CS 577: NATURAL LANGUAGE PROCESSING

Abulhair Saparov

Lecture 20: Morphology and Syntax

COMPUTATIONAL LINGUISTICS ROADMAP

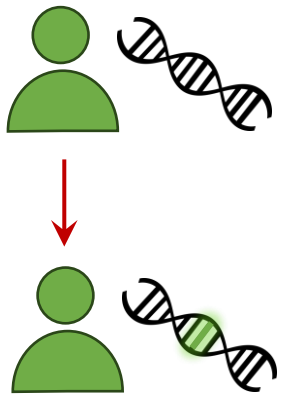
- Last class, we discussed computational linguistics at a high-level.
  - What is a language?
  - High-level models of natural language understanding.
- Today we will discuss:
  - Diversity in natural languages and how they change over time.
  - Morphology
    - What are words? How are words constructed? How do they attain meaning?
  - Syntax
    - How are words arranged to form sentences? What is a grammar?

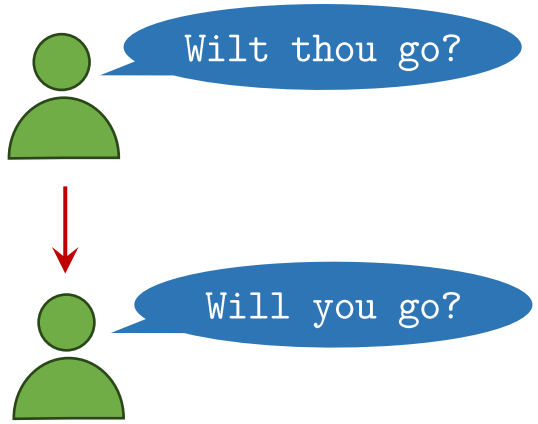
LANGUAGE IS ALWAYS CHANGING

- 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	*aplI*[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

