Overview of Linguistics

CS 759/859 Natural Language Processing Lecture 2

Samuel Carton, University of New Hampshire

Additional intro information

Google Colab

- Must make a Google account
- Free access to GPUs, but occasionally at-capacity
- May need to pay $10 for a month or two of Colab Pro
  https://colab.research.google.com/signup

Textbook

- *Speech and Language Processing, 3rd ed*. Jurafsky & Martin (2025)
- https://web.stanford.edu/~jurafsky/slp3/
- I don't teach from it, but it's a helpful reference

This lecture

Brief introduction to linguistics, focusing on morphology, lexemes and syntax.

Content largely borrowed from http://demo.clab.cs.cmu.edu/NLP/ Lec. 4, 7, 12

(Sadly now disappeared)

For more information:

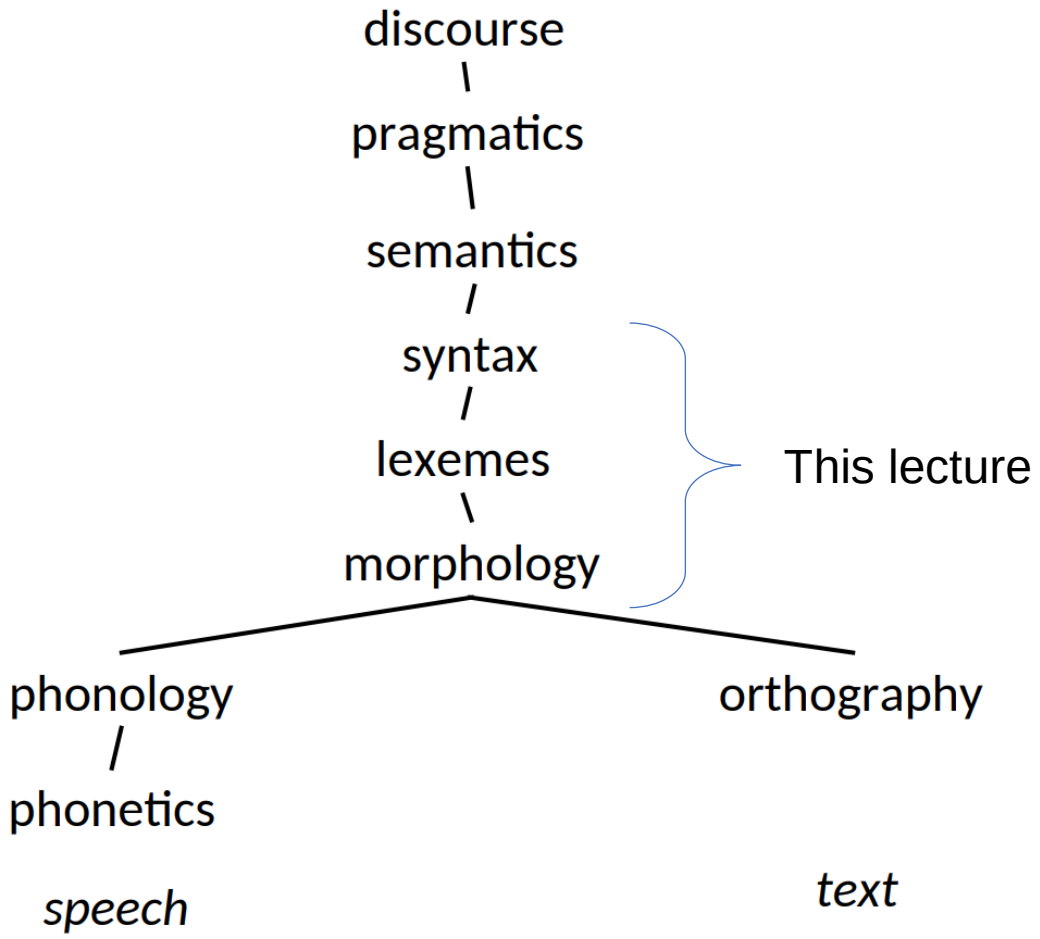
Speech and Language Processing (2nd edition): https://github.com/rain1024/slp2-pdf

- Has FSA stuff missing from 3rd edition

The Syntax of Natural Language: An Online Introduction:

https://www.ling.upenn.edu/~beatrice/syntax-textbook/

Content warning: social media moderation—personal attacks/references to suicide



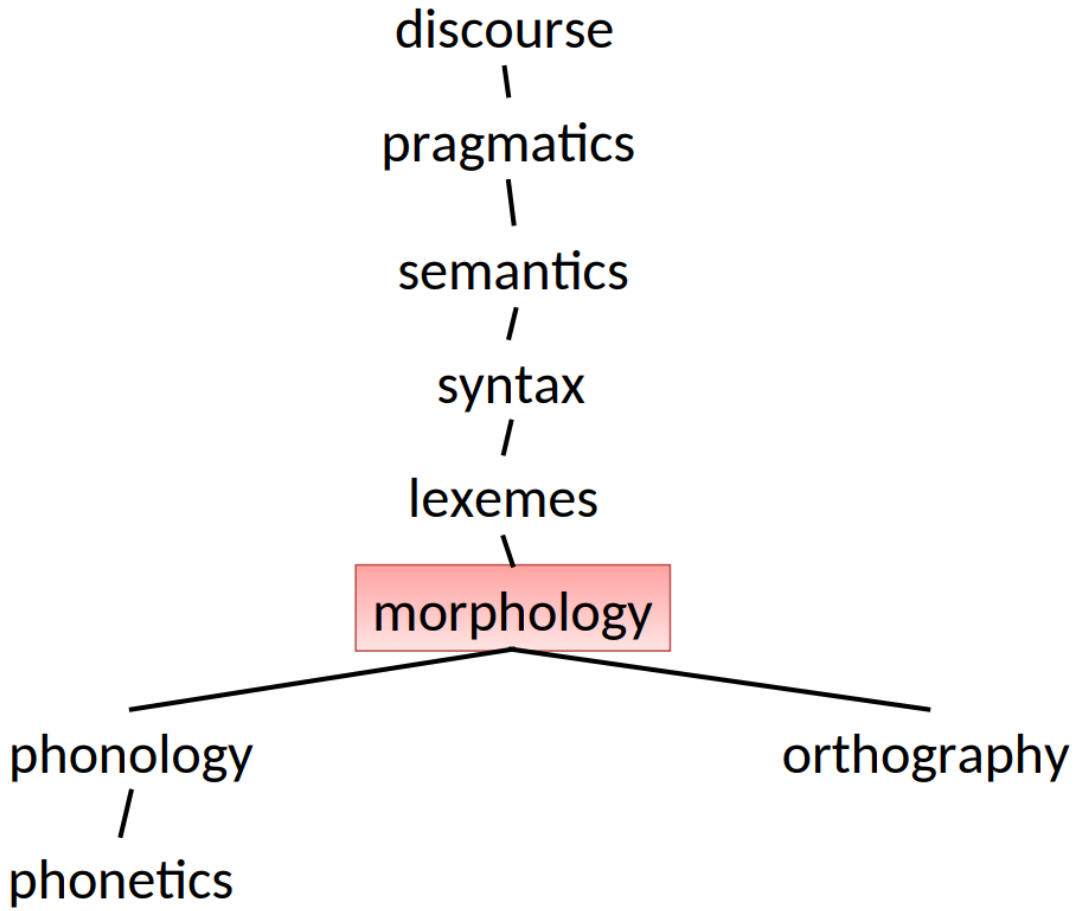
Morphology

Analysis of words into meaningful components

Run → running, ran, runs, runner…

Important for normalizing language, speech recognition, etc.

Contemporary NLP models actually look at wordpieces rather than words



Lexemes

Lexical analysis considers words one at a time.

Spelling issues

Word sense disambiguation

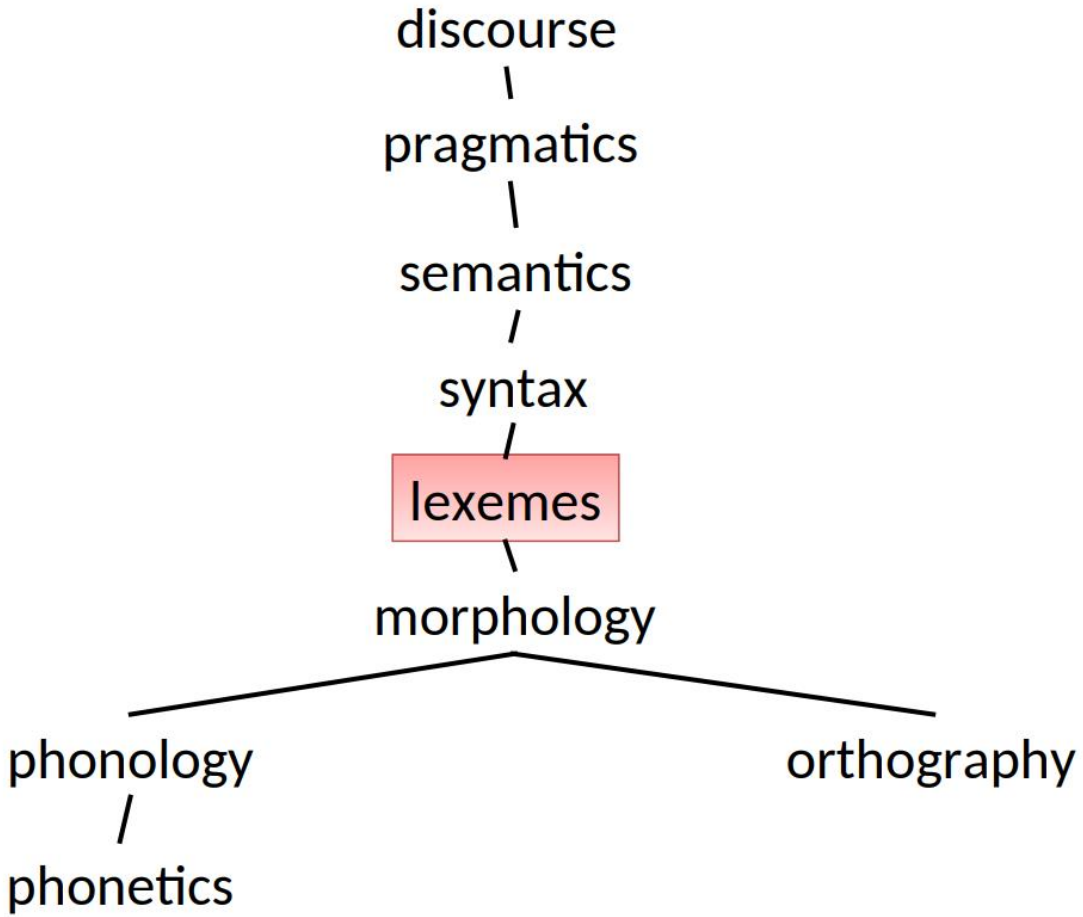
- I went to the *bank*
- I climbed the river *bank*

Multi-word expressions

- take out, make up, etc.

Part-of-speech tagging

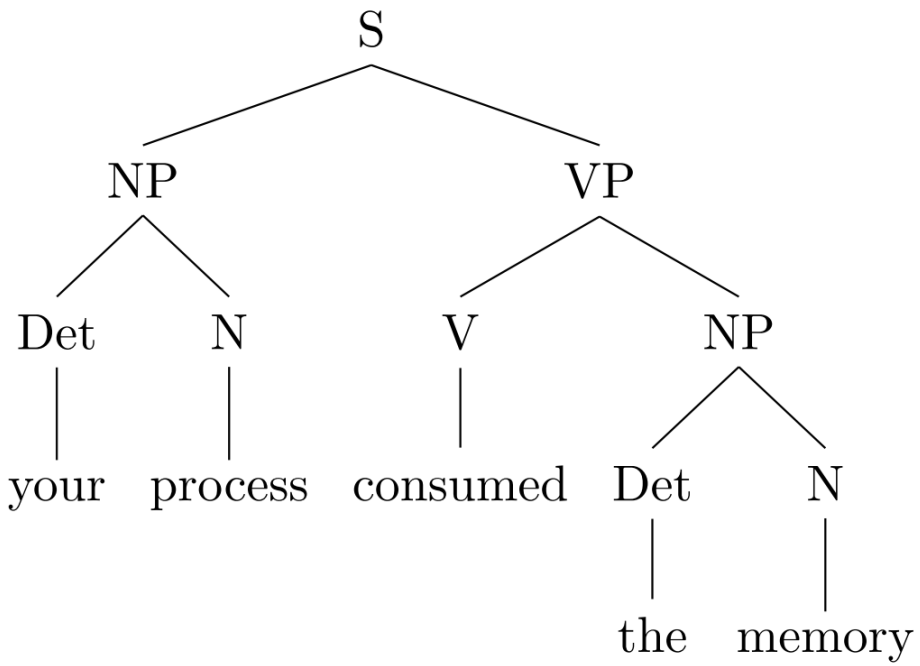
- Noun, verb, adjective, etc.

