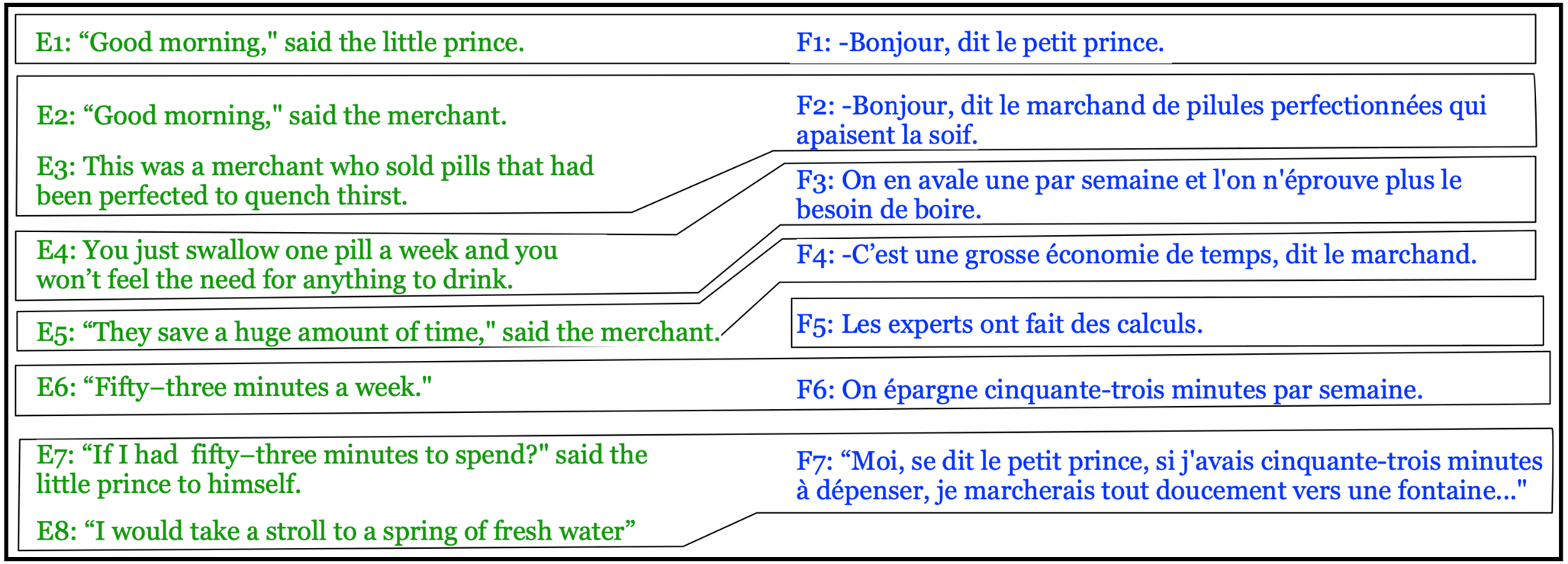
What can you learn from a parallel corpus?

*Je **fais** un **bureau***	*I'm **making** a **desk***	*I'm **constructing** an **office***
*Il **fait** beau*	*The weather **is** nice*	*It**'s** beautiful*
Tu fais quoi?	*What are you doing?*	*What's up?*

Translation is hard because it isn't word-for-word. There are also often multiple translations for one source.

Data

We don’t always have a clean one-to-one mapping of sentences across languages:

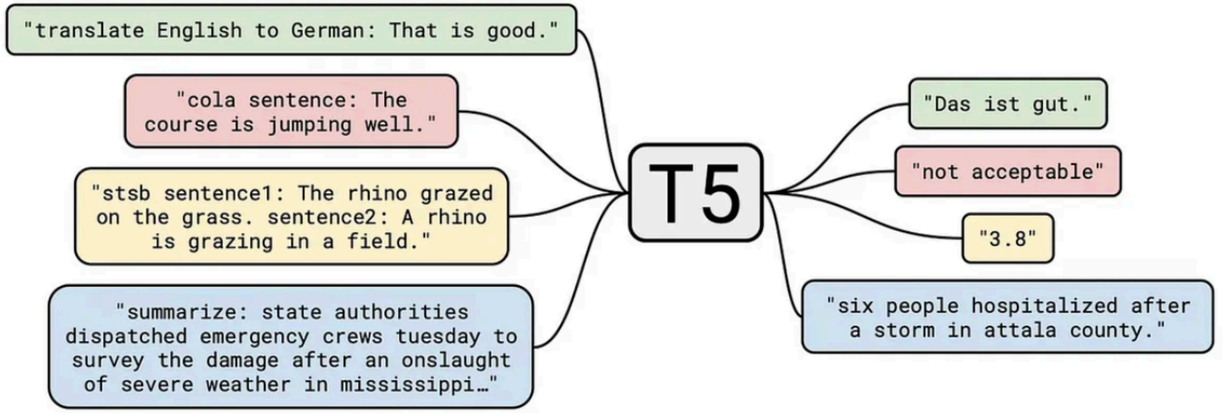


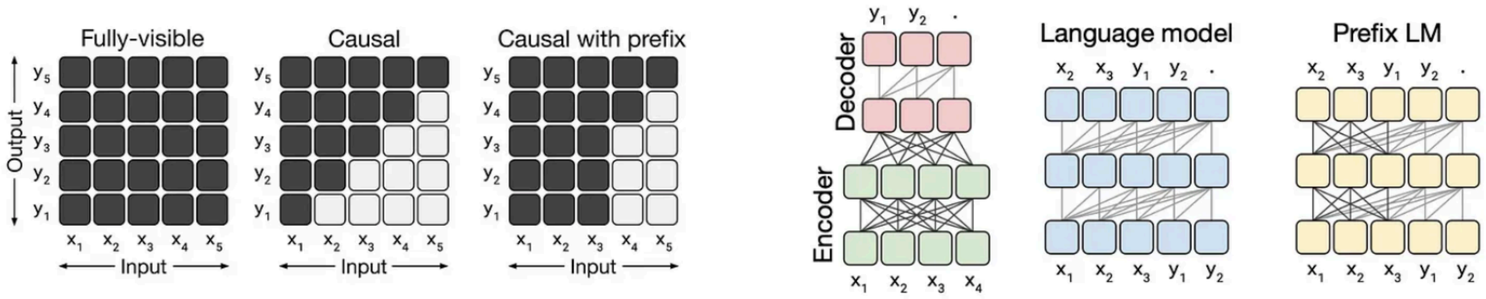
Encoder-decoder Models

Recall the encoder-decoder from a couple lectures back:

We'll use a similar setup today.

Main idea: encode the entire **source** sequence; condition on this and generate the **target** sequence token-by-token.





Encoder-decoder Models

Train a system θ to maximize the probability of the **target** sequence $p(y_1, \ldots, y_m)$ given the aligned **source** sequence $p(x_1, \ldots, x_n)$:

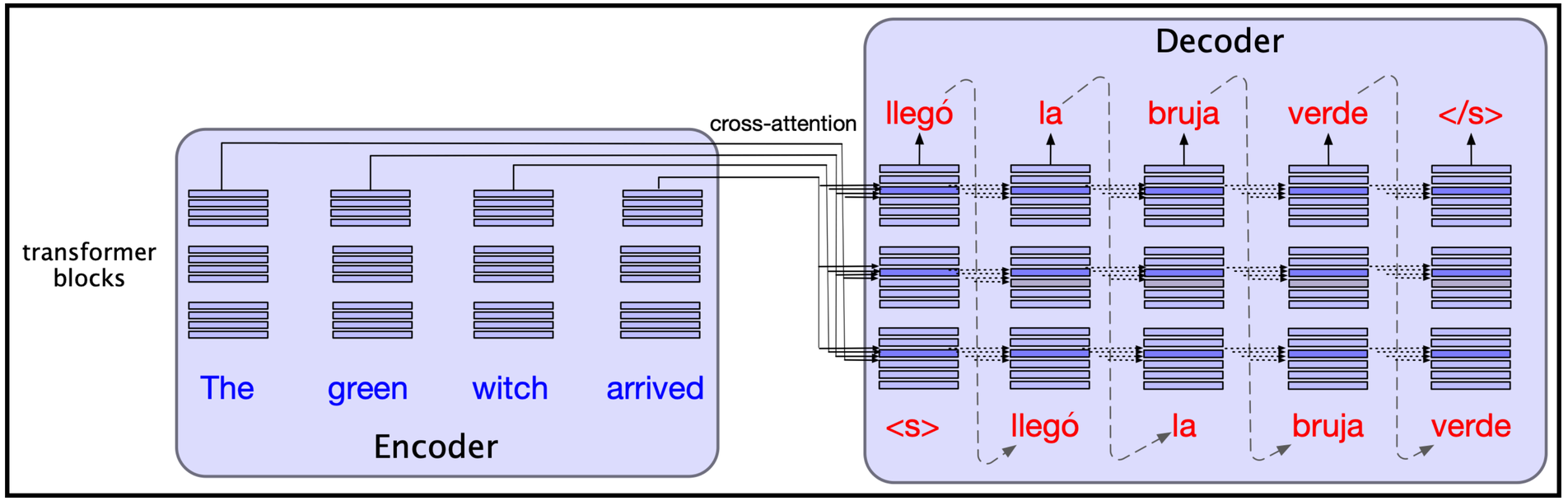
$$\arg\max_{\theta} p_\theta(y_1, \ldots, y_m \,|\, x_1, \ldots, x_n)$$