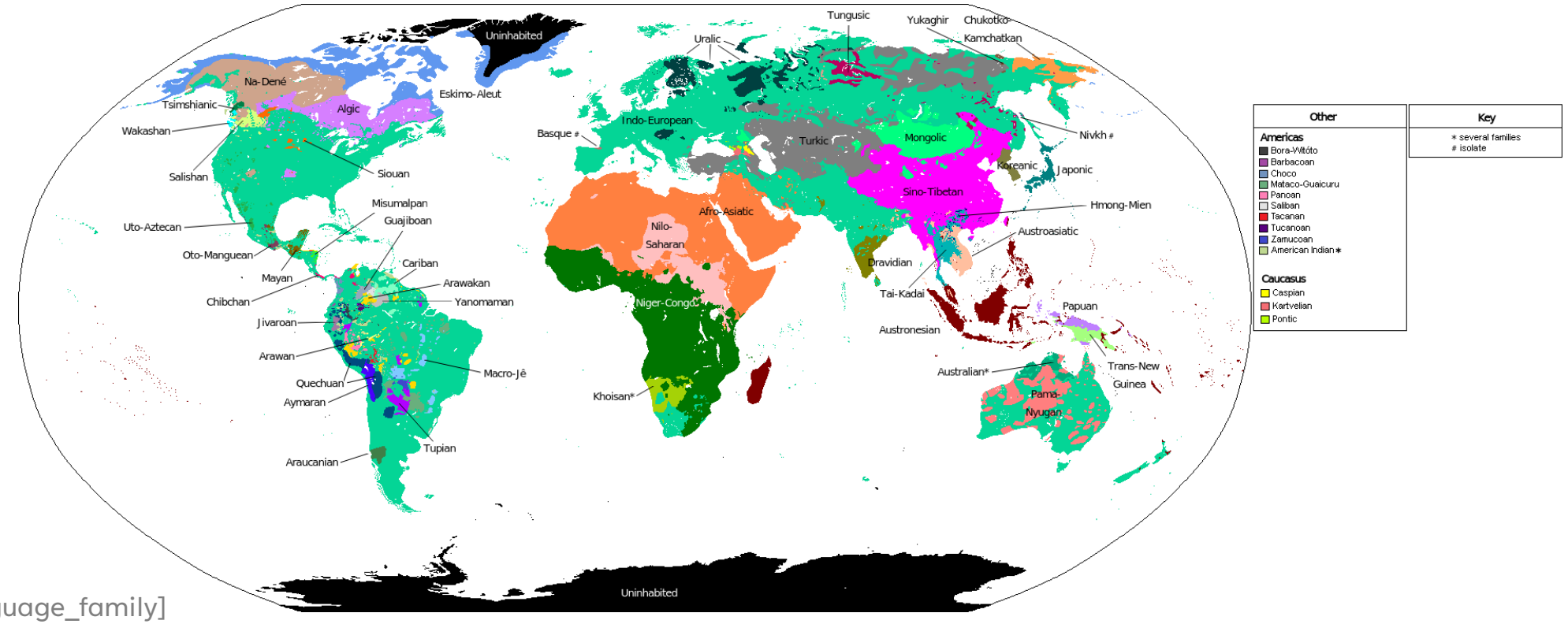
COMPARATIVE LINGUISTICS

- 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]

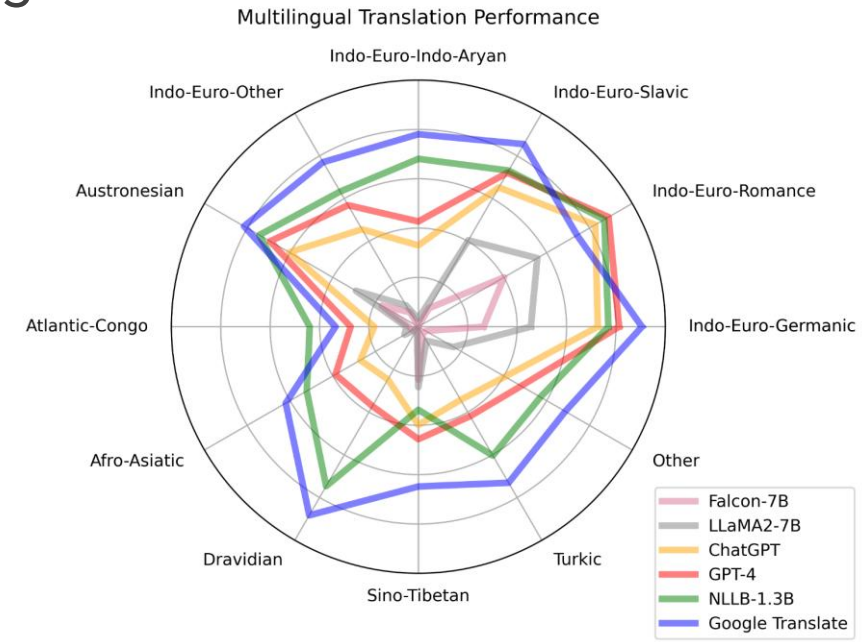
LANGUAGE FAMILIES

PIE	English	Gothic	Latin	Ancient Greek	Sanskrit	Iranian	Slavic	Baltic	Celtic	Armenian	Albanian	Tocharian	Hittite
***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 **TuaƮer** (**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

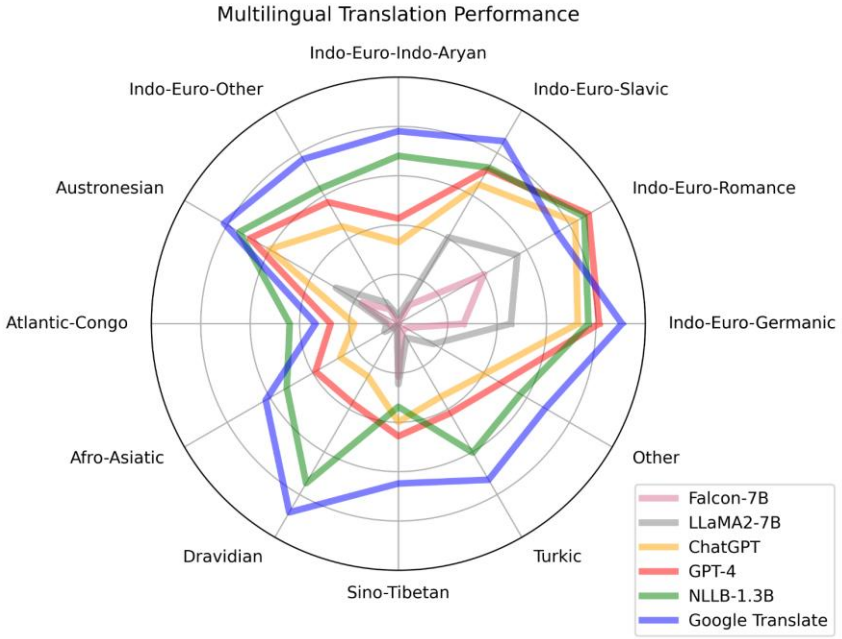
- Machine translation is easier between languages that are more closely related.
- Zhu et al. (2024) tested various LLMs on the translation task from English to various target languages.