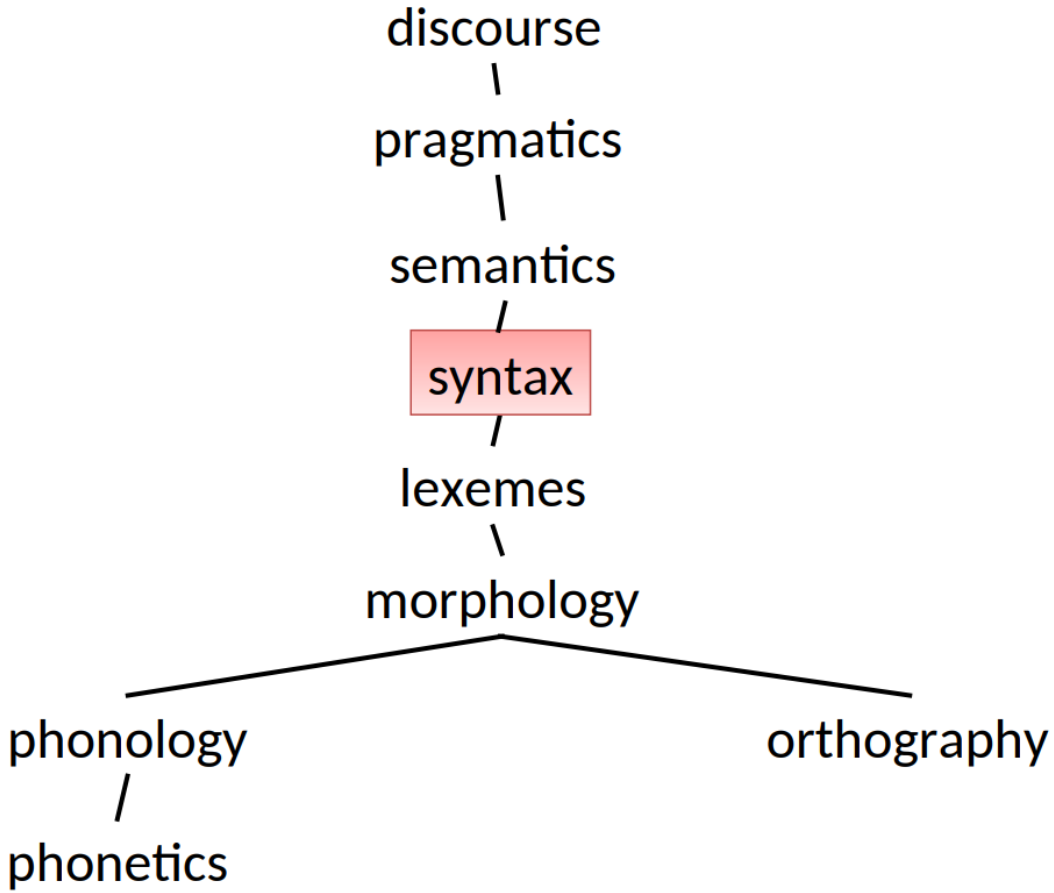


Syntax

Concerned with the compositional structure of word sequences.

Grammar

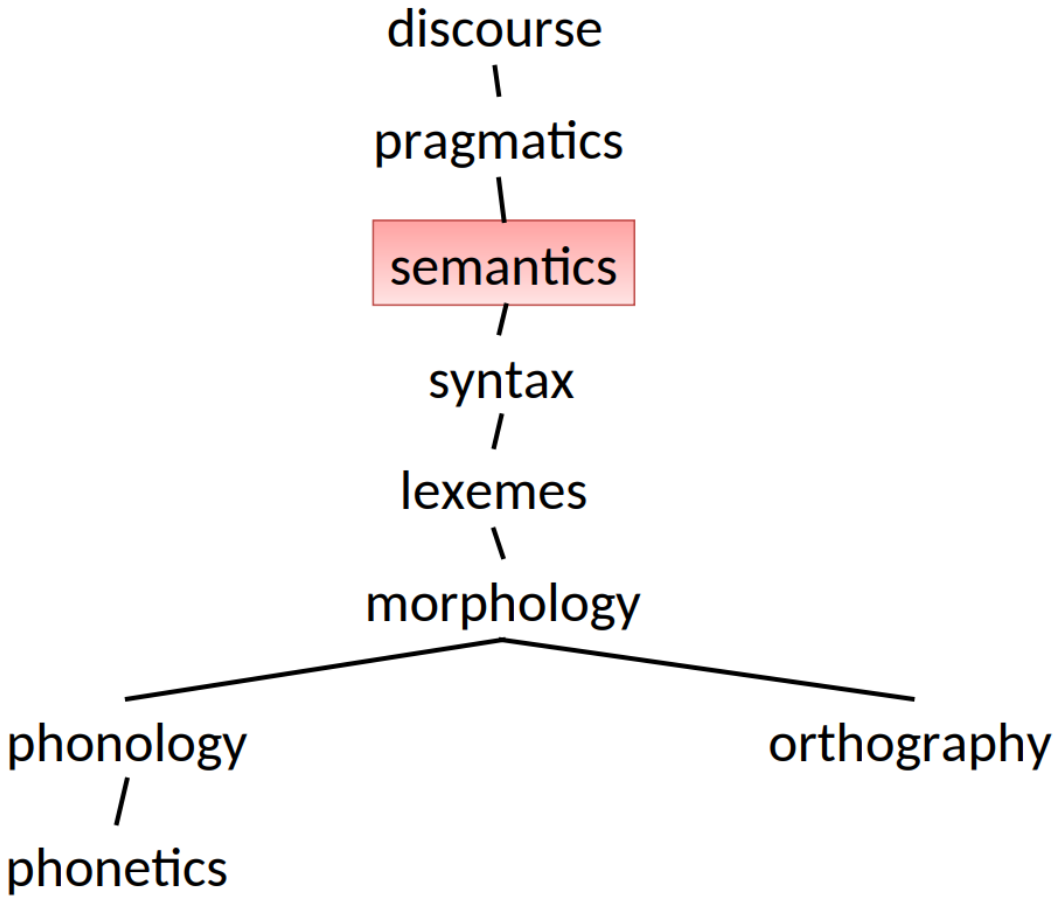




Semantics

Mapping language to meaning.