We will do this by encoding the source sequence into representation $\mathbf{h}$, then decoding token-by-token conditioned on this representation:

$$\mathbf{h} = \text{encoder}(X)$$

$$y_{t+1} = \text{decoder}(\mathbf{h}, y_1, \ldots y_t)$$

Encoder-decoder Models



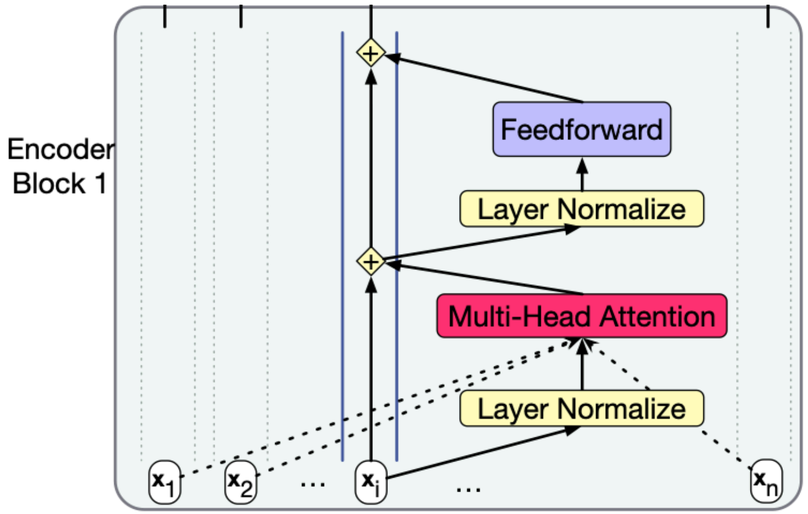
The encoder works much the same way as models like BERT.

The decoder is kind of like the Transformer decoder, but with two additions:

1. The decoder conditions on the source representation *and* generated target tokens.
2. We add a **cross-attention** layer that attends to the source sequence.
    - Like multi-head attention, but keys and values come from the *source sequence*

Encoder-decoder Models

$$\text{CrossAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$



Encoder

$$Q = H^{\text{dec}[\ell-1]}W_Q$$
$$K = H^{\text{enc}}W_K$$
$$V = H^{\text{enc}}W_V$$

Training an Encoder-decoder Model

- Training looks nearly the same as in training an encoder-decoder language model. We again use **cross-entropy loss:**

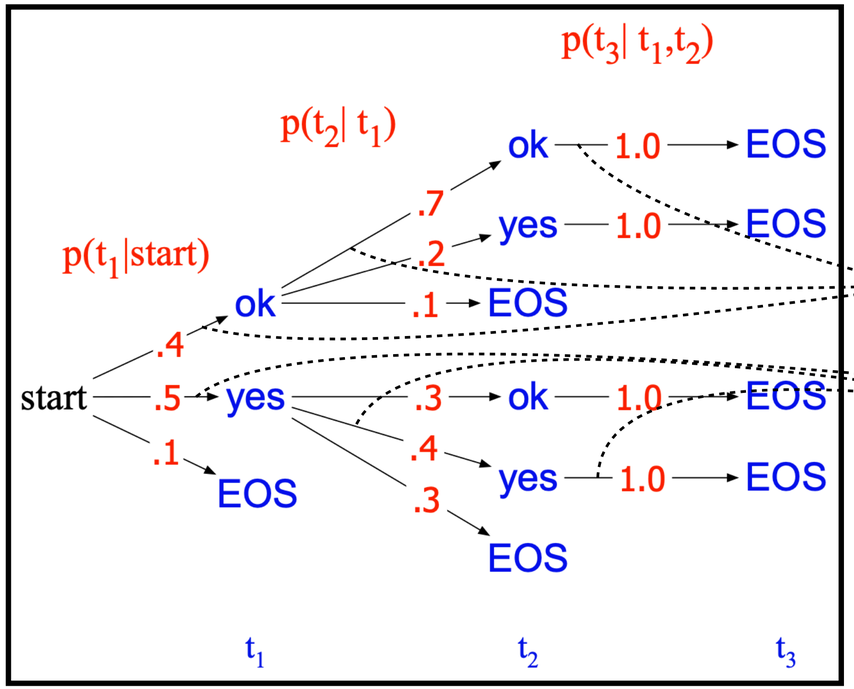
$$L_{\text{CE}}(\hat{y}_t, y_t) = -\log \hat{y}_t[w_{t+1}]$$

negative log-probability of correct next token given the correct prior context

- Note that we also use teacher forcing here, like before.

Decoding in Machine Translation

- In previous lectures, we've discussed greedy decoding, top-k sampling, temperature sampling, and nucleus sampling.
- In machine translation, the most common decoding technique is **beam search**.



If we greedily pick the best token at each step, we get a sequence that is *less* probable than the globally most probable sentence!

Best: ok ok EOS

Greedy: yes yes EOS

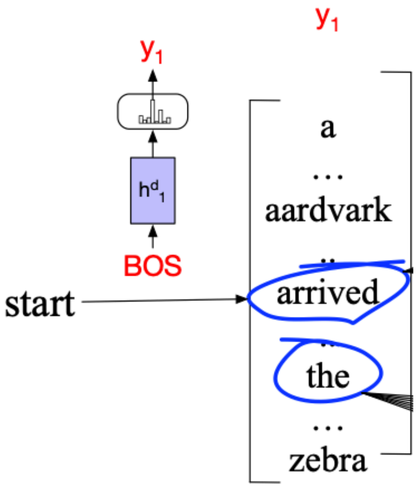
To get better outputs, we may need to keep track of multiple **branches** in a **search tree**.

Beam Search

Instead of picking the best token at each step, we'll keep track of the k best sequences at each step, where k is the **beam size**.