[Zhu et al., 2024]

COMPARATIVE LINGUISTICS IN NLP

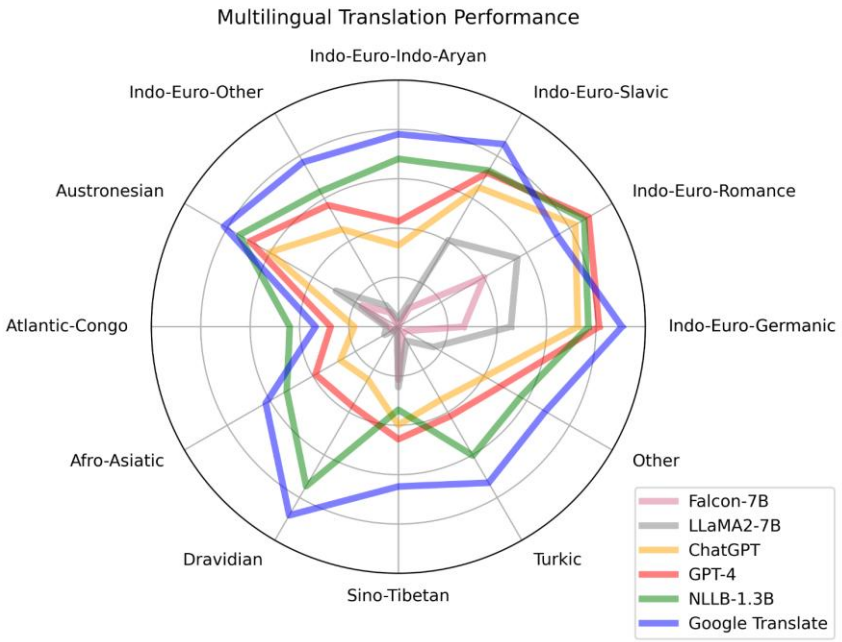
- Interestingly, while Google Translate performs best from English to other Germanic languages or to Slavic languages,
- LLMs perform better when translating to Romance languages (i.e., descendants of Latin).



[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]

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”
  - m**a**k**t**a**b** مَكتَب or مكتب “desk” or “office”
  - m**a**k**t**a**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.

SYNTAX

SYNTAX

- Consider the sentence ‘Sam ran to the bank.’
- The order of the words has a significant effect on the grammaticality and meaning of the sentence.
- Some orderings produce ungrammatical sentences: ‘Ran to Sam the bank.’
- Other orderings change the meaning/semantics: ‘The bank ran to Sam.’
- Other orderings produce grammatical sentences that preserve the meaning:
  - ‘To the bank, ran Sam.’
  - ‘To the bank, Sam ran.’
  - ‘Sam, to the bank, ran.’
  - Notice these sentences maintain three contiguous phrases: ‘Sam’, ‘ran’, ‘to the bank’.

SYNTAX

- Another way to understand syntax is to consider substitutions of words and/or phrases.
  - What can we substitute for ‘Sam’ while preserving grammaticality?
  - ‘A cat ran to the bank’ (grammatical)
  - ‘Fast ran to the bank’ (ungrammatical, unless ‘Fast’ is a name)
  - The words ‘Sam’ and ‘a cat’ belong to the same grammatical category.
    - They are nouns.
  - But ‘fast’ is in a different grammatical category.
    - An adjective or adverb.