“Alexa, play Despacito” → [Alexa plays Despacito]



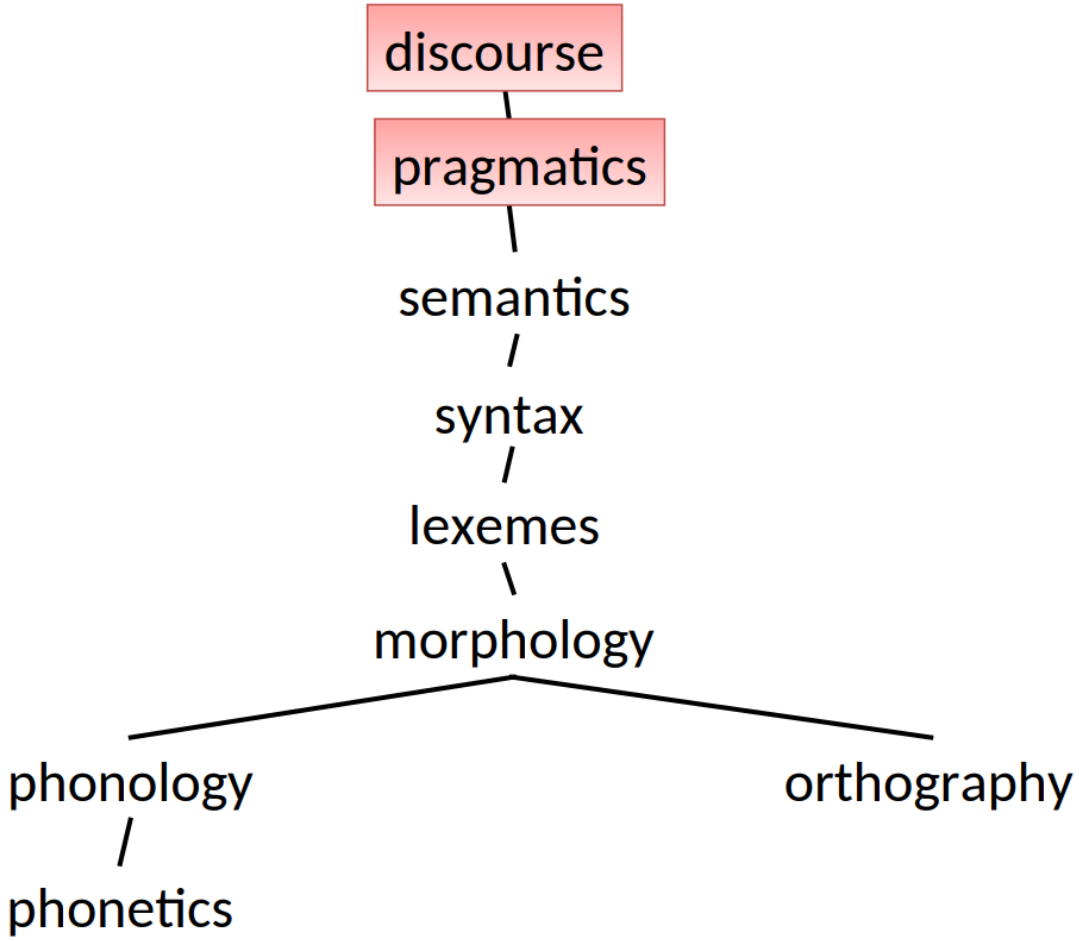
Pragmatics and discourse

Pragmatics: Effect of context on meaning

- "Can you pass the salt?"
- "Is he 21?" "Yes, he's 25."

Discourse: effect of social context on meaning

- Texts, dialogues, multi-party conversations.



