CS 490: NATURAL LANGUAGE PROCESSING

Dan Goldwasser, Abulhair Saparov

Lecture 10: Computational Linguistics and Morphology

COMPUTATIONAL LINGUISTICS

- In the first part of the course, we covered the fundamentals of NLP.
  - What are examples of tasks in NLP?
  - What tools are available to solve NLP tasks?
- 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 conference names such as 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.
  - (hint: The transformer was invented in 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.
- Which of these sentences are in the English language?

‘Mary caught the ball.’

‘Mary ball the caught.’

‘Colorless green ideas sleep furiously.’

‘The robb’d that smiles, steals something from the thief.’

‘They vorbled the chornis yesterday.’

‘I asked ChatGPT for the tl;dr and tweeted it.’

from Shakespeare’s Othello

Would you consider this to be a valid sentence 10 or 20 years ago?

WHAT IS LANGUAGE?

- One simple definition of a language:
  - A language is a set of “acceptable” sentences/utterances.
- A basic task in computational linguistics: language recognition.
  - Given an input string *s*, does *s* belong to the language?
- How easy is this task?
  - It depends on the language!
  - For natural languages, this can be hard.
  - What about the language of arithmetic expressions?
  - E.g., ‘`1 + 2 = 3`’, ‘`-7 * 8.4 = 49`’, etc.
- Programming languages are also languages.
  - The compiler’s job is to perform language recognition.

LANGUAGE RECOGNITION

- For languages like programming languages and arithmetic expressions, there exist algorithms that can do language recognition 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 $\forall x$`(cat(`x`) -> mammal(`x`))`.
  - We can use deduction rules to deduce `mammal(alex)`.
  - Then we use generation to convert this into ‘`Alex is a mammal.`’