PARTS OF SPEECH

- These grammatical categories are called parts of speech (POS).
  - Nouns: ‘cat’, ‘bank’, ‘Sam’
  - Verb: ‘run’, ‘sleep’, ‘is’
  - Adjectives: ‘fast’, ‘orange’, ‘short’
  - Adverbs: ‘quickly’, ‘probably’, ‘slowly’
  - Prepositions: ‘to’, ‘in’, ‘of’
  - Determiners: ‘a’, ‘the’, ‘those’
  - Pronouns: ‘I’, ‘me’, ‘mine’
- Some words can have multiple parts of speech:
  - ‘The fast car drove by.’
  - ‘The car drove by fast.’

PARTS OF SPEECH

- Parts of speech can vary across languages.
  - Many East Asian Languages have measure words or classifiers.
  - E.g., ‘버스 티켓 열 장’

    "bus" "ticket" "10" [measure word for sheet-like objects]
  - ‘피자 한 조각’

    "pizza" "1" [measure word for slices]
- Some languages use postpositions:
  - Similar to prepositions, except the postposition occurs after the noun phrase.
  - E.g., in Hungarian: ‘`fa alatt`’ is literally ‘tree under’ but means ‘under the tree’.

PARTS OF SPEECH

- Some parts of speech are closed classes; new words are rarely added.
  - Determiners, prepositions, postpositions, pronouns, measure words
- Others are open classes; and new words are added readily.
  - Nouns, verbs, adjectives, adverbs
- Closed classes do change, but typically over much longer time periods.
  - E.g., the preposition ‘during’ comes from the verb ‘dure’ (‘dure’+‘ing’).
    - Probably before or around the time of Middle English.
  - The verb ‘dure’ means to continue, to endure, to last.

SYNTAX

- We can also substitute words with phrases:
  - ‘`The fast cat with the stripes ran to the bank`’ (grammatical)
  - ‘`The fast cat with the stripes`’ is a phrase that behaves like a noun.
    - This is a noun phrase.
- We can similarly define other kinds of phrases:
  - Verb phrase: ‘`ran to the bank`’
  - Adjective phrase: ‘`taller than a mountain`’
  - Adverbial phrase: ‘`very quickly`’
  - Prepositional phrase: ‘`to the bank`’

