CS 577: NATURAL LANGUAGE PROCESSING

Abulhair Saparov

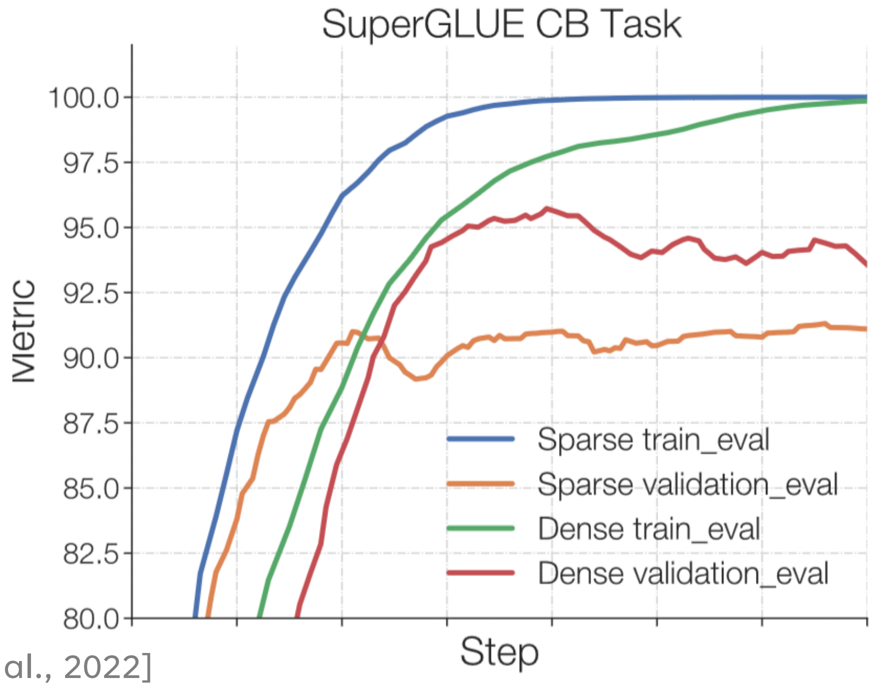
Lecture 19: Retrieval and Computational Linguistics

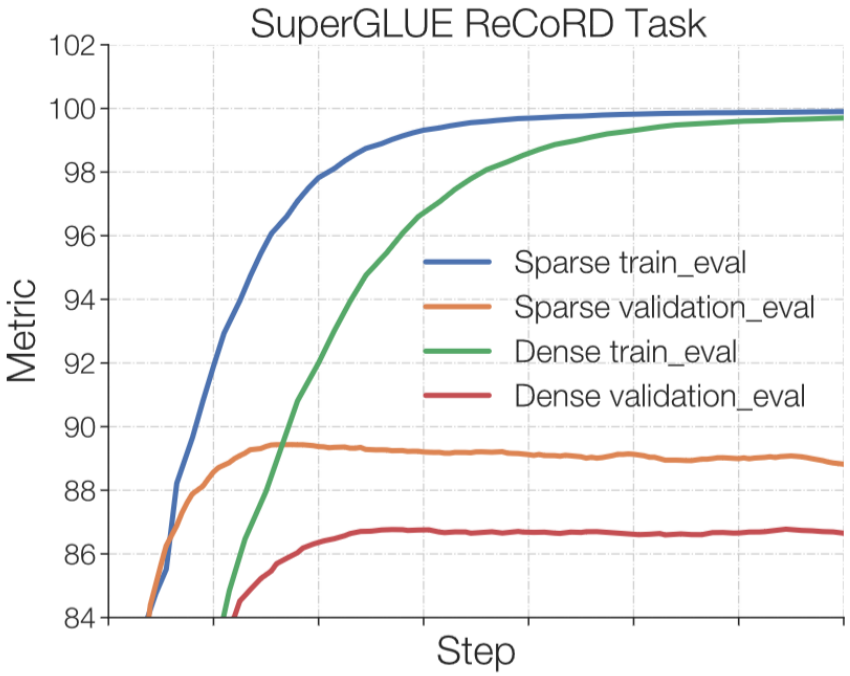
WRAPPING UP MIXTURE OF EXPERTS

- Last lecture, we discussed mixture of experts (MoE), which provides a way to break a large model (or a large component within a model) into much smaller sub-models (i.e., experts).
- For sparsely-gated MoE, forward passes become much cheaper since each forward pass can rely only on a small number (e.g., 1) of experts.
- We also discussed how to train MoE models.
    - How to mitigate training instability.

FINE-TUNING MOE MODELS

- Interestingly, MoE models have been found to overfit more easily (Zoph and Bello et al., 2022).
  - This is apparent in fine-tuning.

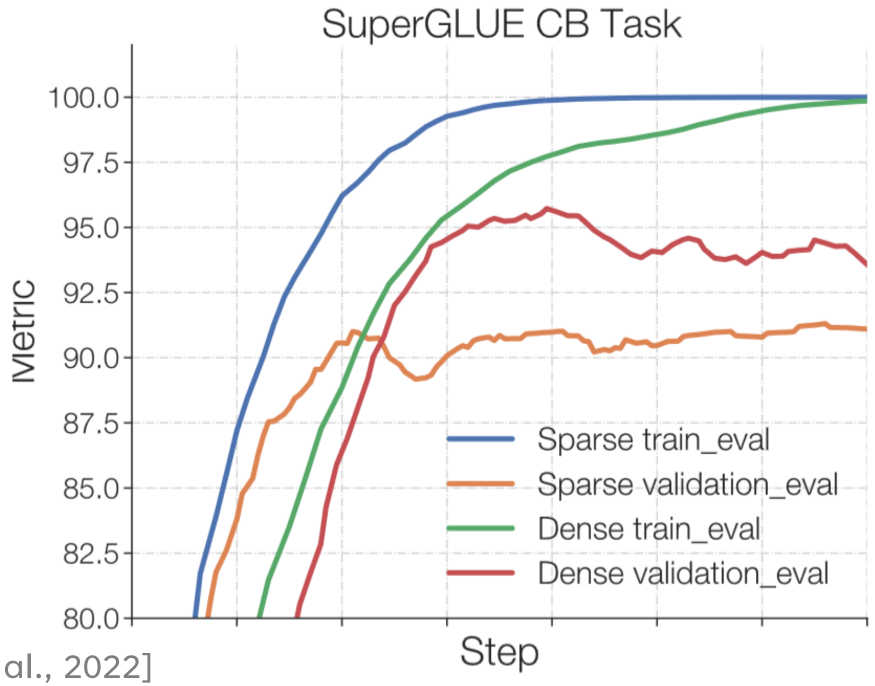


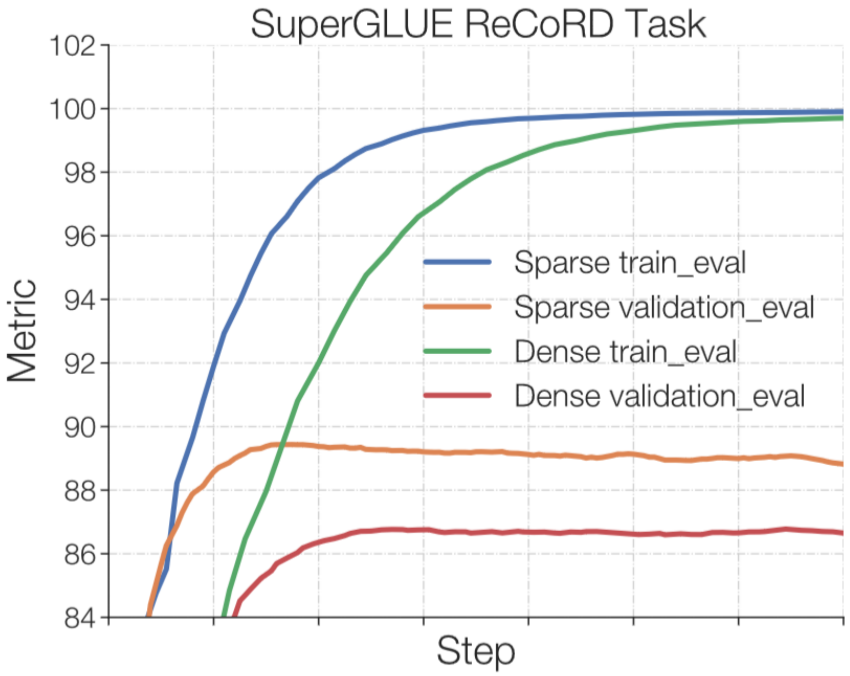


[Zoph and Bello et al., 2022]

FINE-TUNING MOE MODELS

- The SuperGLUE Commitment Bank (CB) task is an entailment task.
- The MoE model learns faster than the dense model, but overfits.