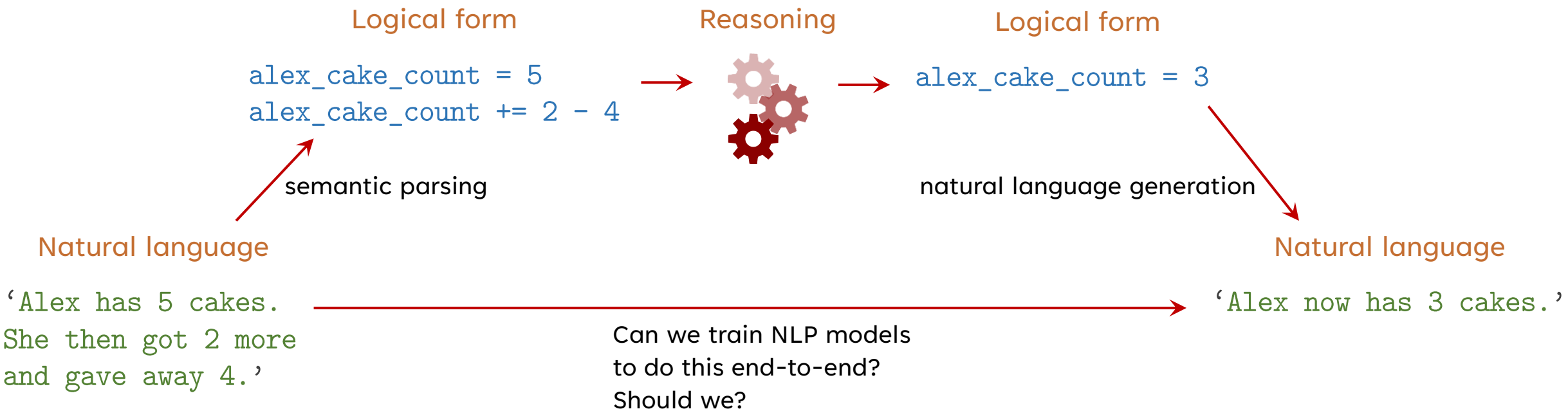
SEMANTICS AND REASONING

- 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: ???
```



COMPUTATIONAL LINGUISTICS ROADMAP

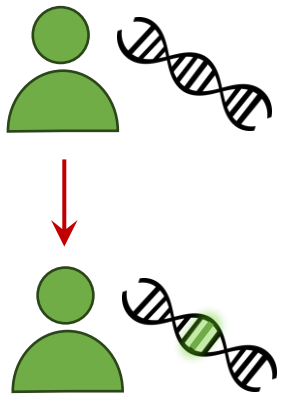
- We will cover these topics and consider possible answers to all of these research questions over the next few lectures.
- We will follow the following rough roadmap:
  - Morphology
    - What are words? How are words constructed? How do they attain meaning?
  - Syntax
    - How are words arranged to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

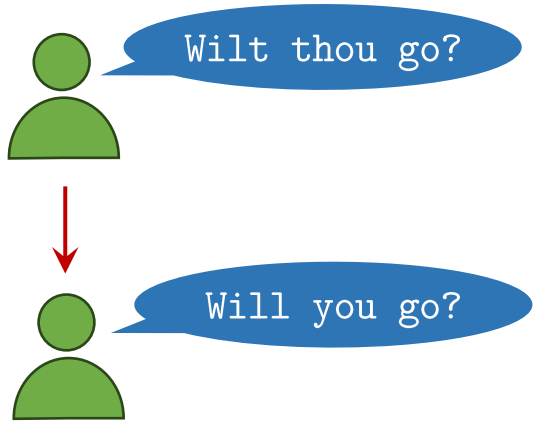
LANGUAGE IS ALWAYS CHANGING

- 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 of England.
  - 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

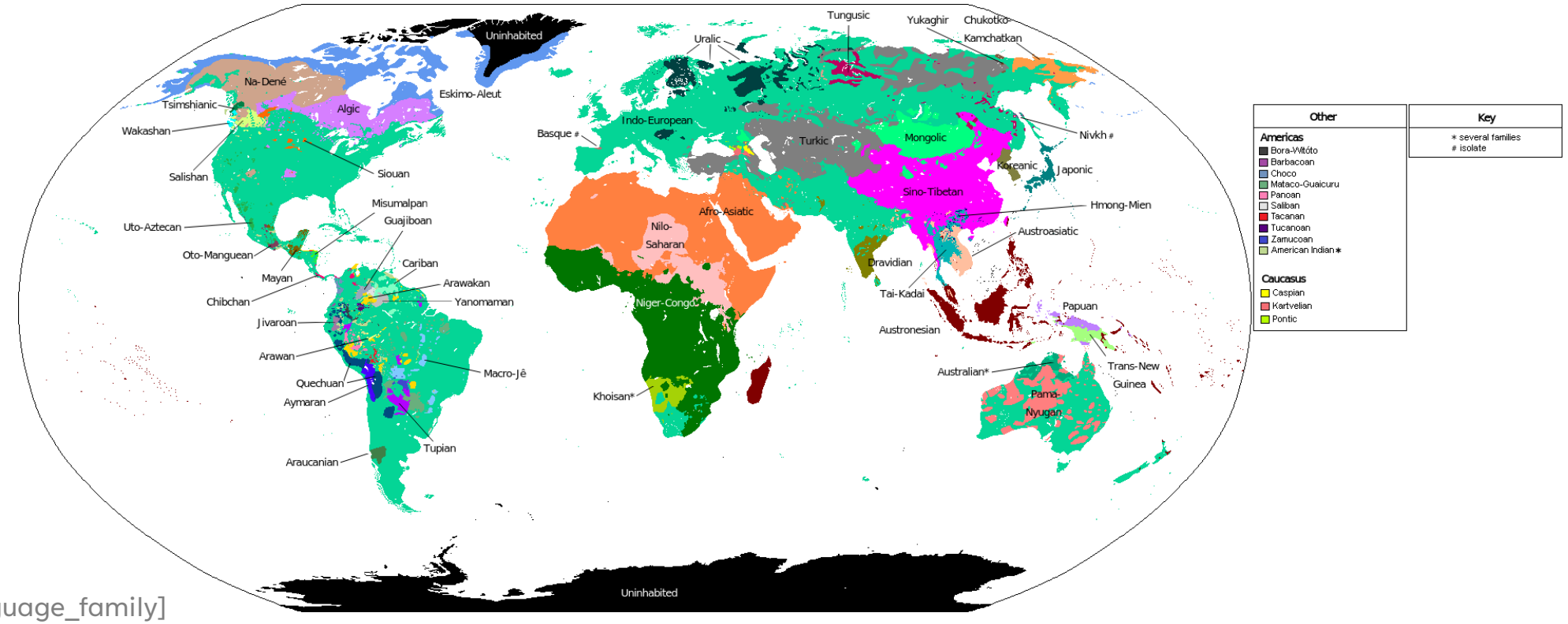
COMPARATIVE LINGUISTICS

- 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]

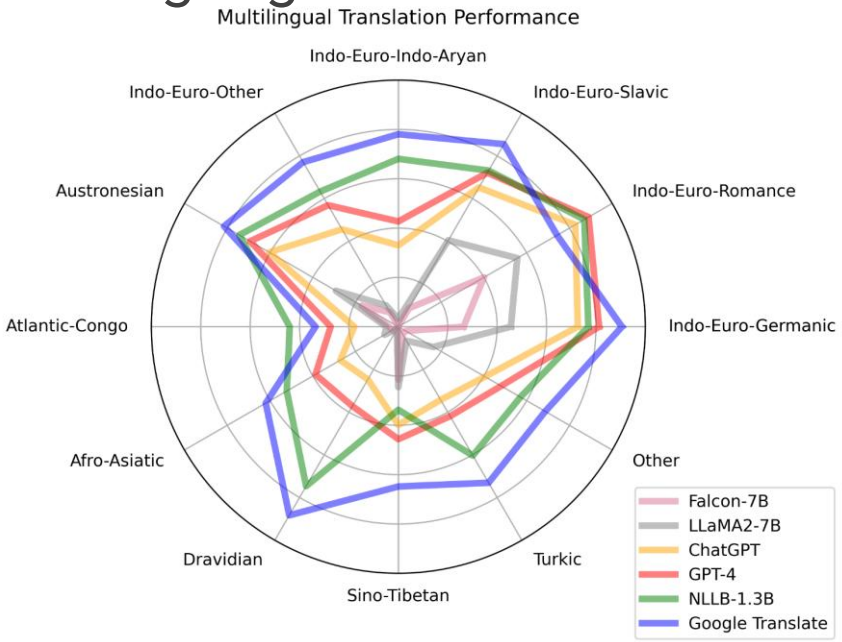
LANGUAGE FAMILIES

PIE	English	Gothic	Latin	Ancient Greek	Sanskrit	Iranian	Slavic	Baltic	Celtic	Armenian	Albanian	Tocharian	Hittite
***bʰréh₂tēr** "brother"[6][7][8]	**brother** (< OE *brōþor*)	**brōþar** "brother"	**frāter** "brother" ⇒ [note 4]	**pʰrā́tēr** "member of a phratry (brotherhood)" (> **phratry**)	**bʰrā́tr̥ , 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"	**svásr̥, svasar, swasā** "sister"	Av **x̌vaŋhar-** "sister"; NPers **ḫwāhar** "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 (kʿiŕ)**, nom.pl **kʿur-kʿ** "sister"[e]	**vashë**, **vajzë** "girl" (< **varjë* < **vëharë* < PAlb **swesarā*)	A **ṣar'**, *B* **ṣer** "sister"	
***dʰugh₂tḗr** "daughter"[16] [17][18][19]	**daughter** (< OE *dohtor*)	**daúhtar** "daughter"	Oscan **futír** "daughter"	**θugátēr** "daughter"; Myc **tu-ka-te** "daughter"[20] [f]	**dúhitr̥, duhitar, duhitā** "daughter"	Av **dugədar-, duγðar-**, NPers **doḫtar** "daughter" Kurd **dot** "daughter"	OCS **dŭšti, dŭšter-** "daughter"	Lith **duktė, 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 **duttariia-**; [g] Lyc **kbatra** "daughter"[h]

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

COMPARATIVE LINGUISTICS IN NLP

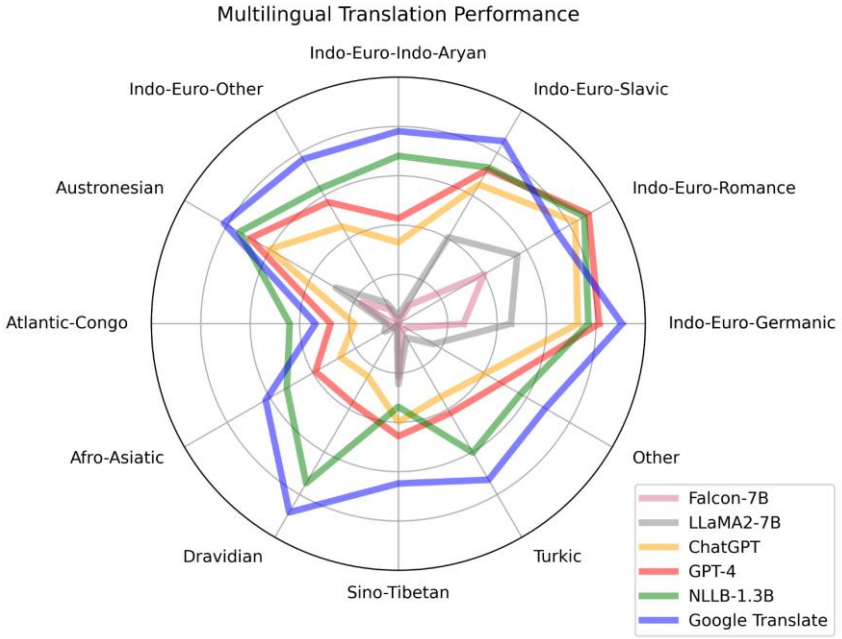