1. Compute softmax over vocabulary
2. Select the top-k tokens by probability
3. For each top sequence, give the model that sequence as context and compute softmaxes over possible next tokens. Find the next top k sequences (**hypotheses**) among all possible new sequences
4. If a hypothesis contains EOS, remove from frontier and decrease beam size by 1
5. Continue searching until beam size is 0

Example



t1

Example

Beam Search

Algorithm

```
function BEAMDECODE(c, beam_width) returns best paths

  y_0, h_0 ← 0
  path ← ()
  complete_paths ← ()
  state ← (c, y_0, h_0, path)          ;initial state
  frontier ← ⟨state⟩                   ;initial frontier

  while frontier contains incomplete paths and beamwidth > 0
      extended_frontier ← ⟨⟩
      for each state ∈ frontier do
          y ← DECODE(state)
          for each word i ∈ Vocabulary do
              successor ← NEWSTATE(state, i, y_i)
              extended_frontier ← ADDTOBEAM(successor, extended_frontier,
                                            beam_width)

      for each state in extended_frontier do
          if state is complete do
              complete_paths ← APPEND(complete_paths, state)
              extended_frontier ← REMOVE(extended_frontier, state)
              beam_width ← beam_width - 1
      frontier ← extended_frontier

  return completed_paths
```

Start with empty frontier

For each path, get the probability distribution over next tokens

Concatenate generated token to frontier; if this new sequence scores better than another sequence in the frontier, remove the worst frontier sequence and add this one

If an EOS token appears, remove this path from the frontier and decrease the beam size

Beam Search

Algorithm

```
function BEAMDECODE(c, beam_width) returns best paths

  y_0, h_0 ← 0
  path ← ()
  complete_paths ← ()
  state ← (c, y_0, h_0, path)          ;initial state
  frontier ← ⟨state⟩                   ;initial frontier

  while frontier contains incomplete paths and beamwidth > 0
      extended_frontier ← ⟨⟩
      for each state ∈ frontier do
          y ← DECODE(state)
          for each word i ∈ Vocabulary do
              successor ← NEWSTATE(state, i, y_i)
              extended_frontier ← ADDTOBEAM(successor, extended_frontier,
                                            beam_width)

      for each state in extended_frontier do
          if state is complete do
              complete_paths ← APPEND(complete_paths, state)
              extended_frontier ← REMOVE(extended_frontier, state)
              beam_width ← beam_width - 1
      frontier ← extended_frontier

  return completed_paths
```

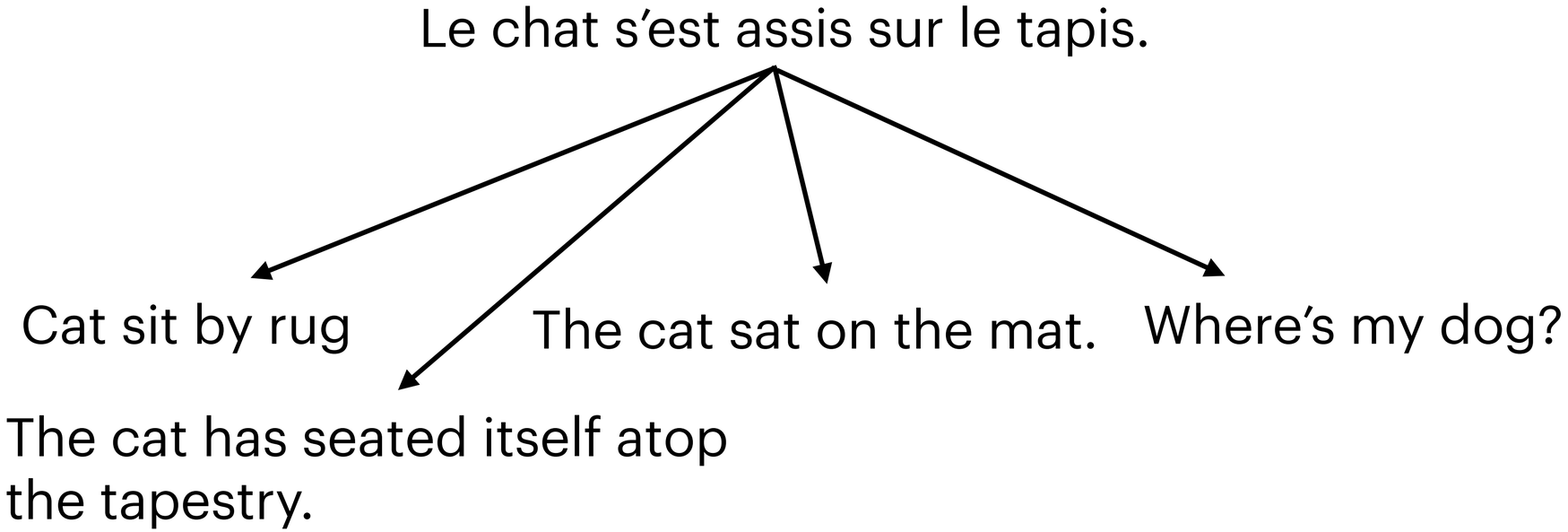
```
function ADDTOBEAM(state, frontier, width) returns updated frontier

  if LENGTH(frontier) < width then
      frontier ← INSERT(state, frontier)
  else if SCORE(state) > SCORE(WORSTOF(frontier))
      frontier ← REMOVE(WORSTOF(frontier))
      frontier ← INSERT(state, frontier)
  return frontier
```

Penalizing Overly Brief Outputs

- Beam search as-is will favor shorter hypotheses. More tokens means more multiplications by (usually very low) probabilities.
- Thus, we often apply length normalization methods, like dividing by the number of words:

$$\text{score}(y) = \frac{1}{t} \sum_{i=1}^{t} \log p(y_i \mid y_1, \ldots, y_{i-1}, x)$$

Evaluating Machine Translation Systems



Clearly, some of these translations are better than others. But how can you tell?