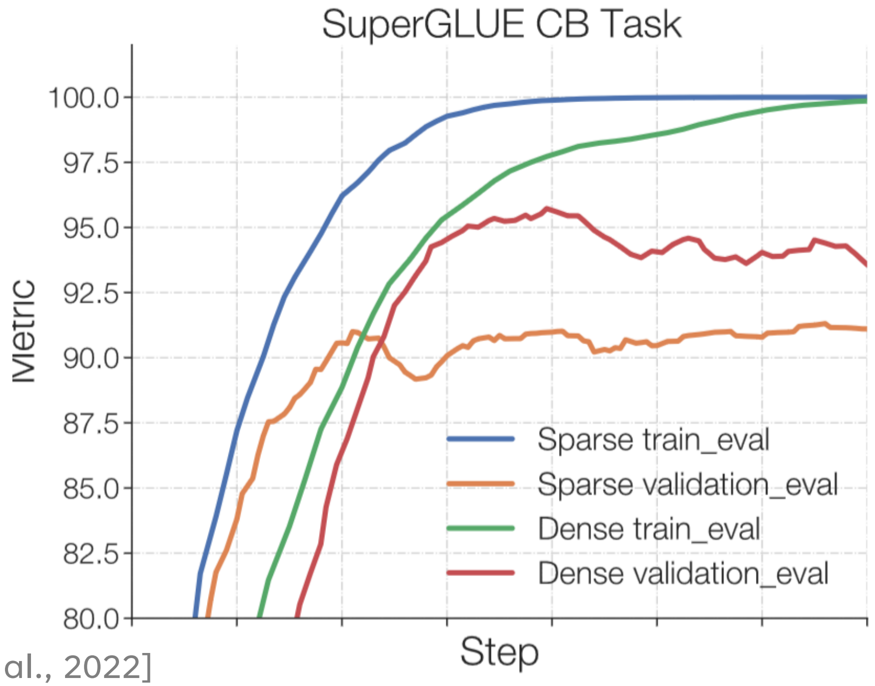


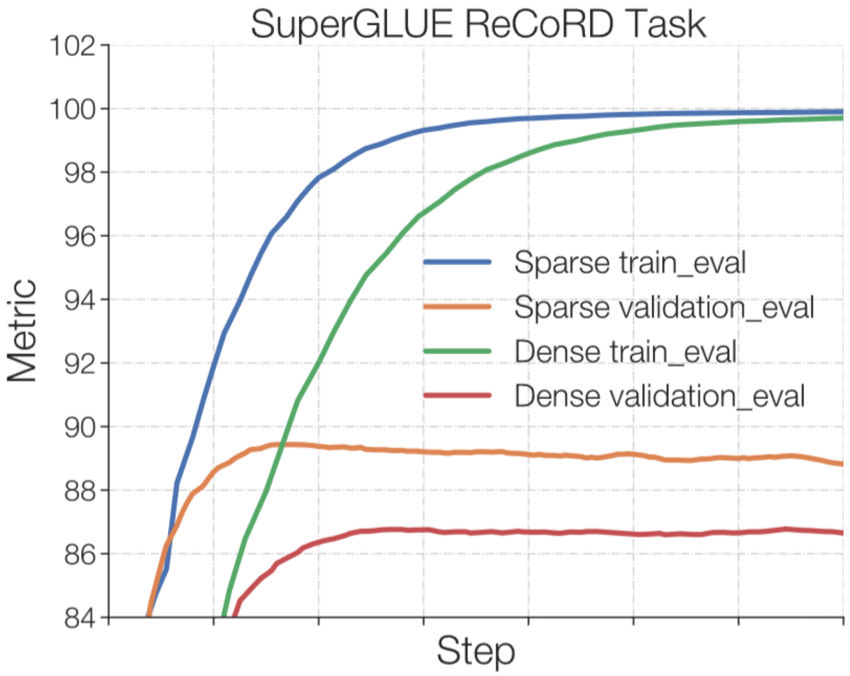


[Zoph and Bello et al., 2022]

FINE-TUNING MOE MODELS

- SuperGLUE ReCoRD is the Reading Comprehension with Commonsense Reasoning task.
- The MoE model learns faster than the dense model and generalizes better.



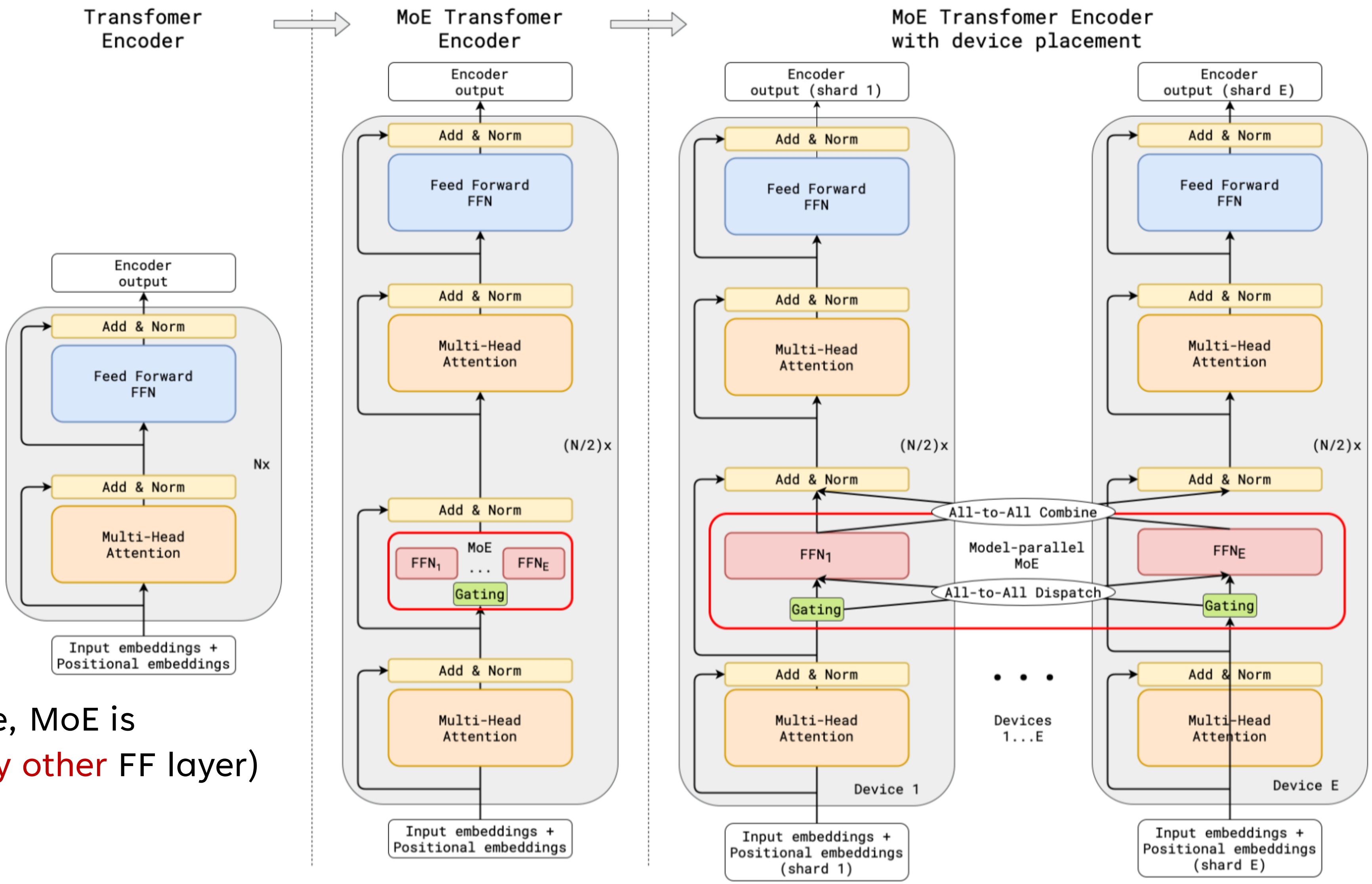


[Zoph and Bello et al., 2022]

PARALLELIZING MIXTURE OF EXPERTS

- One big advantage of MoE is that the experts provide a natural way to parallelize the model.
- Each device (i.e., GPU) can be assigned to one expert.
- Whenever we perform a forward pass with the FF layer,
  - We run the router model and determine which tokens should be sent to which experts.
  - We communicate the routing information to all devices so that each expert can perform the forward pass on their respective assigned tokens.
  - Finally, we perform an all-reduce operation to share the result across devices.
- The other parts of the model are not as memory-intensive, and so they can be replicated on each device.

PARALLELIZING MIXTURE OF EXPERTS



(in this example, MoE is applied to every other FF layer)

[Lepikhin et al., 2020]

RETRIEVAL

DISADVANTAGES OF PRETRAINED LLMS

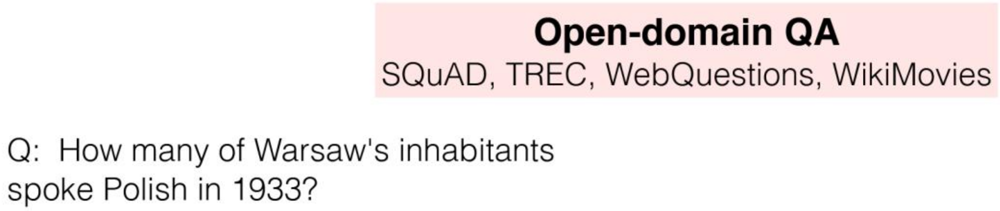
- Pretrained large models, such as LLMs, have demonstrated impressive abilities especially on tasks that are represented in their training data.
- However, there is a lot of information that is not available to pretrained large models.
  - Current events,
  - Private information unavailable on any public dataset,
  - Information/facts in the "long tail"
    - (i.e., that are poorly represented in the training data).
- Can we continually add information to the model about current events as they become available?
  - Fine-tuning? Continual pre-training?

DISADVANTAGES OF PRETRAINED LLMS

- Even if a model has the correct information and produces a response to a query,
- How can we know where this information came from?
  - It is highly intractable to search the pretraining corpus.
- Pretrained models lack information attribution.
- Can we train models to attribute information in their responses to the correct sources?