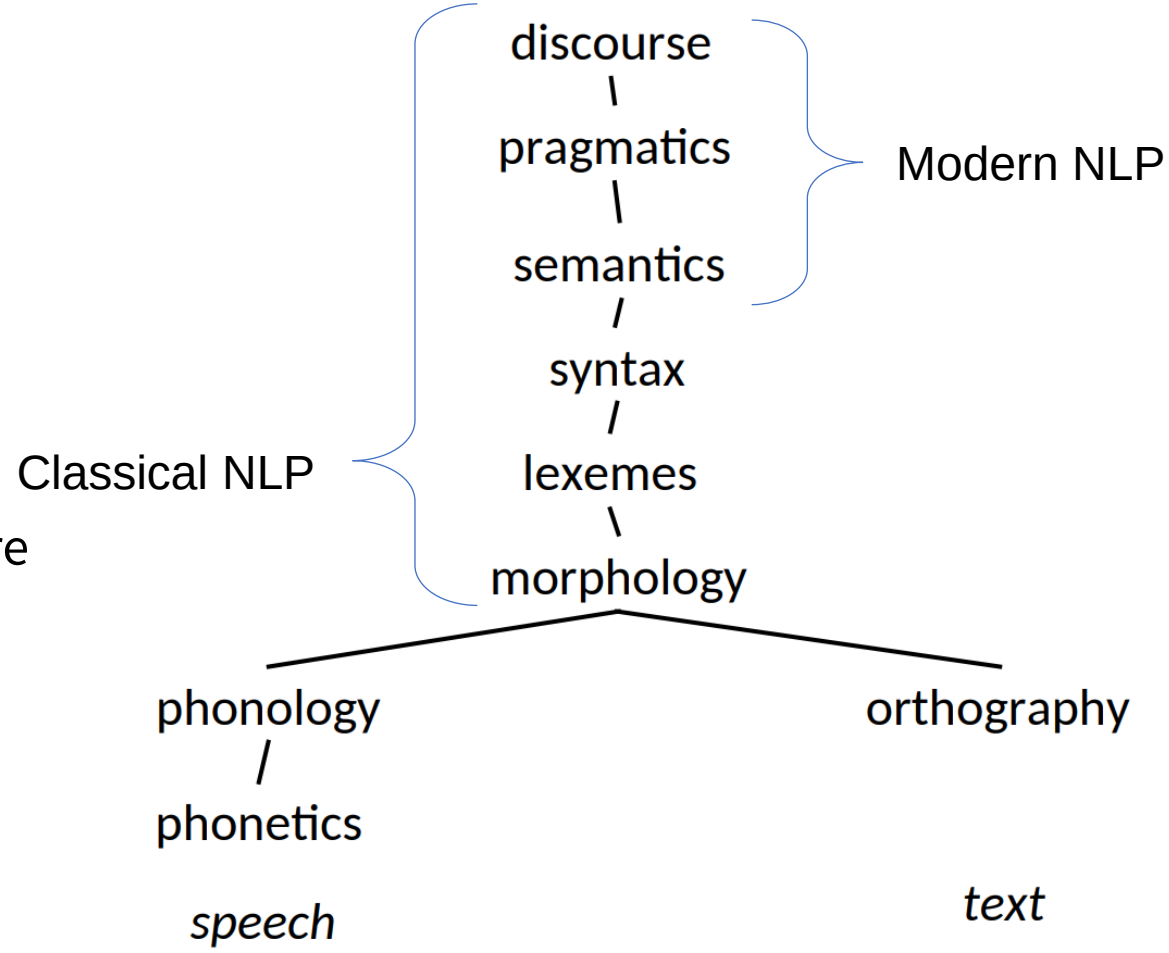
In practice

In modern NLP, semantics, pragmatics and discourse tend to blur together

In classical NLP, we tended to build up pipelines of functionality, from morphology up through discourse

Modern NLP tends to just let the models figure out the lower levels.

Still important to know though.



Morphology

What is morphology?

Words have internal structure

Words are composed of morphemes.

A morpheme is the **minimum meaningful component of a word**

- Smaller than a word (often)
- Bigger than a character

Examples:

- Misunderstandings → **mis**-understand-**ing**-**s**
- 同志们 Tongzhimen (“comrades”) → tongzhi-**men**

Types of morpheme

Roots

- Central morpheme which carries the main meaning
  - **run**-ning

Affixes

- Prefixes
  - **Pre**-nuptual, **ir**-regular
- Suffixes
  - Run-**ning**, iterat-**or**
- Infixes
  - Pennsyl-**f***ing**-vanian
- Circumfixes
  - **En**-light-**en**

Nonconcatenative morphology

Apophany

- Foot → feet, tooth → teeth
- Sing → sang, sung
- Related to ablaut, umlaut

Root-and-pattern or templatic morphology

- Common in Arabic, Hebrew, and other Afroasiatic languages
- Roots made of consonants, into which vowels are shoved

Infixation

- Gr-um-adwet

https://en.wikipedia.org/wiki/Nonconcatenative_morphology

Types of morphology

Inflectional morphology

- Adds information to a word consistent with its context in a sentence
- Examples
  - Pluralization: automaton → automata
  - Conjugation: walk → walks
  - Case: he, his, him

Derivational morphology

- Creates new words with new meanings (often with new parts of speech)
- Examples
  - Parse → parser
  - Repulse → repulsive
  - Purpose → repurpose

Lemmatization and stemming

For many NLP tasks, it is important to remove inflectional morphology but keep derivational morphology

- E.g. if we search “repurpose” on Google, we probably want results containing “repurposing”, but not ones containing “purpose”

Stemming “chops off” inflectional affixes to yield “stem”

- Studies → studi
- Studying → study
- Can be done without dictionary, but has limitations

Lemmatization returns the “dictionary” form of a word (which may be different from the root morpheme)

- Studying, studies → study
- Requires language dictionary to do right

Irregularity

Formal irregularity
Inflectional marking can differ depending on the base word

- I walk, I walked, the dog was walked
- I sing, I sang, the song was sung
- I run, I ran, the race was run

Semantic irregularity/unpredictability
The same derivational morpheme may have different meanings depending on the base it attaches to

- Kind → kindly
- Slow → slowly

Morphological typology

Different languages have different morphological characteristics

- **Isolating/analytic**: Very little inflectional morphology and not rich in derivation
  - Examples: Chinese, English
- **Agglutinative**: Many affixes that can be stacked together ad nauseum
  - Examples: Turkish, Telugu
- **Fusional/flexional**: Many inflectional meanings packed into single affixes, so morphologically rich without “stacking”
  - Examples: Spanish, German
- **Templatic**: Special type of fusional language which makes changes to root rather than via affixes
  - Examples: Arabic, Amharic

Notation

Level	hugging	panicked	foxes
Lexical form	hug +V +Prog	panic +V +Past	fox +N +Pl fox +V +Sg
Morphemic form (intermediate form)	hug^ing#	panic^ed#	fox^s#
Orthographic form (surface form)	hugging	panicked	foxes

Lexical form is a standardized way to show how a root morpheme relates to its **orthographic form** (how it appears in text)

- hug+V+Prog → "verb form of 'hug' in the progressive tense" → hug^ing# → hugging

Morphemic form is sometimes used as an intermediate stage

- This will actually show up again when we talk about transformers, so put it in your pocket for later

Finite state machines (FSMs)

Old technology, (supposedly) still used in industry, probably still useful to know about

Graphical representation of a set of rules which are used to generate a set of strings

- Such as a morphology
- **Finite state automata** definite transitions (and are simpler), **finite state transducers** define transformations (and are more complicated)

Example: what are the rules for generating the noise a sheep makes, of possibly infinite length, with an exclamation point on the end?

- ~~ba!~~
- baa!
- ~~aaaaaaaaaaaaaa!~~
- baaaaaaaaaaaaaaaaaaaaaa!
- ~~baaaaaaaaaaaaaaaaaaaaaaa~~

A FSA for sheep noises

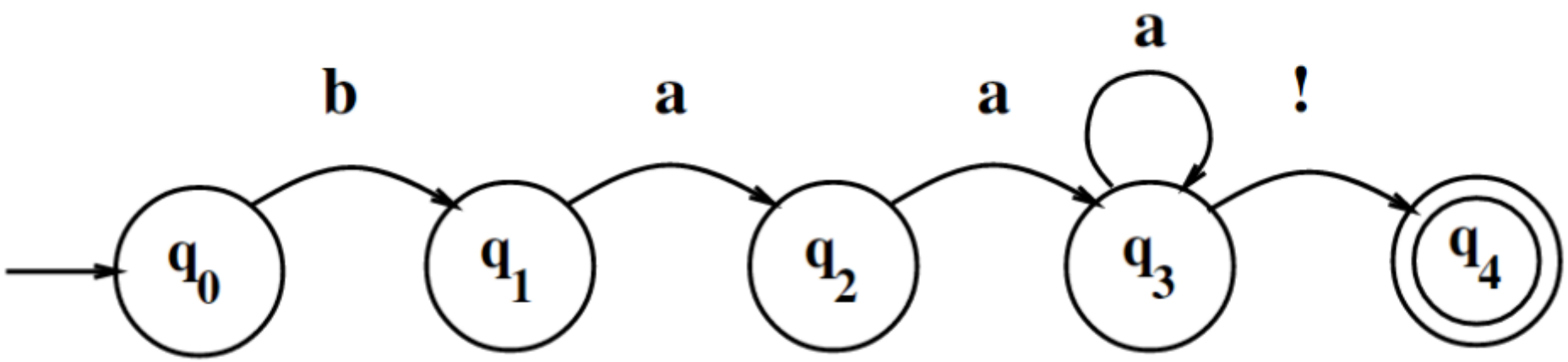


Figure 2.10 A finite-state automaton for talking sheep.