Adequacy: how well does the translation capture the meaning of the sentence?

Fluency: how fluent (grammatical, clear, natural) is the translation?

Evaluating Machine Translation Systems

Human Evaluation

- The most conceptually simple but logistically difficult way to evaluate is to have humans judge output quality.
- Many factors to consider:
  - Are they monolingual or bilingual native speakers?
  - Do the judges understand the subject matter?
  - What kind of judgment are they making? Yes/no, numeric rating, preference judgment?
  - Is there a reference?
  - Are they judging whether the output is grammatical? Whether the content is correct?

	How do you judge the fluency of this translation? It is:
5	Flawless English
4	Good English
3	Non-native English
2	Disfluent English
1	Incomprehensible

	How much of the meaning expressed in the gold-standard translation is also expressed in the target translation?
5	All
4	Most
3	Much
2	Little
1	None

Evaluating Machine Translation Systems

A Naïve Metric

- A simple metric to implement is the **word error rate** (WER)
- Given a **reference** translation, we can compute the word-level edit distance between the generated translation and the reference:

$$\text{WER} = \frac{S + I + D}{N}$$

S : substitutions

I : insertions

D : deletions

- Unfortunately, there are *many* correct translations for any given sentence. Many good translations will differ in word choice and order. We need something better.

Evaluating Machine Translation Systems

BLEU

- By far the most commonly used metric in MT is the **BLEU score**.
- This is basically an n-gram word precision metric.

$$\text{BLEU} = \min\left(1, \exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right) \cdot \left(\prod_{i=1}^{4} i\text{-gram precision}\right)^{\frac{1}{4}}$$

Brevity penalty

N-gram precision for $n \in \{1,2,3,4\}$

- The n-gram precision is easy to game by generating very short sequences. Brevity penalty controls for this.
- Typically computed over entire corpus, not single sentences.
- Matching larger *word clusters* yields much better scores.

Example

Input: Gestern schickte der Vorsitzende alle seine Männer nach Hause.

Reference: Yesterday , the chairman **sent all of his men home .**

Generated translation: The leader **sent all of his men home** yesterday **.**

Brevity penalty: $\min(1, \exp(1 - \frac{\text{reference-length}}{\text{output-length}})) = \exp(1 - \frac{11}{10}) \approx 0.905.$

1-gram precision: $\frac{7}{10} = 0.7$

2-gram precision: $\frac{5}{9} = 0.556$

3-gram precision: $\frac{4}{8} = 0.5$

4-gram precision: $\frac{3}{7} = 0.429$

$$0.905 \times (0.7 \cdot 0.556 \cdot 0.5 \cdot 0.429)^{\frac{1}{4}} = 0.52$$

Example

Input: Le chat est sur le tapis

Reference: **The cat** is **on** the **mat**

Generated translation: **The cat on mat**

Brevity penalty: $\min(1, \exp(1 - \frac{\text{reference-length}}{\text{output-length}}))$ = $\exp(1 - \frac{6}{4}) \approx 0.607.$

1-gram precision: $\frac{4}{4} = 1$

2-gram precision: $\frac{1}{3} = 0.3\bar{3}$

3-gram precision: 0

4-gram precision: 0

Multiplying by precisions of 0 makes the BLEU score 0.

Summary

- Machine translation (MT) systems are typically based on encoder-decoder neural networks.
- Some work has started to embrace LLMs, but they are not as popular here as in other areas of NLP.
- Beam search is the most common decoding method in MT.
- BLEU is the main evaluation metric in MT.

Multilingual NLP

Multilingual NLP

- MT requires us to map between strings in different languages.
- Possibly easier: what if we just train a model on a **multilingual corpus** (not necessarily parallel)?
- We'll go over some ways of pre-training multilingual language models and then revisit MT.

Multilingual Pre-training

- We can concatenate corpora for many languages and train on all of them
  - E.g., let's take the 104 biggest Wikipedias and put them together

Natural language processing (**NLP**) is the processing of natural language information by a computer. NLP is a subfield of computer science and is closely associated with artificial intelligence. NLP is also related to information

Doğal Dil İşleme, yaygın olarak **NLP** (*Natural Language Processing*) olarak bilinen yapay zekâ ve dilbilim alt kategorisidir. Türkçe, İngilizce, Almanca, Fransızca gibi

자연어 처리(自然語處理, natural language processing, **NLP**) 또는 **자연 언어 처리**(自然言語處理)는 인간의 언어 현상을 컴퓨터와 같은 기계를 이용해서 묘사할 수 있도록 연구하고 이를 구현하는 컴퓨터과