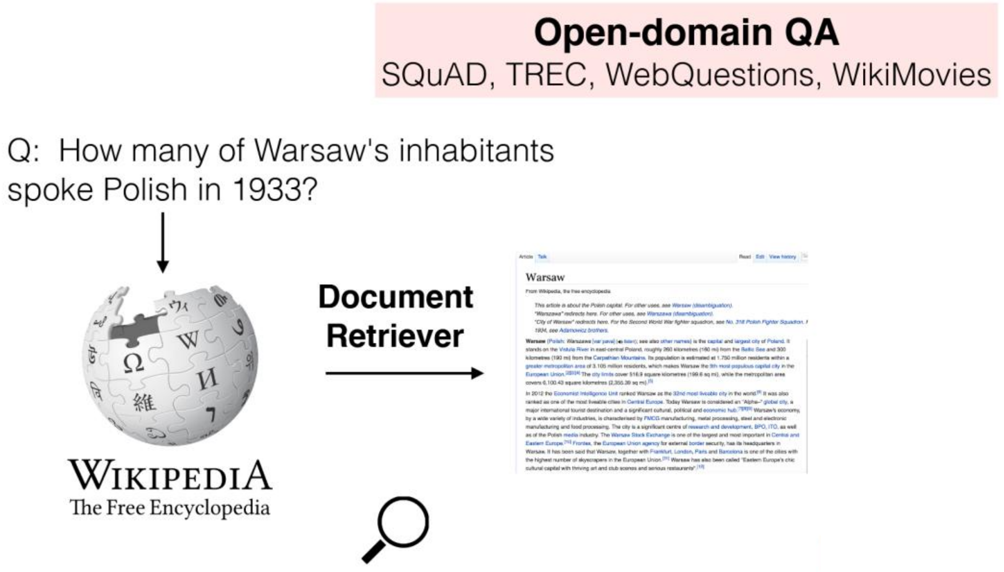
RETRIEVAL-AUGMENTED GENERATION

- Retrieval-augmented generation (RAG; Chen et al., 2017) proposes a solution to these shortcomings:
  - For a given query, use a model to retrieve a set of documents that are most relevant to the query.



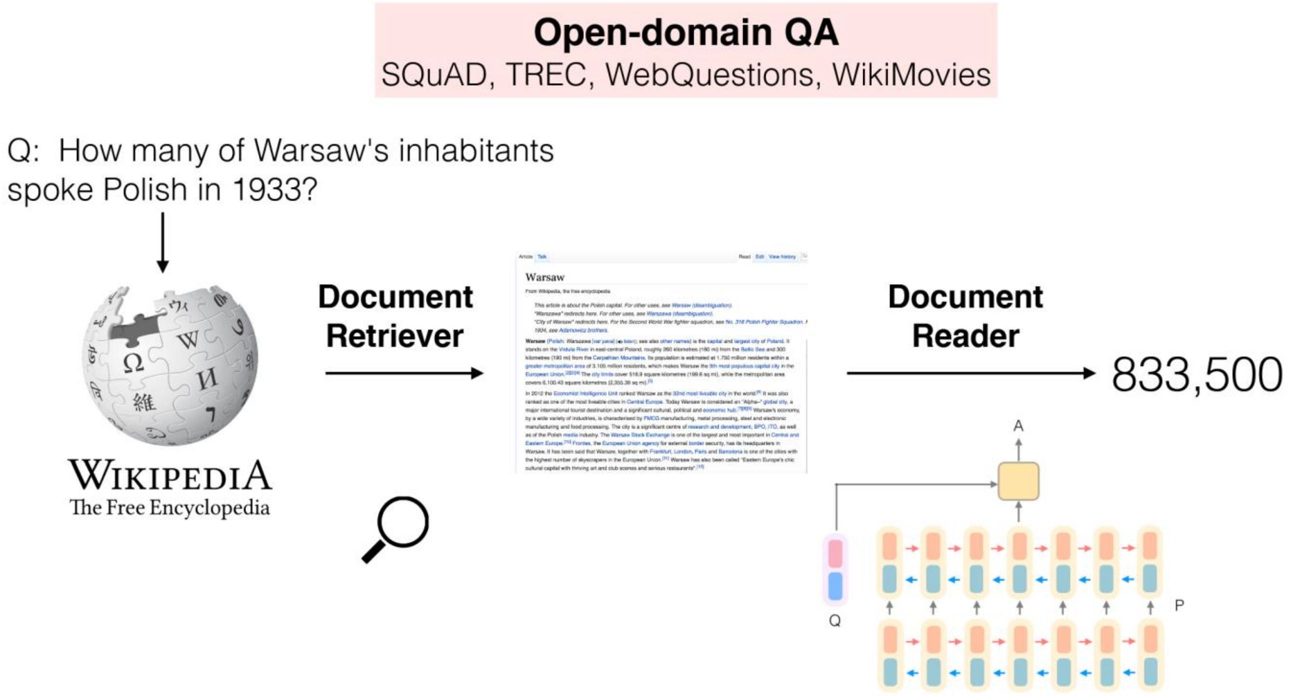
RETRIEVAL-AUGMENTED GENERATION

- Retrieval-augmented generation (RAG; Chen et al., 2017) proposes a solution to these shortcomings:
  - For a given query, use a model to retrieve a set of documents that are most relevant to the query.



RETRIEVAL-AUGMENTED GENERATION

- Retrieval-augmented generation (RAG; Chen et al., 2017) proposes a solution to these shortcomings:
  - For a given query, use a model to retrieve a set of documents that are most relevant to the query.
  - Search for the answer in the documents and return it.



RETRIEVAL-AUGMENTED GENERATION

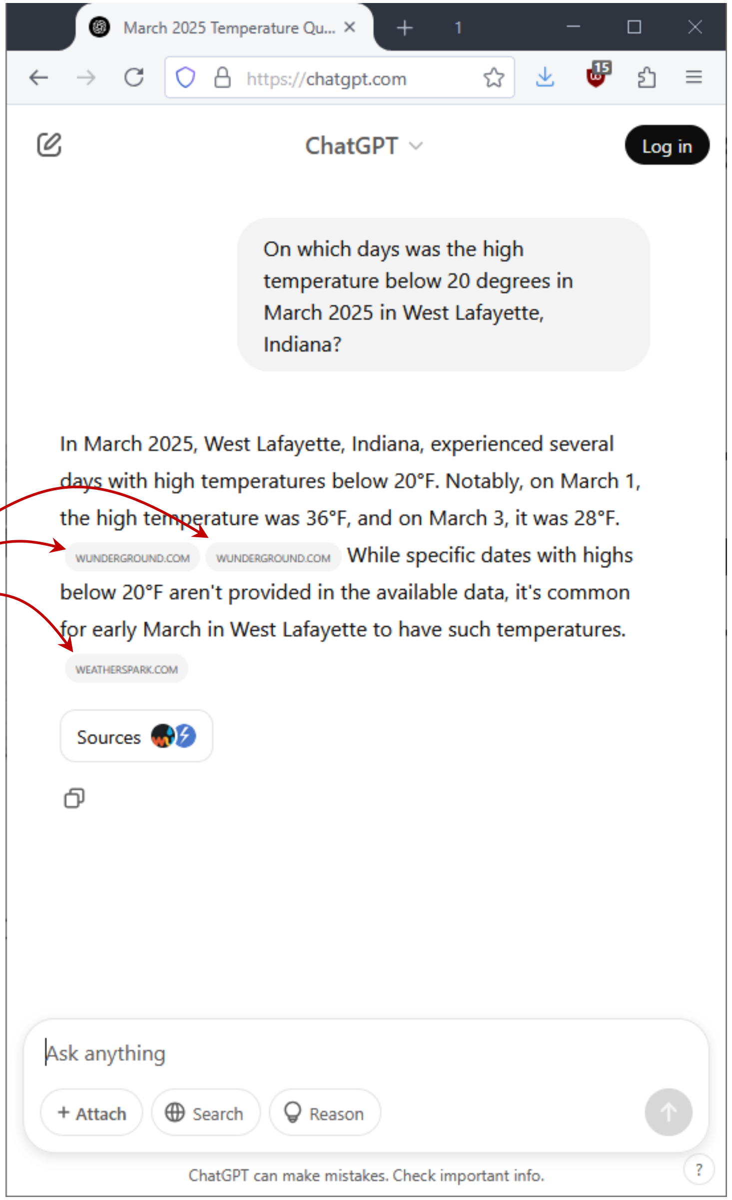
- We can imagine RAG as clearly defining two separate components:
  - Knowledge store: Large corpus of documents.
  - Retriever/generator: The model that retrieves the relevant documents and provides an answer to the query.

    (note: in principle, we can further divide this into two roles)
- Monolithic LLMs are effectively attempting to perform both roles simultaneously.
  - Both its knowledge and generation/reasoning abilities are encoded in the learned parameters.
  - RAG is an attempt to better compartmentalize these roles.