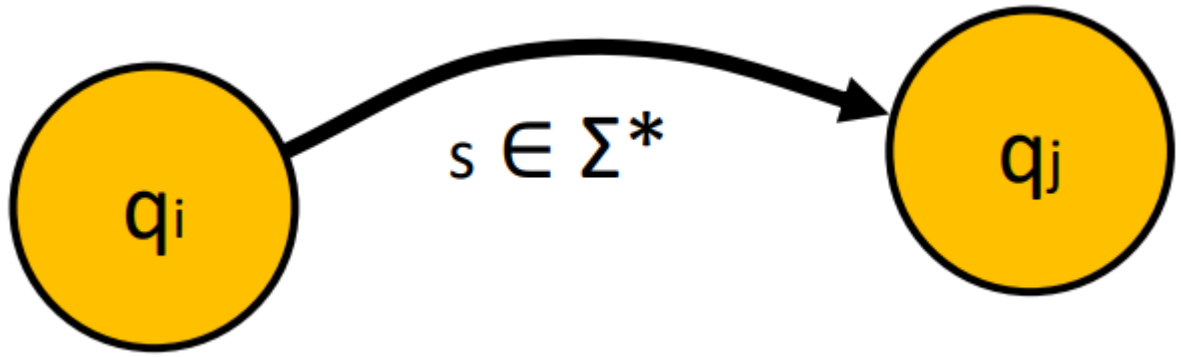
Finite state automata

Q: a finite set of states

$q0 \in Q$: special start state

$F \subseteq Q$: set of final states

Σ: Finite alphabet



Encodes a set of strings that can be recognized by following paths from q0 to some final state in F

A FSA for sheep noises

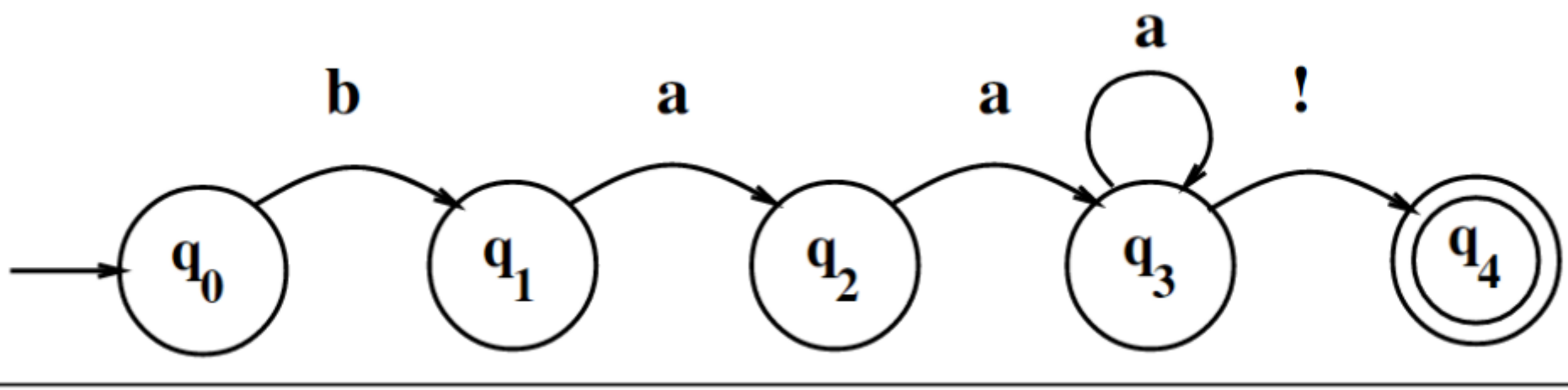


Figure 2.10 A finite-state automaton for talking sheep.

~~ba!~~
baa!
~~aaaaaaaaaaaaa!~~
baaaaaaaaaaaaaaaaaaaaaa!
~~baaaaaaaaaaaaaaaaaaaaaa~~

Formal, natural, and regular languages

A **formal** language is a language with a specific vocabulary and rules, made by humans for some set purpose

- All programming languages are formal languages

A **natural** language is a real language. They are messy, inconsistent, and constantly changing

- Though in NLP, we try to impose structure on them so that we can work with them computationally

A **regular** language is a formal language that can be recognized by a finite state automata

- Natural languages are not regular, but their **morphologies** mostly are (though complex)

Regular expressions

Regular expressions are FSAs

- Note use of the word “regular”

Super useful, not covered in this class

Regex for the sheep example: “baaa.!” or “baa+!” or “ba{2,}!”