- Machine translation is easier between languages that are more closely related.
- Zhu et al. (2024) tested various NLP models on the translation task from English to various target languages.



[Zhu et al., 2024]

COMPARATIVE LINGUISTICS IN NLP

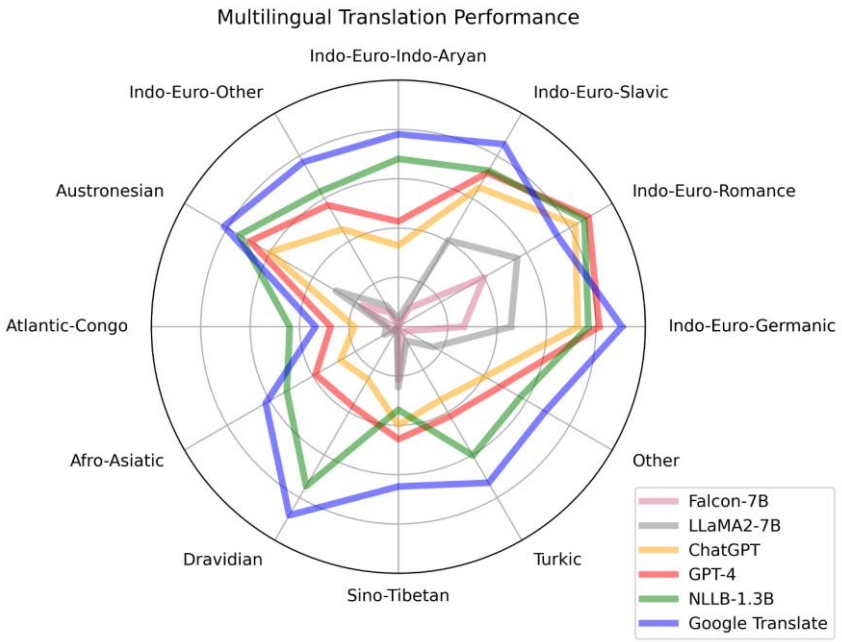
- Interestingly, while Google Translate performs best from English to other Germanic languages or to Slavic languages,
- Other models perform better when translating to Romance languages (i.e., descendants of Latin).



[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]

MORPHOLOGY

MORPHOLOGY

- Morphology is the study of the internal structure of words.
- Languages often have ways to create new words from existing words.
  - E.g., ‘fortunate’ -> ‘unfortunate’
  - ‘unfortunate’ -> ‘unfortunately’
- Many languages have inflection.
  - These are word markings that reflect the syntactic context of the word.
  - E.g., ‘A cat sleeps on the couch’ vs ‘Cats sleep on the couch’.
  - ‘A cat grooms’ vs ‘A cat groomed’.
- Many languages have compound words.
  - E.g., ‘skyscraper’, ‘tablecloth’, ‘subway’, ‘doomscroll’, etc.

MORPHOLOGY

- Words can be broken down into a root and a collection of affixes.
  - E.g., ‘unfortunately’
    - Root: ‘fortunate’
    - Affixes: ‘un-’ [negation prefix], ‘-ly’ [adverb suffix]
  - E.g., ‘runs’
    - Root: ‘run’
    - Affixes: ‘-s’ [present tense, singular, 3rd person]
  - E.g., ‘am’
    - Root: ‘be’
    - Affixes: ‘am’ [present tense, singular, 1st person]
- This task is called morphological parsing.

MORPHOLOGY AND TOKENIZATION

- How is morphology relevant to NLP models?
  - How should NLP models tokenize text?
- Suppose a tokenizer splits text into words, but does not split words into sub-word components.
  - The model would see ‘`fortunate`’ and ‘`unfortunate`’ as two separate entities.
    - The model would need to learn the meaning of each word independently.
  - Such a model would never learn the general meaning of the prefix ‘`un-`’,
  - Or the meaning of any other sub-word component.

MORPHOLOGY AND TOKENIZATION

- Imagine a model with unbounded number of parameters and training data.
  - But the tokenizer does not split words into sub-word components.
- Such a model would not generalize well to unseen words.