RAG EXAMPLE

- ChatGPT and many consumer-facing LLMs are able to perform retrieval to answer queries about information after their training cutoff.

attribution of retrieved documents

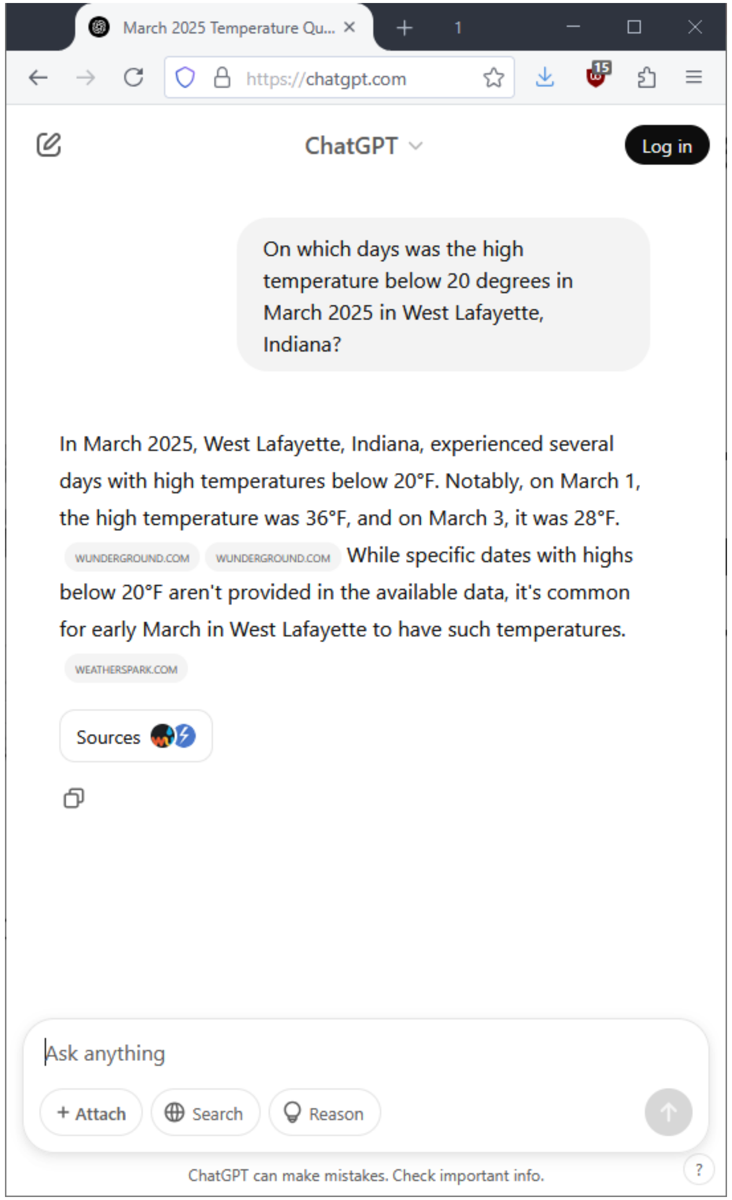


RAG EXAMPLE

- ChatGPT and many consumer-facing LLMs are able to perform retrieval to answer queries about information after their training cutoff.
- How accurate are these attributions?
- Toney-Wails (2024) found that for GPT-4, attribution accuracy varies depending on the reference type.

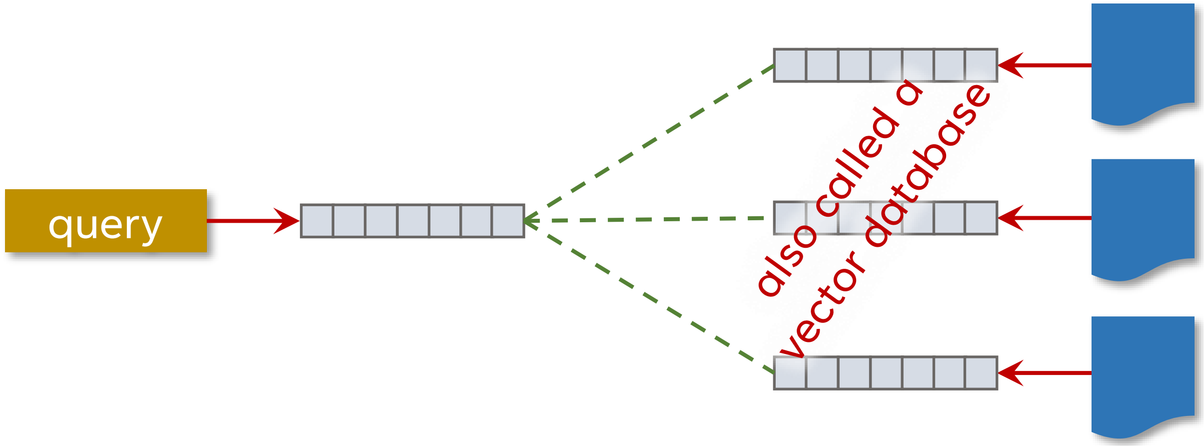
Type	Count	% Incorrect	Frequent Error
Article	297	13%	Fabrication
Textbook	600	1%	Fabrication
URL	429	42%	Page Not Found

- Clearly, more work is needed.



RETRIEVAL METHODS

- How do we retrieve relevant documents?
- One easy option: Use a search engine.
- Another option is to use embeddings:
  - Convert each document into an embedding vector.
  - Convert the query into an embedding vector.
- Retrieve the k documents with embeddings closest to the query embedding.



- The embeddings can be obtained from several methods.
- E.g., the activations in the last layer of a transformer.

LEARNING EMBEDDINGS FOR RETRIEVAL

- Another way to obtain embeddings is to train a model to produce them.
- Suppose we have a dataset containing queries,
  - Where each query is annotated with a set of positive and negative examples of retrieved documents (D_+ and D_-, respectively).
- We can then learn the embedding function f using a contrastive loss (e.g., hinge loss):

$$L(\theta, q) = \sum_{d_+ \in D_+} \sum_{d_- \in D_-} \max\{0, \mathtt{dist}(f_\theta(q), f_\theta(d_+)) - \mathtt{dist}(f_\theta(q), f_\theta(d_-))\}.$$

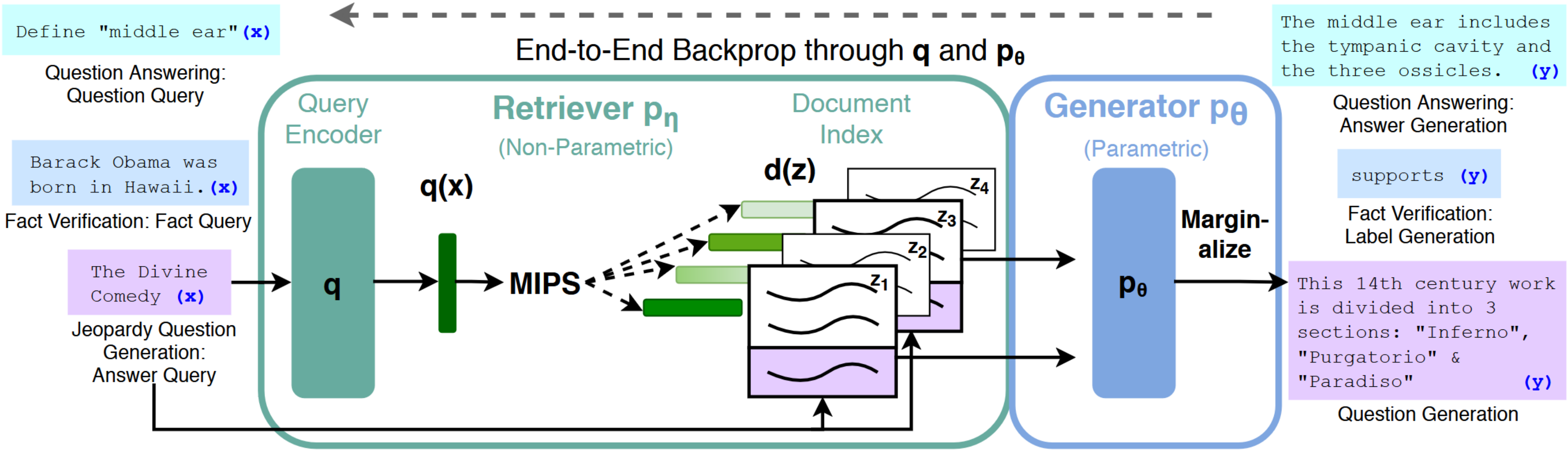
- Where dist is a vector distance function (e.g., L2, cosine).
- Examples: DPR (Karpukhin et al., 2020),
  - Contriever (Izacard et al., 2022).

GENERATION AFTER RETRIEVAL

- Once we have a set of retrieved documents, what do we do next?
- Perhaps the simplest approach is to simply provide the documents followed by the query in the prompt of a language model.
  - The LM can be fine-tuned to *only* use the information in context rather than information from pretraining.
  - Fine-tuning can also help the model to learn to properly attribute/cite the information in context.

END-TO-END TRAINING OF RETRIEVAL+GENERATION

- Another approach is to treat the full retrieval+generation pipeline as a single model and to train it end-to-end.
- This was the approach suggested by Lewis et al. (2021).



[Lewis et al., 2021]