SYNTAX

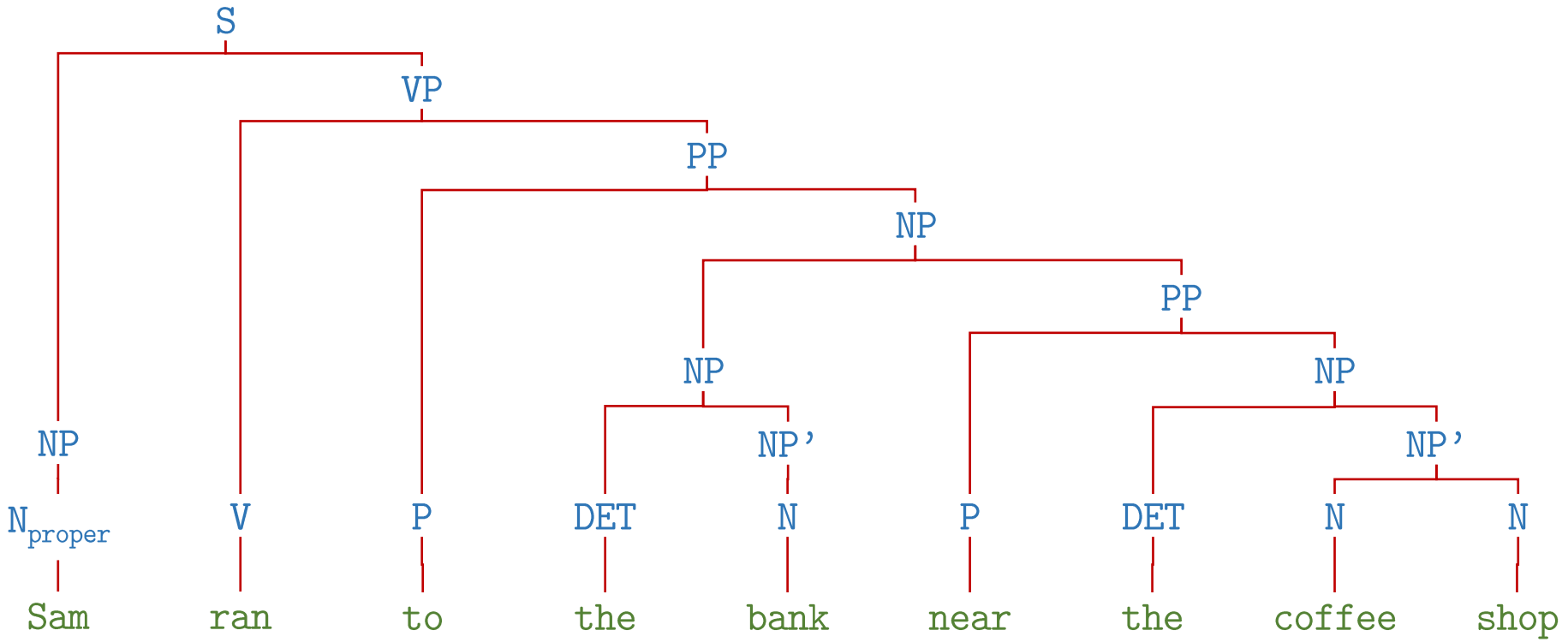
- Focusing on the phrase ‘to the bank’, each component is necessary to maintain the grammaticality and meaning of the sentence.
  - ‘Sam ran bank’ (ungrammatical)
  - ‘Sam ran to bank’ (ungrammatical)
  - ‘Sam ran the bank’ (grammatical, but different meaning)
  - All three words are needed to form the phrase ‘to the bank.’
- But looking closely at the prepositional phrase, we see that it is composed of smaller components:
  - A preposition (‘to’) and a noun phrase (‘the bank’).
  - We can conclude that one way to construct prepositional phrases is to combine any preposition with a noun phrase.

SYNTAX

- For example, we can rephrase the sentence as ‘Sam ran to Chase Bank.’
- So ‘Chase Bank’ and ‘the bank’ play the same grammatical role.
  - They are noun phrases.
- Language enables to construction of larger noun phrases:
  - E.g., ‘Sam ran to the bank near the coffee shop.’
  - Here, we made a larger noun phrase by adding the prepositional phrase “near the coffee shop.”
- So we have seen that prepositional phrases can contain subordinate noun phrases, and noun phrases can contain subordinate prepositional phrases.
  - This is an example of recursion,
  - Recursion is a key property of languages, including natural language.

SYNTAX TREE

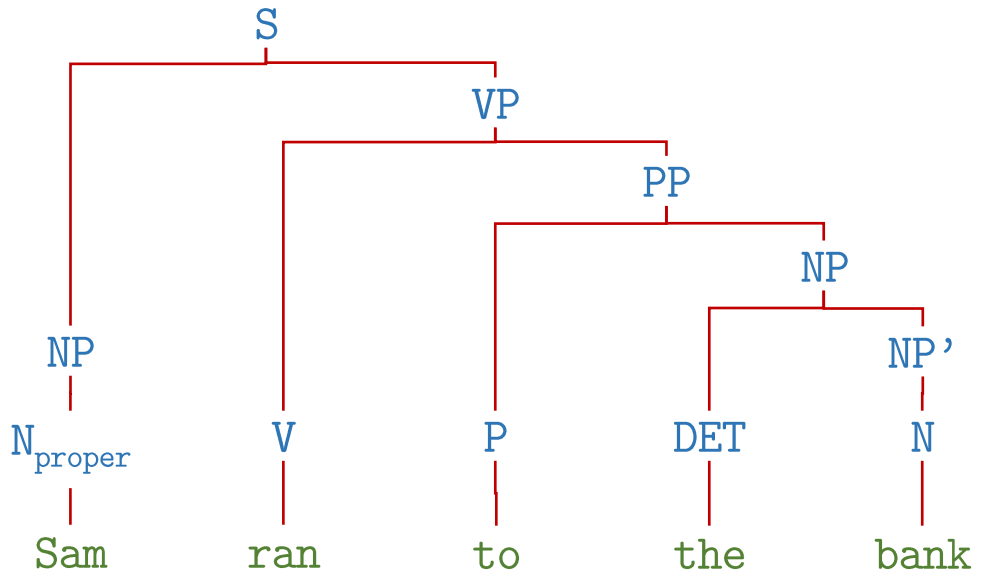
- A natural way to represent recursive structure is as a tree:



SYNTAX TREE

- Syntax trees can also be represented in bracketed notation:

```
(S
    (NP (N_proper 'Sam'))
    (VP
        (V 'ran')
        (PP
            (P 'to')
            (NP
                (DET 'the')
                (NP' (N 'bank'))
            )
        )
    )
)
```



GRAMMAR

- We have observed that noun phrases can be constructed from smaller phrases:
  - Either a proper noun (e.g., ‘Sam’),
  - A determined common noun (e.g., ‘the bank’),
  - Or a noun phrase + prepositional phrase (e.g. ‘the bank near me’),
  - (or other rules)
- Similarly, we have observed prepositional phrases can be constructed from smaller phrases.
  - And similarly for verb phrases, etc.
- These are grammatical rules.
  - Altogether, these rules form the grammar of a language.

GRAMMAR

- We can write the rules of a grammar:
- Each rule is called a production rule.
- S is the root.

```
S  -> NP VP
VP ->  V NP
VP ->  V PP
PP ->  P NP
NP -> NP PP
NP ->  N_proper
NP -> DET NP'
NP' -> N
```

```
V -> 'ran'
V -> 'sees'
V -> 'see'
P -> 'to'
P -> 'near'
N -> 'bank'
N -> 'coffee'
N -> 'shop'
```

```
N_proper -> 'Ada'
N_proper -> 'Alex'
N_proper -> 'Sam'
N_proper -> 'Zach'
DET -> 'the'
DET -> 'a'
```

GRAMMAR

- Nonterminals: S, NP, VP, PP, NP', DET, N_{proper}, N, V, P
- Terminals: 'ran', 'sees', 'see', 'to', …, 'Ada', 'Alex', …
- Preterminals: DET, N_{proper}, N, V, P (this is a subset of nonterminals)

```
S   -> NP VP
VP  ->  V NP
VP  ->  V PP
PP  ->  P NP
NP  -> NP PP
NP  ->  N_proper
NP  -> DET NP'
NP' -> N
```

```
V -> 'ran'
V -> 'sees'
V -> 'see'
P -> 'to'
P -> 'near'
N -> 'bank'
N -> 'coffee'
N -> 'shop'
```

```
N_proper -> 'Ada'
N_proper -> 'Alex'
N_proper -> 'Sam'
N_proper -> 'Zach'
DET -> 'the'
DET -> 'a'
```

GRAMMAR

- This grammar specifies a set of strings (i.e., a language).
- To sample a string from the grammar, start with the root S.
  - Recursively: For any nonterminal in the input, pick a rule with a matching left-hand side. Rewrite the nonterminal with the right-hand side.

```
S  -> NP VP
VP ->  V NP
VP ->  V PP
PP ->  P NP
NP -> NP PP
NP ->  N_proper
NP -> DET NP'
NP' -> N
```

```
V -> 'ran'
V -> 'sees'
V -> 'see'
P -> 'to'
P -> 'near'
N -> 'bank'
N -> 'coffee'
N -> 'shop'
```

```
N_proper -> 'Ada'
N_proper -> 'Alex'
N_proper -> 'Sam'
N_proper -> 'Zach'
DET -> 'the'
DET -> 'a'
```

GRAMMAR

- This grammar specifies a set of strings (i.e., a language).
- To sample a string from the grammar, start with the root S.
  - Recursively: For any nonterminal in the input, pick a rule with a matching left-hand side. Rewrite the nonterminal with the right-hand side.
- Example:
  1. S
  2. NP VP
  3. N_{proper} VP
  4. Sam VP
  5. Sam V NP
  6. Sam sees NP
  7. Sam sees N_{proper}
  8. Sam sees Ada