https://www.regular-expressions.info/

FSA for English derivational morphology

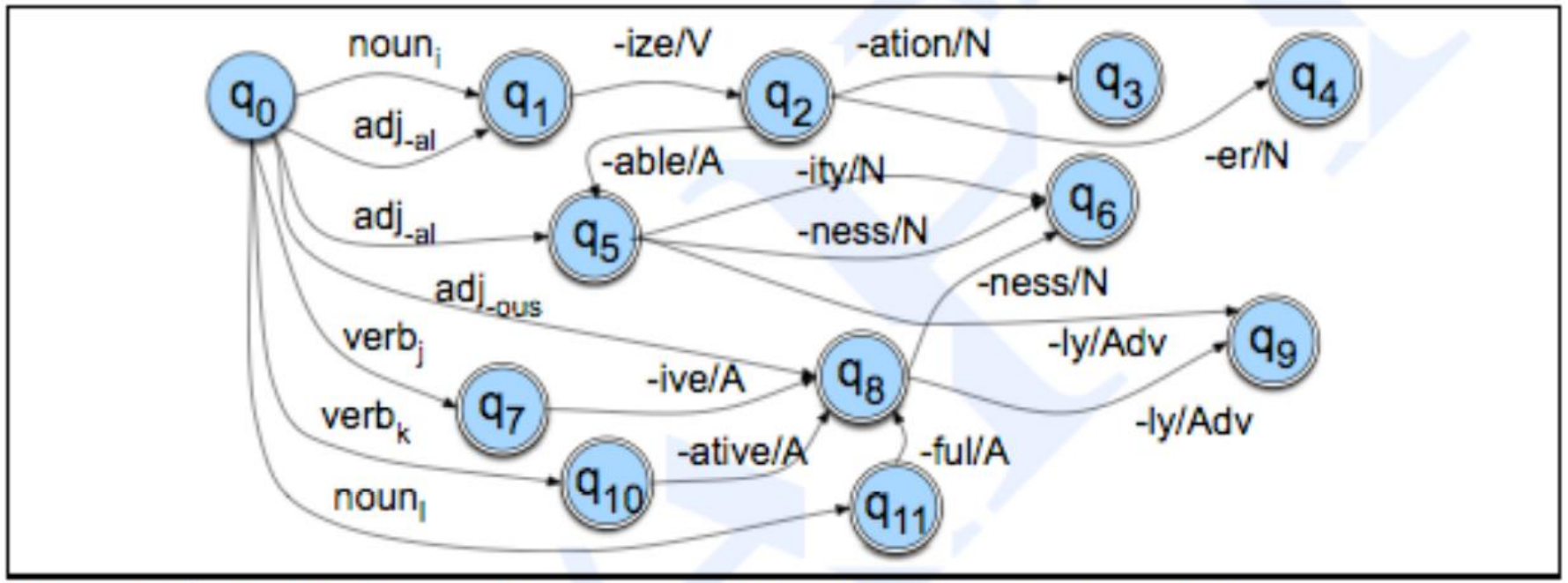


Figure 3.6 An FSA for another fragment of English derivational morphology.

https://github.com/rain1024/slp2-pdf/blob/master/chapter-wise-pdf/%5B03%5D%20Words%20%26%20Transducers.pdf

FSA for English nominal inflection

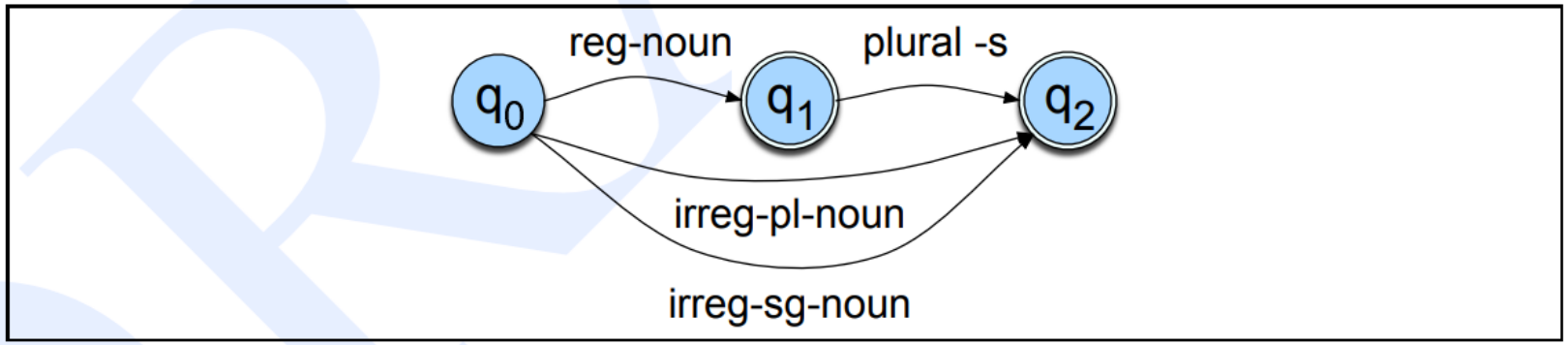


Figure 3.3 A finite-state automaton for English nominal inflection.

reg-noun	irreg-pl-noun	irreg-sg-noun	plural
fox	geese	goose	-s
cat	sheep	sheep	
aardvark	mice	mouse	

A similar model for English verbal inflection might look like Fig. 3.4.

https://github.com/rain1024/slp2-pdf/blob/master/chapter-wise-pdf/%5B03%5D%20Words%20%26%20Transducers.pdf