  (poor out-of-distribution generalization)
  - E.g., consider the word ‘`undivide`’.
  - This is not a real English word, but we can easily guess its meaning.

  (something like “to combine” or “to recombine”)
- Similarly, arbitrarily splitting words into sub-word components would lead to similar problems.
  - E.g., Tokenizing ‘`unfortunate`’ into [‘`unfor`’,‘`tunate`’] will not help the model to learn the meaning of ‘`un-`’.

MORPHOLOGY AND TOKENIZATION

- Why not tokenize at the character-level?
  - This increases the computation cost of NLP models.
  - For example, consider autoregressive LMs:
    - More forward passes are needed if every token is a single character.
- The model must learn more relationships between tokens.
  - It must learn that the sequence `['i','n','g']` has the meaning of a continuous action,
  - E.g., 'playing',
  - As opposed to a single token if the word was tokenized as `['play','ing']`.

SUBWORD TOKENIZATION

- How do we tokenize at the right level of granularity?
  - One approach is to train a tokenizer from data.
  - This is the approach taken by byte pair encoding (BPE; Sennrich et al., 2016).

- In BPE, we start with a vocabulary containing all individual characters.

  Σ = {'A','B','C',…,'Y','Z','a','b','c',…,'y','z','0','1',…}

- Then repeat k times:
  - Choose the two symbols in Σ that occur most frequently together in the training corpus (e.g., 'u' and 'n').
  - Add a new merged symbol (e.g, 'un') to Σ.
  - Replace all adjacent 'u' and 'n' in the training corpus with 'un'.

BYTE PAIR ENCODING

- Some preprocessing is typically done with BPE:
  - The corpus is split into words by spaces.
  - The space is added to the end of each word as a special token,
    - E.g., `'the stars shone' -> ['the_', 'stars_', 'shone']`.
- Once we have a learned vocabulary, we can tokenize any new text:
  - Perform each merge operation in the same order that it was learned during training.
- BPE is used in all major LLMs.

WORDPIECE TOKENIZATION

- An alternative subword tokenization method is WordPiece tokenization.
- It is largely identical to BPE, with the core difference being the merge rule:
- In BPE, at each iteration, the two symbols that most frequently appear together in the training corpus are merged.
- In WordPiece, we instead select the two symbols a and b that maximize the quantity:

$$\frac{\texttt{frequency}(ab)}{\texttt{frequency}(a)\cdot\texttt{frequency}(b)}.$$

WORDPIECE TOKENIZATION

- Once trained, the WordPiece tokenizer also differs slightly from the trained BPE tokenizer.
- Instead of applying merge operations in the same order in which they were learned,
- The WordPiece tokenizer simply matches tokens greedily.
  - Starting from the beginning of the sequence, it finds the longest subword in its vocabulary that matches the input.
  - Then it repeats.

EFFECT OF TOKENIZATION

- Toraman et al. (2022) empirically tested the effect of different tokenizers on the performance of medium-sized RoBERTa models.
  - They focused on Turkish, which is very morphologically rich (i.e., words often contain many affixes).

		News Classification			Hate Speech Detection			Sentiment Analysis			Named Entity Recognition			Semantic Text Similarity		Natural Language Inference		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	corr	p-value	P	R	F1
	BERT	0.918	0.917	0.917	0.781	0.781	0.781	0.927	0.927	0.927	0.935	0.955	0.945	0.862	<1e-178	0.852	0.852	0.852
R-TR-medium	Char	0.715	0.723	0.713	0.606	0.609	0.607	0.812	0.812	0.812	0.730	0.788	0.757	0.256	<1e-4	0.620	0.619	0.619
	BPE	**0.886**	**0.885**	**0.885** ●	0.742	0.737	0.738	0.882	0.881	0.881 ○	0.851	0.883	0.866 ○	0.487	<2e-32	0.772	0.772	0.772
	WP	0.882	0.881	0.881 ○	**0.745**	**0.745**	**0.745** ●	**0.884**	**0.884**	**0.884** ●	**0.858**	**0.893**	**0.875** ●	**0.718**	<3e-92 ●	**0.778**	**0.778**	**0.778** ●
	Morph	0.869	0.868	0.867	0.726	0.727	0.726	0.824	0.823	0.823	0.839	0.872	0.855	0.655	<5e-63 ○	0.768	0.768	0.768
	Word	0.857	0.857	0.856	0.647	0.649	0.648	0.805	0.805	0.805	0.791	0.740	0.764	0.492	<2e-16	0.603	0.598	0.595

WHAT ABOUT OTHER LANGUAGES?

- Not all languages have morphologies similar to English.
- Some languages have little to no morphology.
  - These are called isolating or analytic languages.
- E.g., Yoruba, Vietnamese
- E.g., Chinese
  - There are some examples of inflection (e.g., '们' or 'mén' can denote plural),