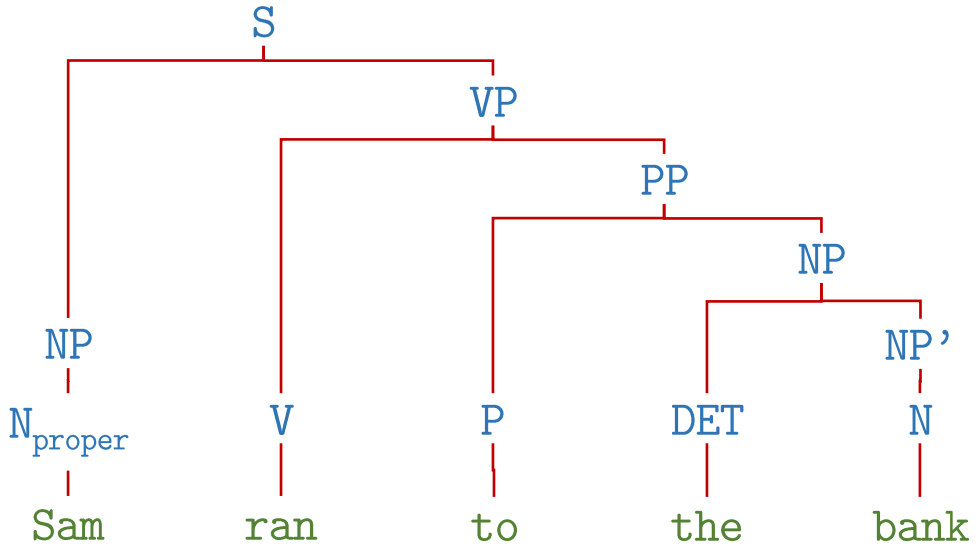
GRAMMAR

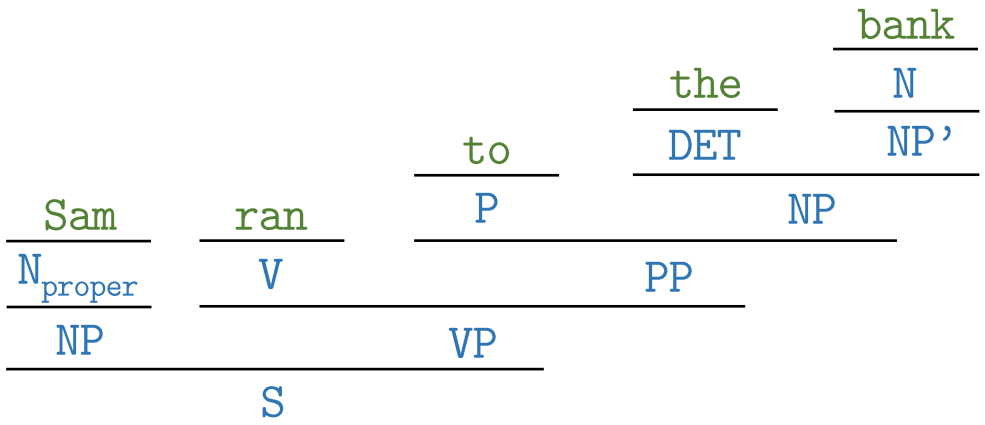
- This grammar specifies a set of strings (i.e., a language).
- To sample a string from the grammar, start with the root S.
  - Recursively: For any nonterminal in the input, pick a rule with a matching left-hand side. Rewrite the nonterminal with the right-hand side.
- The language generated by a grammar is the set of all strings s that can be generated with the above procedure.
- In this example grammar, the generated language is infinite:

```
L = {'Sam sees Ada', 'Sam see Ada', 'Alex sees the coffee', 'Zach ran the shop',
     'Sam ran to the bank', 'Sam ran to the bank near the shop',
     'Sam ran to the bank near the shop near Ada', …}
```