END-TO-END TRAINING OF RETRIEVAL+GENERATION

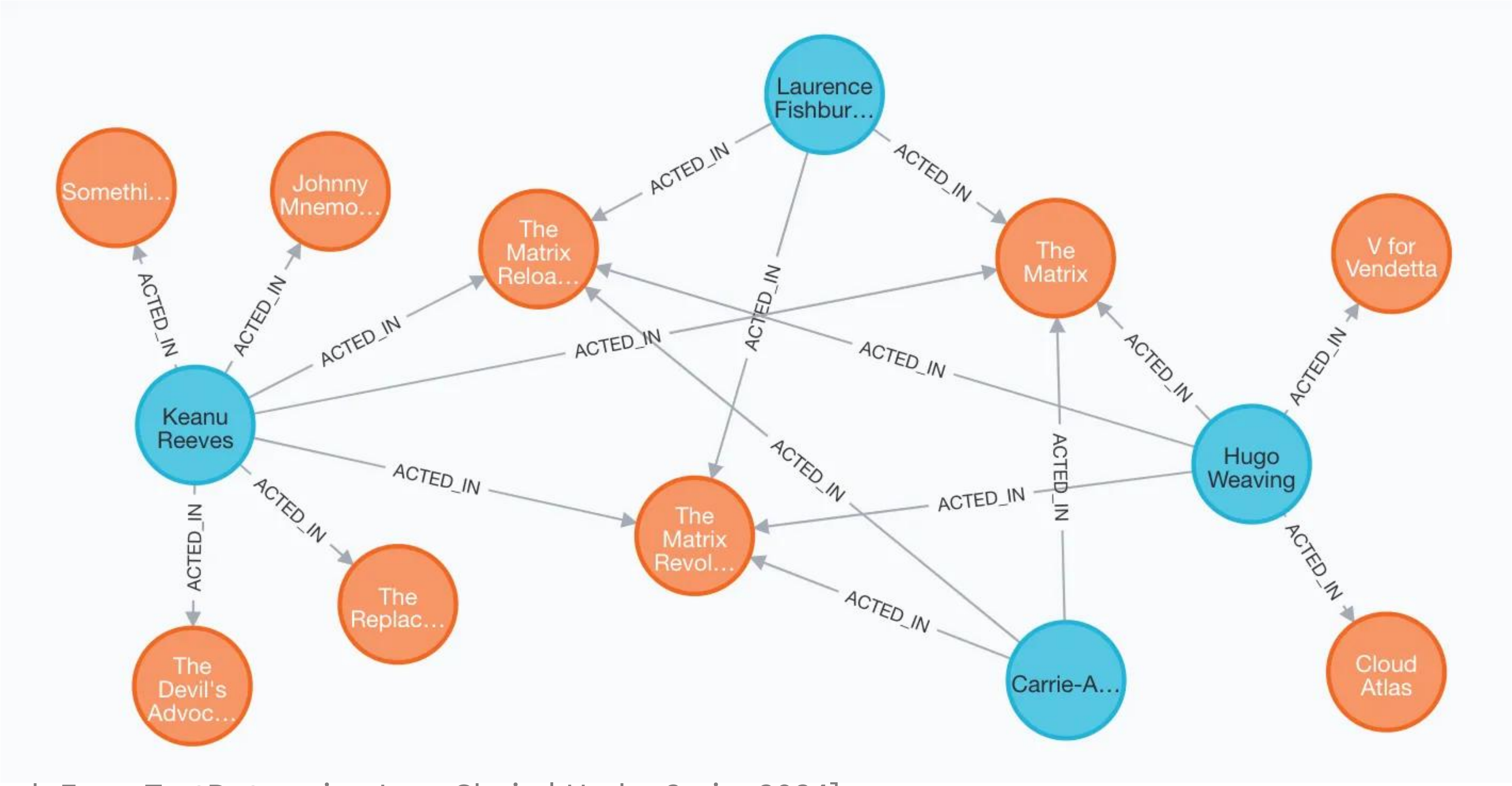
- Another approach is to treat the full retrieval+generation pipeline as a single model and to train it end-to-end.
- This was the approach suggested by Lewis et al. (2021).
- So long as we have a dataset with queries annotated with ground truth answers, we can use the loss on the final answer vs the ground truth answer.
- We can use gradient descent/backprop to update the parameters of both the generator and the retriever.
- But this approach can be expensive, especially if the retriever or generator are very large.
  - Idea: Use parameter-efficient fine-tuning.

GRAPH RAG

- Rather than retrieving information from an unstructured collection of documents,
- What if we retrieve information from structured datasets, such as knowledge graphs?
- A knowledge graph is a graph where each edge represents a fact.
  - Suppose the edge (u, v) has label r.
  - This represents the relation $r(u, v)$.
  - For example, the edge `keanu_reaves -> the_matrix` with label `acted_in` represents the fact `acted_in(keanu_reaves,the_matrix)`.

GRAPH RAG

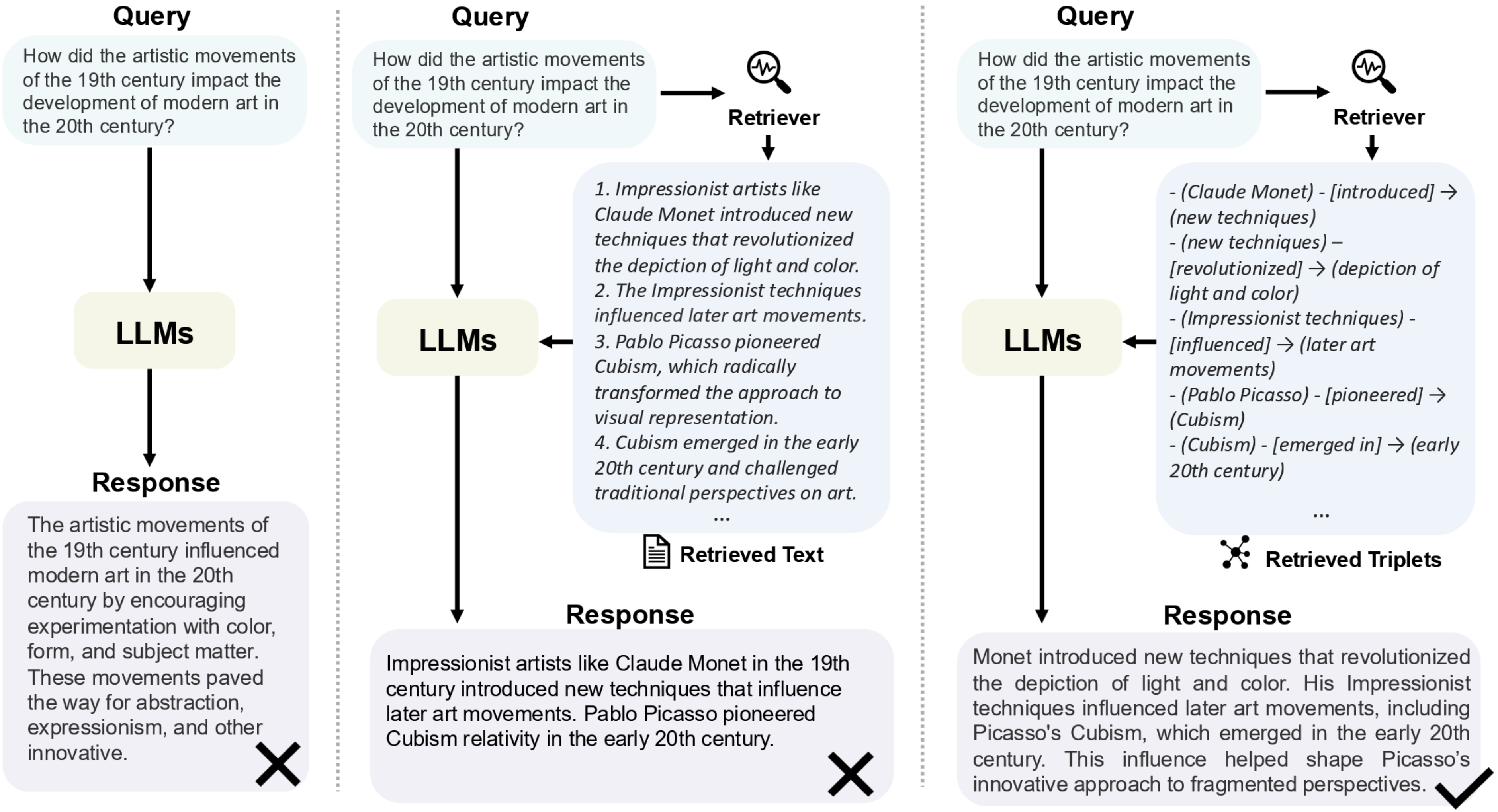
- An example of a very small knowledge graph:



[Raja, Build Knowledge Graph From TextData using LangChain | Under 2min, 2024]

GRAPH RAG

- We can train a retriever to extract the k most relevant relations from a knowledge graph.
- This can be useful if the knowledge graph has information that is missing from documents.
- Can use other structured data (tables, time series).



[Peng and Zhu et al., 2024]

WHEN DO WE RETRIEVE?

- In the simplest RAG setting, we retrieve once at the beginning, and then have the generator produce the final output.
- However, for some tasks (such as multi-hop reasoning/question answering), it may benefit to retrieve multiple times throughout generation.
  - E.g., train the generator to produce a special "search token" whenever we want to perform another retrieval step (Schick et al., 2023).
  - Detect when the generator becomes uncertain by inspecting the probabilities of the output tokens (Jiang et al., 2023).
    - If the log probability drops below a threshold, perform another retrieval step.

TOOLFORMER

- Schick et al. (2023) fine-tuned `GPT-J-6B` to produce special “API-call” tokens.
- These tokens would invoke external tools, such as web search, a calculator, a machine translation model, etc.
- They call their method Toolformer.

The New England Journal of Medicine is a registered trademark of **[QA(“Who is the publisher of The New England Journal of Medicine?”) → Massachusetts Medical Society]** the MMS.

Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

The name derives from “la tortuga”, the Spanish word for **[MT(“tortuga”) → turtle]** turtle.

The Brown Act is California’s law **[WikiSearch(“Brown Act”) → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their meetings open to the public.

TOOLFORMER

- Despite being much smaller than OPT or GPT-3, Toolformer was able to outperform them on question answering and math problem solving tasks.

Model	SQuAD	Google-RE	T-REx
GPT-J	17.8	4.9	31.9
GPT-J + CC	19.2	5.6	33.2
Toolformer (disabled)	22.1	6.3	34.9
Toolformer	**33.8**	**11.5**	**53.5**
OPT (66B)	21.6	2.9	30.1
GPT-3 (175B)	26.8	7.0	39.8

Model	ASDiv	SVAMP	MAWPS
GPT-J	7.5	5.2	9.9
GPT-J + CC	9.6	5.0	9.3
Toolformer (disabled)	14.8	6.3	15.0
Toolformer	**40.4**	**29.4**	**44.0**
OPT (66B)	6.0	4.9	7.9
GPT-3 (175B)	14.0	10.0	19.8

[Schick et al., 2023]

COMPUTATIONAL LINGUISTICS

COMPUTATIONAL LINGUISTICS

- In the first part of the course, we covered a wide variety of methods in modern NLP.
- We focused on the empirical side of NLP.
  - What tools are available to solve NLP tasks?
  - What are the best practices in the application of such tools?
- But how can we expect to solve NLP tasks if we don’t understand language itself?
- Linguistics is the scientific study of language.
- Computational linguistics (CL) is the application of computation in linguistics.
  - I.e., Can we describe language understanding as a computational process?
  - (this is the CL in ACL, EACL, NAACL, TACL, etc)

WHY STUDY COMPUTATIONAL LINGUISTICS?