  - As well as some examples of derivation (e.g., '艺术家' or 'yìshùjiā' which means 'artist').
  - But these are rare compared to other languages.
- Chinese contains a significant number of compound words (~80% of Chinese words are compounds).

WHAT ABOUT OTHER LANGUAGES?

- In fusional languages, each affix can encode information about multiple grammatical features.
- English has some examples of “fusion” but not as much as other languages (Modern English has become more analytic relative to earlier forms).
  - E.g., ‘`-es`’ in ‘`crosses`’ denotes 3rd person, singular, and present tense.
- E.g., Most Indo-European languages: French, Spanish, Italian, Greek, Irish, Polish, Russian, Ukranian, Lithuanian, etc.

  (proto-Indo-European was most likely a fusional language)

FUSIONAL MORPHOLOGY IN SPANISH

		singular			plural		
		1st person	2nd person	3rd person	1st person	2nd person	3rd person
indicative		yo	tú vos	él/ella/ello usted	nosotros nosotras	vosotros vosotras	ellos/ellas ustedes
	present	corro	corres[tú] corrés[vos]	corre	corremos	corréis	corren
	imperfect	corría	corrías	corría	corríamos	corríais	corrían
	preterite	corrí	corriste	corrió	corrimos	corristeis	corrieron
	future	correré	correrás	correrá	correremos	correréis	correrán
	conditional	correría	correrías	correría	correríamos	correríais	correrían
subjunctive		yo	tú vos	él/ella/ello usted	nosotros nosotras	vosotros vosotras	ellos/ellas ustedes
	present	corra	corras[tú] corrás[vos][2]	corra	corramos	corráis	corran
	imperfect (ra)	corriera	corrieras	corriera	corriéramos	corrierais	corrieran
	imperfect (se)	corriese	corrieses	corriese	corriésemos	corrieseis	corriesen
	future[1]	corriere	corrieres	corriere	corriéremos	corriereis	corrieren
imperative		—	tú vos	usted	nosotros nosotras	vosotros vosotras	ustedes
	affirmative		corre[tú] corré[vos]	corra	corramos	corred	corran
	negative		no corras	no corra	no corramos	no corráis	no corran

[https://en.wiktionary.org/wiki/correr#Spanish]

TEMPLATE-BASED FUSIONAL MORPHOLOGY

- Affixes are not always added to the beginning or ends of roots.
- In Semitic languages (i.e., Akkadian, Arabic, Aramaic, Hebrew, Phoenician), the roots of words are three consonants.
  - Triconsonantal roots