GRAMMAR

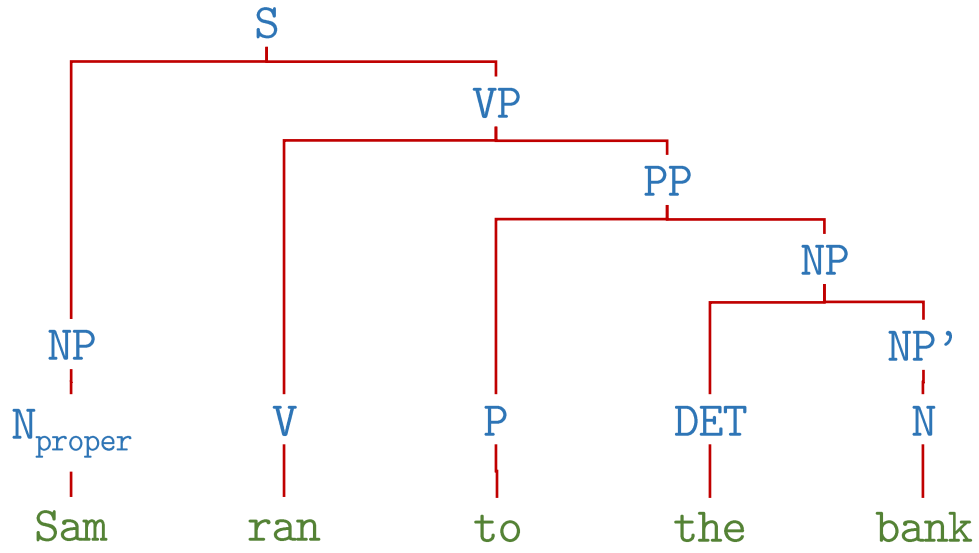
- Another way to think about grammars is as a deductive process.
- Each production rule can be interpreted as a deduction rule.
- E.g., from ‘the’, we can deduce DET from the rule DET -> ‘the’.
- From NP and VP, we can deduce S from the rule S -> NP VP.

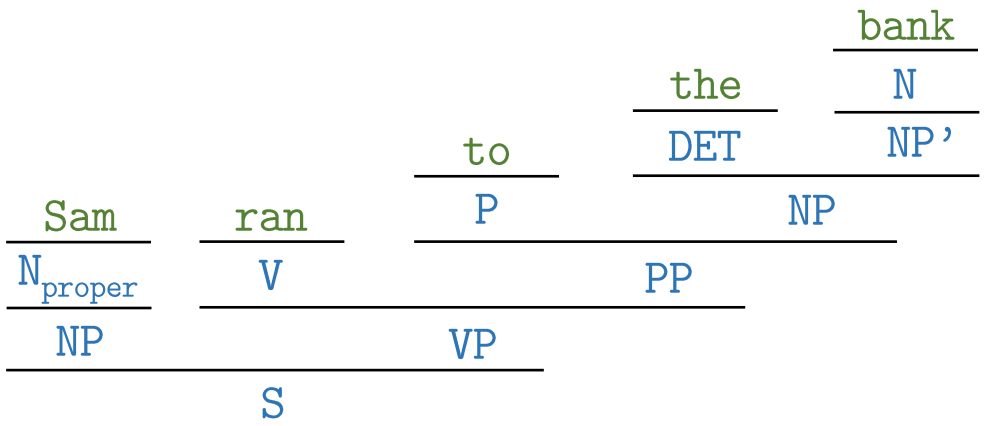




GRAMMAR

- The syntax tree is effectively a proof tree:
  - We have "proven" S from 'Sam ran to the bank'.
- Thus, syntax trees are also interchangeably called derivation trees.
- The task of parsing is equivalent to deriving/proving S from an input string.





GRAMMAR

- It is easier to parse some grammars than others.
- Consider the language containing strings with only 1's.
  - It's easy to recognize strings in this language:
  - Just check if the input string contains only 1's.
- Can we write this as a grammar?
- Note the two grammars below generate the same set of strings.
  - Thus, they are called weakly equivalent.

```
S -> '1' S
S -> '1'
```

or

```
S -> S '1'
S -> '1'
```

NEXT TIME

- Next lecture, we will discuss:
  - What are some classes of grammars that are easier to parse?
  - What kinds of grammars are more difficult to parse?
  - Where does natural language fit in?
- What kinds of grammars can be learned by NLP models?
  - Transformers?

QUESTIONS?