- Since we have focused on empirical methods in the first half of the course, you are well equipped to try to solve NLP tasks empirically.
  - I.e., take some off-the-shelf ML model, train/fine-tune it on some corpus of data, and hope for the best.
- But is this always the best way to solve such problems?
- Consider the problem of medical diagnosis.
  - You are presented with many examples of patients, each with different symptoms, histories, etc.
  - You have access to a lot of data about various medical treatments.
  - The data contains past examples of treatments on patients, and whether those treatments were successful, any side effects, etc.

WHY STUDY COMPUTATIONAL LINGUISTICS?

- If we took a purely empirical approach to medicine, we can imagine training a large-scale black-box model on this medical data.
- That approach may work, with sufficient data.
- To what extent can we expect such a model to generalize out-of-distribution?
  - If a new medical treatment (e.g., a new drug) is developed, will the model be able to apply it readily?
- This approach ignores the vast knowledge we have accumulated about biology, anatomy, and chemistry.
  - Perhaps we can inspect the chemical structure of the new drug and compare it to similar drugs.
  - Or we can examine its structure to predict its mechanism of action.

WHY STUDY COMPUTATIONAL LINGUISTICS?

- Take another example of autonomous navigation.
- Suppose we wish to develop the navigation system of a spacecraft.
- A purely empirical approach would be to provide it with many training examples of previous actions and their corresponding outcomes.
  - E.g., after firing the rockets at half thrust for 10 seconds, the spacecraft's velocity changed by...
- If we were train such a model, but then change the mass of the spacecraft, or change the type of fuel,
  - Can we really expect the model to generalize correctly?
- Such an approach would ignore everything we know about physics.

WHY STUDY COMPUTATIONAL LINGUISTICS?

- In addition, empirical methods can readily change over the course of a few years.
- Consider the state of empirical methods in NLP before 2017.
  - The predominant paradigms for training and using NLP models was entirely different.
- Can we really be confident that the empirical methods we covered in the first half of the course will still be useful/relevant 10 years in the future?
- However, the nature of language, its properties, and what it means to understand language, does not change so readily.

WHAT IS LANGUAGE?

- There are many definitions.
  - Some definitions are more useful than others in certain contexts.
- Formal language theory is the study of the internal structural patterns of languages.
- In formal language theory, we start with an alphabet (or vocabulary), which is a set of elementary symbols in the language.
  - E.g., Σ = {‘`0`’,‘`1`’,‘`2`’,‘`3`’,‘`4`’,‘`5`’,‘`6`’,‘`7`’,‘`8`’,‘`9`’,‘`plus`’,‘`minus`’,‘`equals`’}.
  - This can be a set of letters, phonemes, tokens, words, etc.
- A string is a sequence of symbols from the alphabet.
  - E.g., ‘`7 plus 3 equals 10`’,
  - ‘`minus 01 plus equals 7 equals`’.

WHAT IS LANGUAGE?

- In formal language theory, a language is defined as a set of strings.
- Note that this is a very broad definition.
- The set of strings containing only 1's is a language.
  `L = {'1', '11', '111', '1111', …}`
- The empty set is a language.
- The set of strings expressing true mathematical expressions is a language.
  `L = {'0 equals 0', '1 plus 2 equals 3', '20 equals 29 minus 9', …}`
- The following set of three strings is a language:
  `L = {'plus minus 1', '249 equals', ''}`

LANGUAGE RECOGNITION

- The basic computational task in formal language theory is recognition:
- Given a language $\mathtt{L}$, and a string s, is $s \in \mathtt{L}$?
- For finite languages, we can simply iterate over each element in $\mathtt{L}$ and compare it to s.
  - But this may take a long time if $\mathtt{L}$ is large.
  - Languages often have regular structure that we can exploit to speed-up recognition.
  - This is required for infinite languages.
- E.g., the language of strings containing only 1's is easy to recognize:
  - Simply check every symbol in the string is a '`1`'.
  - The running time is simply the length of s.

LANGUAGE RECOGNITION

- E.g., the language containing all well-formed mathematical expressions (not necessarily true):

  `L = {‘0 equals 0’, ‘1 plus 2 equals 9’, ‘20 equals 29 minus 7’, …}`

- This language is not as easy to recognize as the previous example, but there does exist an algorithm that will do so in $O(|s|^3)$.
  - We will learn about such algorithms in a later lecture.
- Language recognition in *natural language* can be described as grammaticality checking.
  - E.g., ‘`I run to the store`’ and ‘`Alex runs to the store`’ are grammatical,
  - But ‘`I runs to the store`’ and ‘`Alex run to the store`’ are not.

WHAT IS LANGUAGE, REALLY?

- But languages are more than just sets of strings,
  - And "understanding" language is more than just checking grammaticality, or language recognition.
- Language conveys meaning.
  - E.g., '`1 plus 2 equals 3`' has the meaning of `1 + 2 = 3`.
  - `1 + 2 = 3` is a logical form.
- Logical forms can be truth-functional, such as in the above example.
  - We can say `1 + 2 = 3` is true.
  - The logical form of '`1 plus 2 equals 9`' is false.
  - The logical form of '`Mercury is the closest planet to the sun`' is true.

SEMANTICS AND REASONING

- Logical forms capture the meaning of sentences/utterances.
- The task of converting from sentence to logical form is called semantic parsing.
- The task of converting from logical form to sentence is called generation.
- Some logical forms are amenable to reasoning.
  - E.g., logic.
  - The logical form of ‘`Alex is a cat`’ is `cat(alex)`,
  - ‘`All cats are mammals`’ has meaning $\forall x(\texttt{cat}(x)$ `->` $\texttt{mammal}(x))$.
  - We can use deduction rules to deduce `mammal(alex)`.
  - Then we use generation to convert this into ‘`Alex is a mammal.`’

SEMANTICS AND REASONING

- Logical forms can be a language, like logic, or a programming language.
  - Maybe even real-valued vector embeddings?
- The choice of the representation for the logical form is called the logical formalism.
- The correct choice of logical formalism is not always clear.
- Consider the example where '`Alex is a cat`' has meaning `cat(alex)`.
  - What if instead the sentence was '`Alex, the cat that my mom gave me, had probably spent all day sleeping lazily in the sun`'?
  - How to represent '`probably`' in logic? Or '`sleeping lazily`'?
- The study of how to formally represent the meaning of natural language is called formal semantics.

SEMANTICS AND REASONING

```
Q: Alex has 5 cakes. She then got 2 more
and gave away 4. How many cakes does Alex
have?

A: ???
```

Logical form

```
alex_cake_count = 5
alex_cake_count += 2 - 4
```

Reasoning

Logical form

```
alex_cake_count = 3
```

semantic parsing

natural language generation

Natural language

'Alex has 5 cakes. She then got 2 more and gave away 4.'

language modeling?

Natural language

'Alex now has 3 cakes.'

Do LLMs follow this process?

COMPUTATIONAL LINGUISTICS ROADMAP

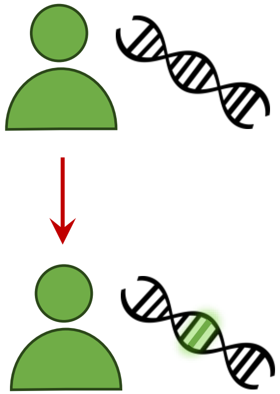
- We will cover these topics and consider possible answers to all of these research questions over the next few lectures.
- We will follow the following rough roadmap:
  - Morphology
    - What are words? How are words constructed? How do they attain meaning?
  - Syntax
    - How are words arranged to form sentences? What is a grammar?
  - Semantics
    - How to represent the meaning of sentences?
  - Discourse and pragmatics
    - How does context contribute to meaning?

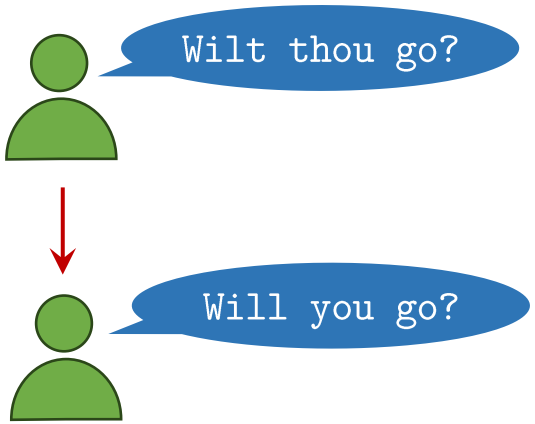
LANGUAGE IS ALWAYS CHANGING

- Languages are constantly changing.
- When humans acquire language, they often don't learn to exactly replicate the way their parents/teachers use language.
  - Sometimes, "mistakes" can turn into new rules.
    - E.g., "work" is traditionally uncountable (i.e., it has no plural form).
    - But you will now often see "works" used, such as in Related Work(s) sections of academic papers.
  - New words are created (e.g., "google", "skyscraper", etc).
  - Old words are lost (e.g., "alsike", "thee", "nigh", etc).