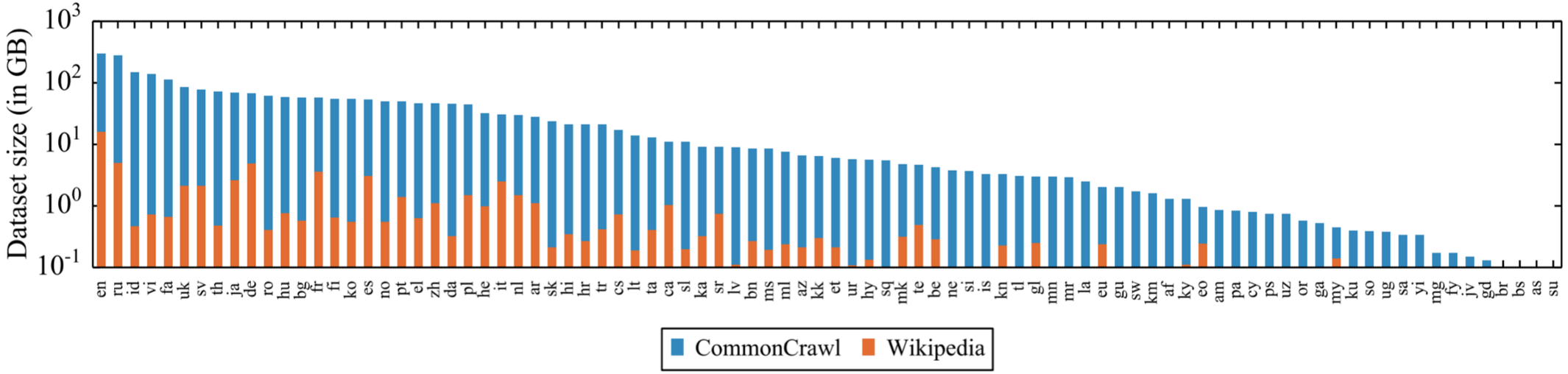
自然語言處理（英語：Natural Language Processing，缩写作 **NLP**）是人工智慧和語言學領域的交叉學科，研究计算机

- The result of this will be **multilingual embeddings**, which should hopefully align similar tokens in different languages.

Multilingual Pre-training

Language Rebalancing

Conneau et al. [2019]



- We have way more data for some languages than others. Don't want our models to overfit to a small set of languages
- We can rebalance a multilingual corpus by upsampling low-resource and downsampling high-resource languages' documents—e.g., via exponential smoothing:

$$p_{\text{new}}(\ell) = \frac{p(\ell)^\alpha}{\sum_{\ell \in L} p(\ell)^\alpha} \qquad \text{where } p(\ell) = \frac{N_\ell}{N}$$

Cross-lingual Transfer

- Alongside BERT, the same team released multilingual BERT (mBERT). It's trained the exact same way, but on multilingual data. How did it perform?

Fine-tuning \ Eval	EN	DE	ES	IT
EN	**96.82**	89.40	85.91	91.60
DE	83.99	**93.99**	86.32	88.39
ES	81.64	88.87	**96.71**	93.71
IT	86.79	87.82	91.28	**98.11**

Table 2: POS accuracy on a subset of UD languages.

Pires et al. [2019]

- Interestingly, we can fine-tune on one language, and often get decent performance on other languages!
- ...but note that all languages here share the same alphabet.

Cross-lingual Transfer

Hindi / Urdu

	HI	UR
HI	**97.1**	85.9
UR	91.1	**93.8**

Bengali / Japanese

	EN	BG	JA
EN	**96.8**	87.1	49.4
BG	82.2	**98.9**	51.6
JA	57.4	67.2	**96.5**

Pires et al. [2019]

- We also see good results when transfering across languages that use different scripts—or are even totally unrelated!
- Hindi/Urdu: very similar languages, but different writing systems
- Bengali/Japanese: different writing systems, different syntax, different morphology, different word origins

Multilingual Evaluations

- Collecting evaluation data in many languages is not easy. People often publish multilingual evaluation datasets as full research papers.

Hu et al. [2021]

Task	Corpus	|Train|	|Dev|	|Test|	Test sets	|Lang.|	Task
Classification	XNLI	392,702	2,490	5,010	translations	15	NLI
	PAWS-X	49,401	2,000	2,000	translations	7	Paraphrase
Struct. pred.	POS	21,253	3,974	47-20,436	ind. annot.	33 (90)	POS
	NER	20,000	10,000	1,000-10,000	ind. annot.	40 (176)	NER
QA	XQuAD	87,599	34,726	1,190	translations	11	Span extraction
	MLQA			4,517–11,590	translations	7	Span extraction
	TyDiQA-GoldP	3,696	634	323–2,719	ind. annot.	9	Span extraction
Retrieval	BUCC	-	-	1,896–14,330	-	5	Sent. retrieval
	Tatoeba	-	-	1,000	-	33 (122)	Sent. retrieval

- Many of these datasets are translations of base datasets and were not built with those languages specifically in mind.