- Words can be constructed by adding different vowels between the consonants.
  - **k**a**t**a**b**a كَتَبَ or كتب “he wrote”
  - **k**a**t**a**b**at كَتَبَت or كتبت “she wrote”
  - **k**i**t**ā**b** كِتَاب or كتاب “book”
  - ma**k**ta**b** مَكتَب or مكتب “desk” or “office”
  - ma**k**ta**b**at مَكتَبة or مكتبة “library” or “bookshop”

AGGLUTINATIVE MORPHOLOGY

- In some languages, each affix encodes information about a single grammatical feature.
  - Multiple affixes can be chained together in a linear and systematic fashion.
- This is called agglutination.
- Examples of agglutinative languages: Finnish, Japanese, Korean, Swahili.

AGGLUTINATIVE MORPHOLOGY

- An extreme example of agglutination from Turkish:

uygarlaştıramadıklarımızdanmışsınızcasına

"(behaving) as if you are among those whom we were not able to civilize"

uygar	"civilized"
+laş	"become"
+tır	"cause to"
+ama	"not able"
+dık	past participle
+lar	plural
+ımız	first person plural possessive ("our")
+dan	ablative case ("from/among")
+mış	past
+sınız	second person plural ("y' all")
+casına	finite verb → adverb ("as if")

[Olfazer and Güzey, 1994]

POLYSYNTHETIC MORPHOLOGY

- Some languages can utilize morphology to encode the meaning of full sentences.
- E.g., many (but not all) Native American languages, Ainu, Mayan, Quechua.
  - E.g., in Yupik: ‘tuntussuqatarniksaitengqiggtuq’

tuntu	-ssur	-qatar	-ni	-ksaite	-ngqiggte	-uq
“reindeer”	“hunt”	[future]	“say”	[negative]	“again”	[3rd person, singular, indicative]

  - Means: “He had not yet said again that he was going to hunt reindeer.”
  - Only ‘tuntu’ can stand alone as a word.
- Verbs can be attached to nouns as affixes.

COMPUTATIONAL LINGUISTICS ROADMAP

- In this lecture, we discussed computational linguistics, at a high level.
- No need to memorize different morphologies of different languages.
  - Key takeaway:
  - Be aware of the diversity of morphologies of languages around the world.
  - And how they may affect NLP methods.
  - Certain NLP methods may work better for some languages than others.
- Next time: Syntax
  - How are words arranged to form sentences?
  - What is a grammar?
  - Syntactic composition
- Later: Semantics

QUESTIONS?