LANGUAGE IS ALWAYS CHANGING

- The imperfect teaching of language from parent/teacher to child is compared with the passing of genetic information from parent to child.
  - The process of copying DNA is not perfect.
  - There will be small changes with each generation.
- But languages change faster than genes.





LANGUAGE IS ALWAYS CHANGING

- Consider English:
  - Old English (circa 1000 CE):

    `Faeder ure, thu the eart on heofonum, si thin nama gehalgod…`
  - Middle English (1384 CE):

    `Oure fadir that art in heuenes, halwid be thi name…`
  - Early Modern English (1534 CE):

    `O oure father which arte in heven, halowed be they name…`
  - Early Modern English (1611 CE):

    `Our father which art in heauen, hallowed by they name…`

LANGUAGE IS ALWAYS CHANGING

- Consider English:
  - You may notice that older pronunciations of English words closely follow the spelling.
  - E.g., “knight” is pronounced /naɪt/ in Modern English.
    - Why does this word have a “k” and a “gh”?

    This is an example of the International Phonetic Alphabet (IPA).
  - In Middle English, it was pronounced /kniːxt/.
  - Another example: “Wednesday” is pronounced /ˈwɛnzdeɪ/.
    - What happened to the first “d”?
    - This word’s etymology (i.e., origin) is from a word meaning “Odin’s day.”

LANGUAGE IS ALWAYS CHANGING

- If languages can evolve analogously to biological organisms, we can study their genetic relationships.
- Some languages are more closely related than others.
- Let’s examine some words in English and other similar languages (Wikipedia):

English	West Frisian	Dutch	Low German[77]	German	Icelandic	Norwegian (Nynorsk)	Swedish	Danish	Gothic †
apple	apel	appel	Appel	Apfel	epli	eple	äpple	æble	*apel*[78]
can	kinne	kunnen	känen	können	kunna	kunne, kunna	kunna	kunne	kunnan
daughter	dochter	dochter	Dochter	Tochter	dóttir	dotter	dotter	datter	dauhtar
dead	dea	dood	dod	tot	dauður	daud	död	død	dauþs
deep	djip	diep	deip	tief	djúpur	djup	djup	dyb	diups
earth	ierde	aarde	Ir(d)	Erde	jörð	jord	jord	jord	airþa
egg[79]	aei, aai	ei	Ei	Ei	egg	egg	ägg	æg	*addi[80]
fish	fisk	vis	Fisch	Fisch	fiskur	fisk	fisk	fisk	fisks

LANGUAGE IS ALWAYS CHANGING

- If languages can evolve analogously to biological organisms, we can study their genetic relationships.
- Some languages are more closely related than others.
- Let's examine some words in English and other similar languages (Wikipedia):

West Germanic					North Germanic				East Germanic
Anglo-Frisian		Continental			West		East		
English	West Frisian	Dutch	Low German[77]	German	Icelandic	Norwegian (Nynorsk)	Swedish	Danish	Gothic †
apple	apel	appel	Appel	Apfel	epli	eple	äpple	æble	*apel*[78]
can	kinne	kunnen	känen	können	kunna	kunne, kunna	kunna	kunne	kunnan
daughter	dochter	dochter	Dochter	Tochter	dóttir	dotter	dotter	datter	dauhtar
dead	dea	dood	dod	tot	dauður	daud	död	død	dauþs
deep	djip	diep	deip	tief	djúpur	djup	djup	dyb	diups
earth	ierde	aarde	Ir(d)	Erde	jörð	jord	jord	jord	airþa
egg[79]	aei, aai	ei	Ei	Ei	egg	egg	ägg	æg	*addi[80]
fish	fisk	vis	Fisch	Fisch	fiskur	fisk	fisk	fisk	fisks

COMPARATIVE LINGUISTICS

- But notice that these comparisons are not perfect.
- Languages can borrow words or features from other nearby languages.
  - E.g., English borrowed “egg” from Old Norse during the Viking invasions.
  - As well as a large amount of vocabulary from French, Latin, Greek, etc.

West Germanic					North Germanic				East Germanic	Reconstructed Proto-Germanic[76]
Anglo-Frisian		Continental			West		East			
English	West Frisian	Dutch	Low German[77]	German	Icelandic	Norwegian (Nynorsk)	Swedish	Danish	Gothic †	
apple	apel	appel	Appel	Apfel	epli	eple	äpple	æble	*apel*[78]	*ap(u)laz
can	kinne	kunnen	känen	können	kunna	kunne, kunna	kunna	kunne	kunnan	*kanna
daughter	dochter	dochter	Dochter	Tochter	dóttir	dotter	dotter	datter	dauhtar	*đuχtēr
dead	dea	dood	dod	tot	dauður	daud	död	død	dauþs	*đauđaz
deep	djip	diep	deip	tief	djúpur	djup	djup	dyb	diups	*đeupaz
earth	ierde	aarde	Ir(d)	Erde	jörð	jord	jord	jord	airþa	*erþō
egg[79]	aei, aai	ei	Ei	Ei	egg	egg	ägg	æg	*addi[80]	*ajjaz
fish	fisk	vis	Fisch	Fisch	fiskur	fisk	fisk	fisk	fisks	*fiskaz

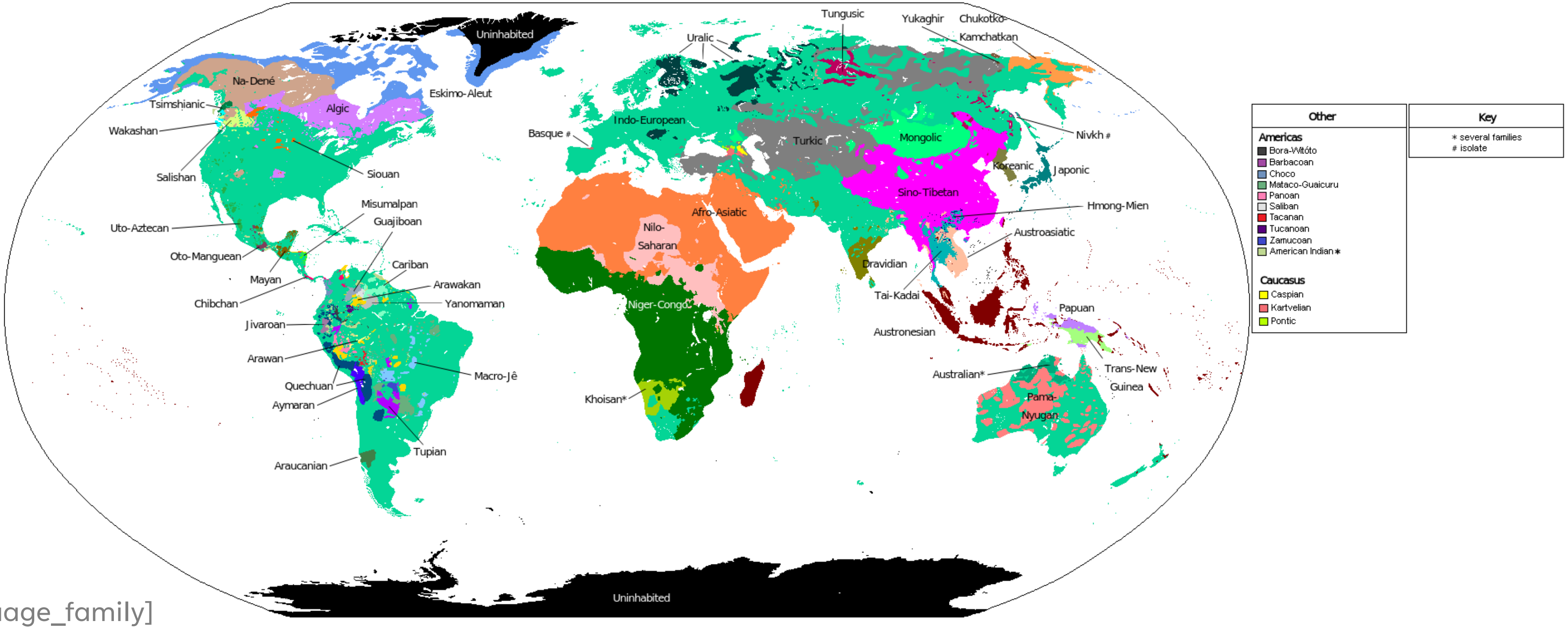
COMPARATIVE LINGUISTICS

- Comparative linguistics is the study of the relationships between languages.
  - What are the most likely sound changes that occurred as languages evolved over time?
  - "Ancestor" or proto-languages can be reconstructed by "undoing" these changes.

West Germanic					North Germanic				East Germanic	Reconstructed Proto-Germanic[76]
Anglo-Frisian		Continental			West		East			
English	West Frisian	Dutch	Low German[77]	German	Icelandic	Norwegian (Nynorsk)	Swedish	Danish	Gothic †	
apple	apel	appel	Appel	Apfel	epli	eple	äpple	æble	*apel*[78]	*ap(u)laz
can	kinne	kunnen	känen	können	kunna	kunne, kunna	kunna	kunne	kunnan	*kanna
daughter	dochter	dochter	Dochter	Tochter	dóttir	dotter	dotter	datter	dauhtar	*đuχtēr
dead	dea	dood	dod	tot	dauður	daud	död	død	dauþs	*đauđaz
deep	djip	diep	deip	tief	djúpur	djup	djup	dyb	diups	*đeupaz
earth	ierde	aarde	Ir(d)	Erde	jörð	jord	jord	jord	airþa	*erþō
egg[79]	aei, aai	ei	Ei	Ei	egg	egg	ägg	æg	*addi[80]	*ajjaz
fish	fisk	vis	Fisch	Fisch	fiskur	fisk	fisk	fisk	fisks	*fiskaz

LANGUAGE FAMILIES

- Languages are grouped into language families, based on their genetic similarity.
- The Germanic language family is further grouped into the Indo-European language family.