Low-resource Languages

- Good performance in a language requires us to have data for that language.
- What if data for a language is scarce?
  - Language has no standard writing system
  - Language is endangered or spoken by very few people
  - Language is stigmatized

- It will be very hard to train good translation models—neural, statistical, or otherwise—when this is the case

Low-resource Languages

[Sennrich & Zhang, 2019]

ID	system	BLEU 100k	BLEU 3.2M
1	phrase-based SMT	15.87 ± 0.19	26.60 ± 0.00
2	NMT baseline	0.00 ± 0.00	25.70 ± 0.33
3	2 + "mainstream improvements" (dropout, tied embeddings, layer normalization, bideep RNN, label smoothing)	7.20 ± 0.62	31.93 ± 0.05
4	3 + reduce BPE vocabulary (14k → 2k symbols)	12.10 ± 0.16	-
5	4 + reduce batch size (4k → 1k tokens)	12.40 ± 0.08	31.97 ± 0.26
6	5 + lexical model	13.03 ± 0.49	31.80 ± 0.22
7	5 + aggressive (word) dropout	15.87 ± 0.09	**33.60** ± 0.14
8	7 + other hyperparameter tuning (learning rate, model depth, label smoothing rate)	**16.57** ± 0.26	32.80 ± 0.08
9	8 + lexical model	16.10 ± 0.29	33.30 ± 0.08

Systems that perform well in high-resource settings may not perform well in low-resource settings, and vice versa.

This was a synthetic small-data setting for German → English.

Translating with No Parallel Corpora

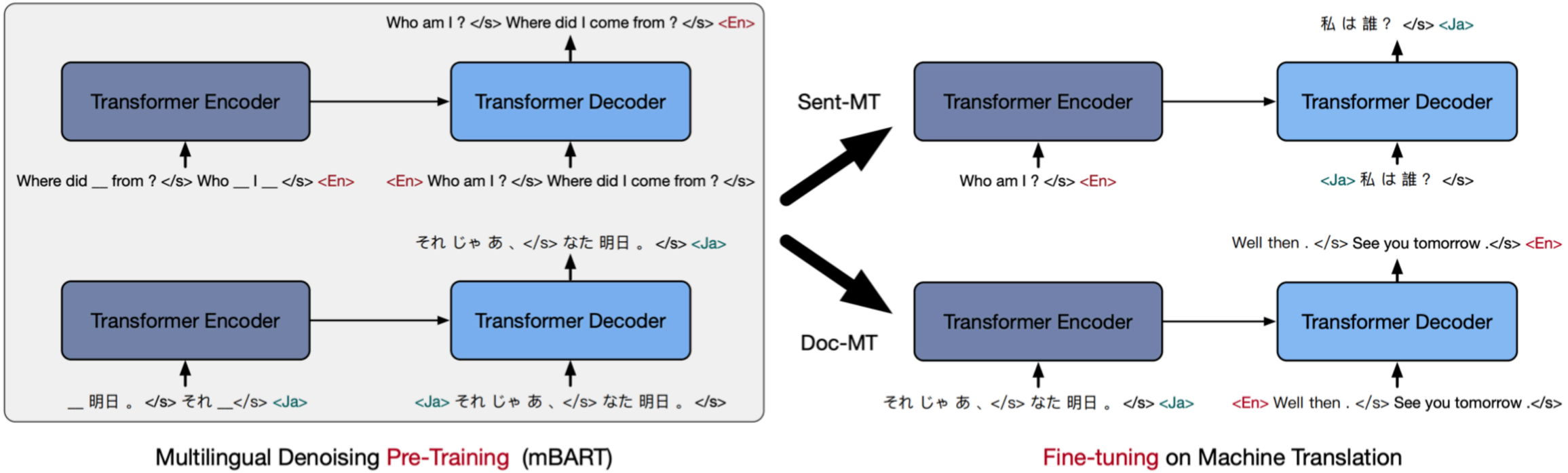
- We often want to do MT for languages where parallel corpora are scarce or non-existent.
- With subword tokenizers like BPE, we can often do this!
- Pre-training on low-resource pairs also seems to help

	BLEU		
Transfer	My→En	Id→En	Tr→En
baseline (no transfer)	4.0	20.6	19.0
transfer, train	17.8	27.4	20.3
transfer, train, reset emb, train	13.3	25.0	20.0
transfer, train, reset inner, train	3.6	18.0	19.1

Table 3: Investigating the model's capability to restore its quality if we reset the parameters. We use En→De as the parent.

Aji et al. [2020]

Multilingual Pre-training for MT



[Liu et al., 2020]

Languages	En-Gu		En-Kk		En-Vi		En-Tr		En-Ja		En-Ko	
Data Source	WMT19		WMT19		IWSLT15		WMT17		IWSLT17		IWSLT17	
Size	10K		91K		133K		207K		223K		230K	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	0.0	0.0	0.8	0.2	23.6	24.8	12.2	9.5	10.4	12.3	15.3	16.3
mBART25	**0.3**	**0.1**	**7.4**	**2.5**	**36.1**	**35.4**	**22.5**	**17.8**	**19.1**	**19.4**	**24.6**	**22.6**

Languages	En-Nl		En-Ar		En-It		En-My		En-Ne		En-Ro	
Data Source	IWSLT17		IWSLT17		IWSLT17		WAT19		FLoRes		WMT16	
Size	237K		250K		250K		259K		564K		608K	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	34.6	29.3	27.5	16.9	31.7	28.0	23.3	34.9	7.6	4.3	34.0	34.3
mBART25	**43.3**	**34.8**	**37.6**	**21.6**	**39.8**	**34.0**	**28.3**	**36.9**	**14.5**	**7.4**	**37.8**	**37.7**