[en.wikipedia.org/wiki/Language_family]

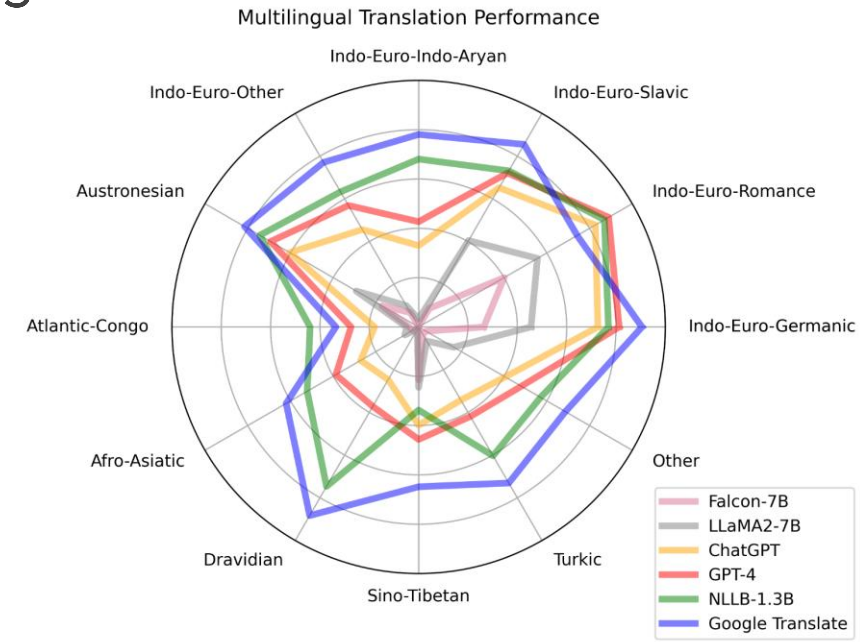
LANGUAGE FAMILIES

PIE	English	Gothic	Latin	Ancient Greek	Sanskrit	Iranian	Slavic	Baltic	Celtic	Armenian	Albanian	Tocharian	Hittite
***bʰréh$_2$tēr** "brother"[6][7][8]	**brother** (< OE *brōþor*)	**brōþar** "brother"	**frāter** "brother" ⇒ [note 4]	**pʰrā́tēr** "member of a phratry (brotherhood)" (> **phratry**)	**bʰrā́tṛ , bhrātar, bhrātā** "brother"; Rom **phral** "brother" (> **pal**)[9][10][c]	Av **brātar-**, OPers **brātar-**, NPers **brādar-**, Ossetian **ärvád** "brother, relative", NPers **barādar**, Kurd **bira/ birader**	OCS **bratrŭ** "brother"	Lith **brõlis**, OPrus **brati** "brother"	Gaul **Bratronos** (pers. name);[11] OIr **bráthair**, W **brawd** (pl. **brodyr**) "brother"	**ełbayr** (gen. **ełbawr**) "brother"		A **pracar**, B **procer** "brother"	Lyd **brafr(-sis)** "brother"[12]
***swésōr** "sister"[13][14][8]	**sister** (< OE *sweostor*, influenced by ON *systir*)	**swistar** "sister"	**soror** "sister" ⇒ [note 5]	**éor** "cousin's daughter"	**svásṛ, svasar, swasā** "sister"	Av **x̌vaŋhar-** "sister"; NPers **ẖwāhar** "sister"; Kurd **xwişk** "sister"[d]	OCS **sestra** "sister"	Lith **sesuo, seser-**, OPrus **sestra** "sister"	Gaul **suiorebe** "with two sisters" (dual) [15] OIr **siur**, W **chwaer** "sister"	**kʿuyr (kʿiṙ)**, nom.pl **kʿur-kʿ** "sister"[e]	**vashë, vajzë** "girl" (< **varjë* < **vëharë* < PAlb **swesarā*)	A **ṣar', *B* ṣer** "sister"	
***dʰugh$_2$tḗr** "daughter"[16] [17][18][19]	**daughter** (< OE *dohtor*)	**daúhtar** "daughter"	Oscan **futír** "daughter"	**θugátēr** "daughter"; Myc **tu-ka-te** "daughter"[20] [f]	**dúhitṛ, duhitar, duhitā** "daughter"	Av **dugədar-, duɣδar-**, NPers **doẖtar** "daughter" Kurd **dot** "daughter"	OCS **dŭšti, dŭšter-** "daughter"	Lith **duktė̃, dukter-**, OPrus **dukti** "daughter"	Gaulish **duxtir** "daughter"; Celtib **TuaTer (duater)** "daughter"[22] [23][24]	**dustr** "daughter"		A **ckācar**, B **tkācer** "daughter"	HLuw **túwatara** "daughter"; [25] ?Lyd **datro** "daughter"; CLuw/Hitt **duttarii̯ata-**; [g] Lyc **kbatra** "daughter"[h]

[en.wikipedia.org/wiki/Indo-European_vocabulary]

COMPARATIVE LINGUISTICS IN NLP

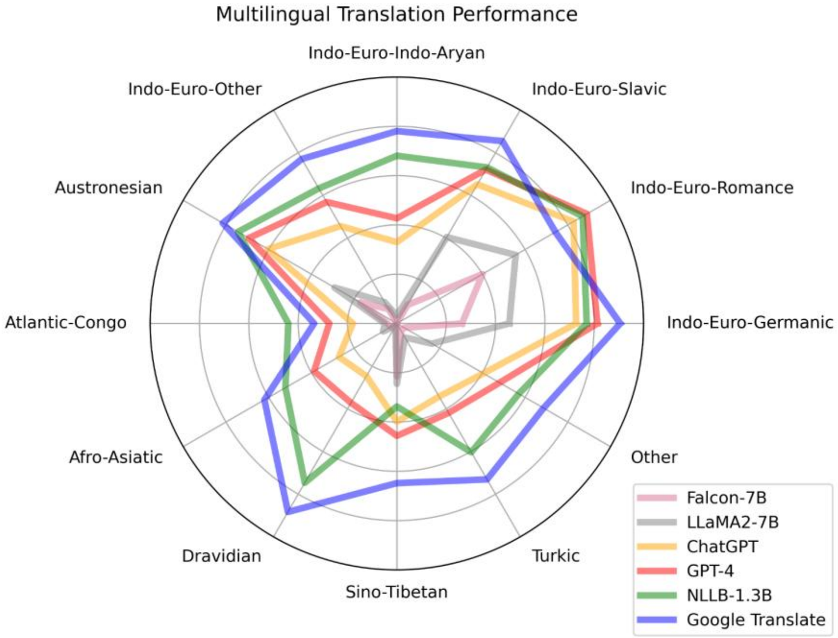
- Machine translation is easier between languages that are more closely related.
- Zhu et al. (2024) tested various LLMs on the translation task from English to various target languages.



[Zhu et al., 2024]

COMPARATIVE LINGUISTICS IN NLP

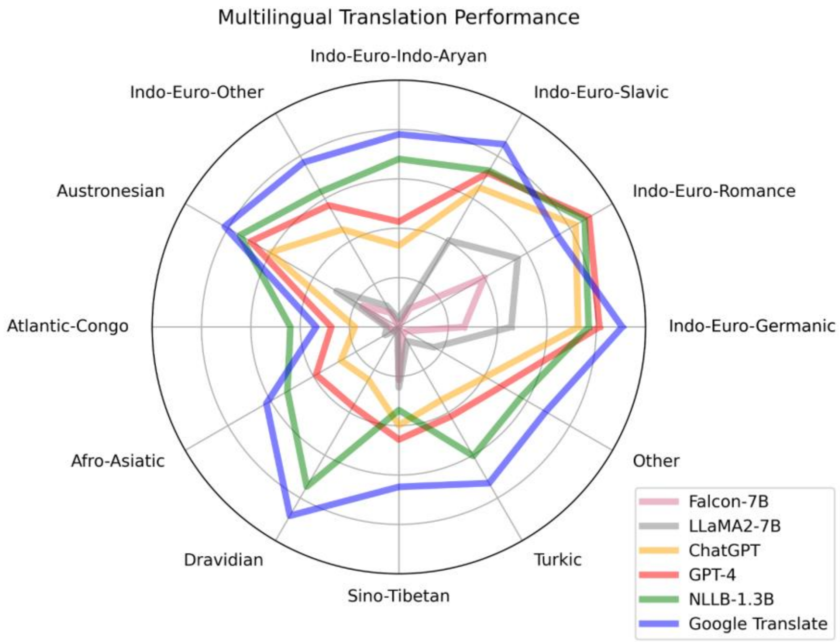
- Interestingly, while Google Translate performs best from English to other Germanic languages or to Slavic languages,
- LLMs perform better when translating to Romance languages (i.e., descendants of Latin).



[Zhu et al., 2024]

COMPARATIVE LINGUISTICS IN NLP

- All tested translation methods perform worst when translating to non-Indo-European languages, such as languages in the Atlantic-Congo family.
- But this may be due to a smaller corpus of Atlantic-Congo data.



[Zhu et al., 2024]

COMPUTATIONAL LINGUISTICS ROADMAP

- In this lecture, we discussed computational linguistics, at a high level.
- Next time: Morphology
  - What is a word?
  - How do individual words convey meaning?
- Syntax
  - How are words arranged to form sentences?
  - What is a grammar?
  - Syntactic composition
- Later: Semantics

QUESTIONS?