[Liu et al., 2020]

Prompting LLMs in New Languages

- It turns out that you can just prompt an LLM like ChatGPT with a grammar book for a language it hasn't seen much of, and it often works well!

bo VERB to go / until
boda INTRANSITIVE VERB to be stupid
bol NOUN mouth; rim
bola NOUN ball
bolkoyal VERB to eat
bolkul NOUN lip
bolodak ADVERB just a little
bolon QUANTIFIER a little
bon VERB to bring
=bon GRAMMATICAL MARKER [comitative case marker]; with; and
bonaras (VERB) to be angry (with child)

13.1.5 With give-constructions

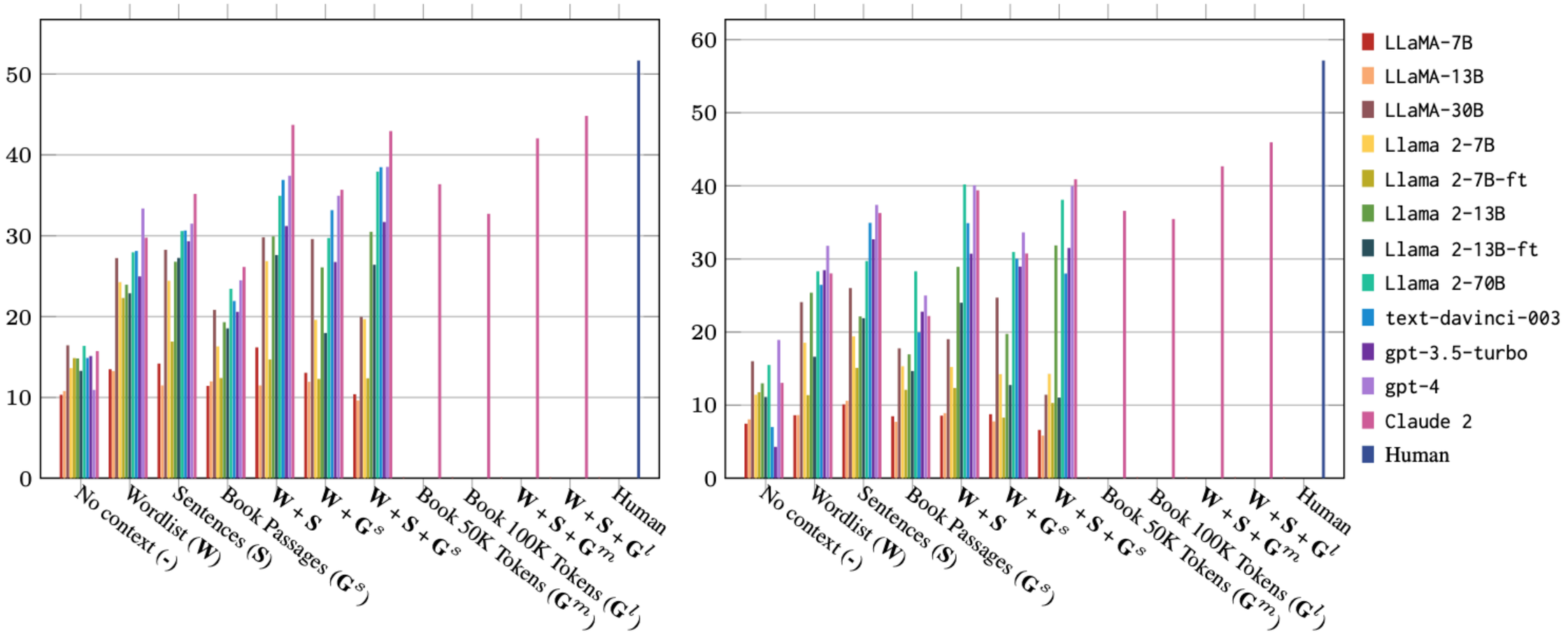
Give-constructions (§12.2.1.2) are made with a zero morpheme 'give'. They may and frequently do occur without any other verb in the clause. However, they also occur in complex predicates with predicate linker *=i*. The verb marked with *=i* precedes the recipient. The zero morpheme 'give' comes after the recipient, which makes these discontinuous complex predicates. The verbs only share their subject, and the recipient comes between the two verbs. The theme (pandanus leaf in the first example and fish in the second) is the direct object of both verbs.

(31) *naman=a padanual=at* ***rep=i*** ***ka*** ***∅***
who=FOC pandanus=OBJ get=PLNK 2SG give
'Who got pandanus [leaf] and gave it to you?'

(32) *an toni kuru ma* ***yap=i*** ***sontum=ki*** ***∅***
1SG say bring move_landwards divide=PLNK person=BEN give
'I said bring it here and divide it among people.'

[Tanzer et al., 2024]

Prompting LLMs in New Languages



MT performance goes up with parallel word lists (**W**), sentence pairs (**S**), and grammar book entries (**G**)

[Tanzer et al., 2024]