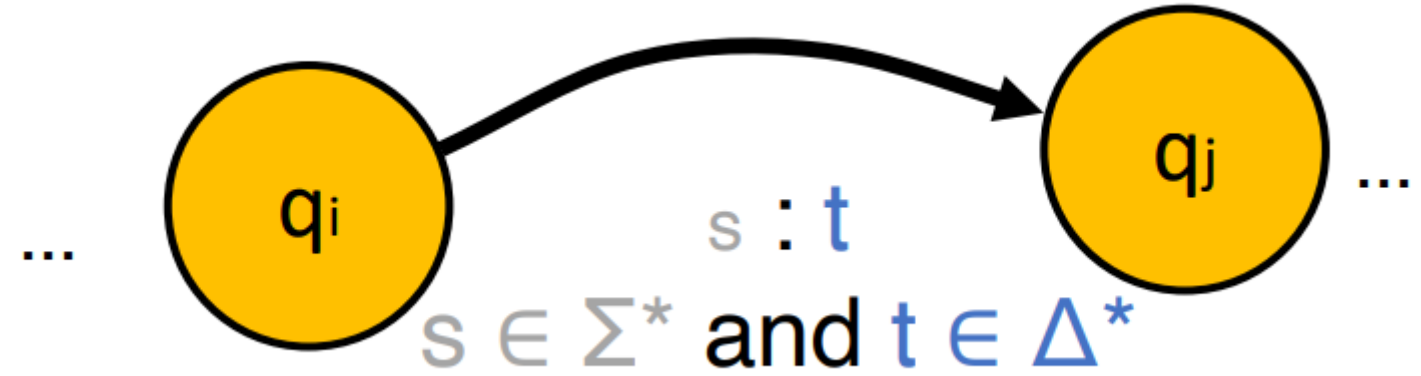
Finite state transducers

Q: a finite set of states

q0 ∈ Q: special start state

F ⊆ Q: set of final states

Σ and Δ: Two finite alphabets



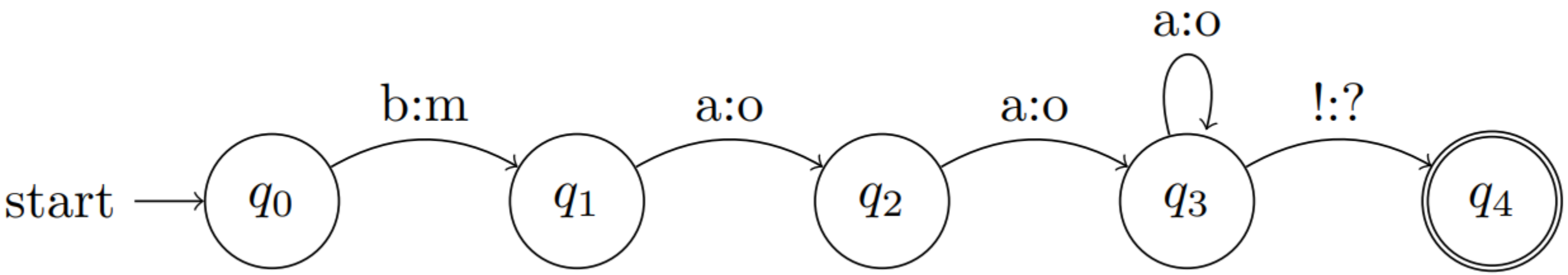
Encodes a relationship between two sets of strings

If a FSA is a “recognizer” or “approver”, then a FST is a “translator”

Translating from sheep exclamation to cow question

“baaaaaa!” → “moooooo?”

“baa!” → “moo?”



Morphological parsing with FSMs

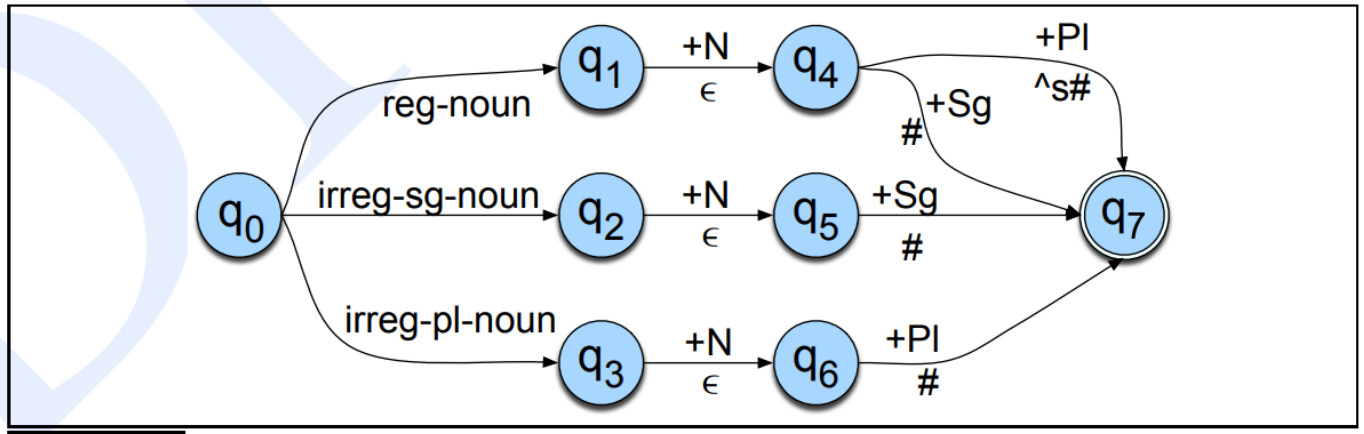


Figure 3.13 A schematic transducer for English nominal number inflection T_{num}. The symbols above each arc represent elements of the morphological parse in the lexical tape; the symbols below each arc represent the surface tape (or the intermediate tape, to be described later), using the morpheme-boundary symbol ^ and word-boundary marker #. The labels on the arcs leaving q_0 are schematic, and need to be expanded by individual words in the lexicon.

This would convert morphemic form to lexical form:

- cat^s# → cat +N +Pl
- cow^s# → cow +N +Pl
- teeth# → teeth + N + PL

https://github.com/rain1024/slp2-pdf/blob/master/chapter-wise-pdf/%5B03%5D%20Words%20%26%20Transducers.pdf

Lexical analysis

Lexical analysis

Lexeme: a word or multi-word expression (which can be treated like a single word)

- Dog, cat, bust-up, take-home, take-out

Two important tasks: part-of-speech identification and word sense disambiguation

The two tasks are related:

- Business is going **well**. [adverb]
- All is **well** with us. [adjective]
- **Well**, who would have thought he could do it? [interjection]
- The **well** was drilled fifty meters deep. [noun]
- Tears **well** up in my eyes. [verb]

Parts of speech

Define the type of role a word can play in a sentence

Divided into **open-class** and **closed-class**

Open-class parts of speech

- Nouns
- Verbs
- Adjectives
- Any class to which you can easily add new members

Closed-class parts of speech

- Pronouns
- Determiners
- Conjunctions
- Any class to which you cannot easily add new members

Interjection: subjects, verbs and objects

Sentence phrases in English generally consist of a subject, a verb, and an object, describing one entity doing something to a second entity

- "I walked the dog
- "I went for a run
- "She went outside

The **subject** is the entity doing the thing, the **verb** is what's being done, and the **object** is what it is being done to.

English is a **subject-verb-object** (SVO) language—other languages use different orders.

Nouns

- Open-class
- Can be subjects and objects of verbs
  - This **book** is about **geography**.
  - **I** read a good **book**.
- Can be objects of prepositions
  - **I**'m mad about **books**.
- Can be plural or singular (books, book)
- Can have determiners (the book)
- Can be modified by adjectives (blue book)
- Can have possessors (my book, John's book)

Verbs

- Open-class
- Takes nouns phrases as arguments
  - At least a subject
    - Dr. Mortensen **parsed** aggressively.
- Sometimes one or two objects
  - Dr. Mortensen **parsed** the data.
  - Prof. Black **passed** [the function] [an argument].
- Can take tense morphology (past/non-past)
- Can be modified by adverbs

Adjectives

Modify nouns

- Open-class
- his **pitiful** code (attributive)
- His code is **pitiful**. (predicative)

Can take comparative/superlative (-er/-est) suffixes when allowed by prosody

- **big**, **bigger**, **biggest**
- But **pitiful**, more **pitiful**, most **pitiful**

Not all languages have adjectives—some languages (like Korean, Hmong, and Vietnamese) use verbs to modify nouns in this way

Adverbs

Modify verbs, adjectives and other adverbs

- Open-class
- He **erroneously** concluded that PHP is a real programming language **simply** because it is Turing complete.
- He concluded **erroneously** that PHP is a real programming language.
- The design of PHP is **exceptionally poor**.
  - Is this correct?
- My code runs **very slowly**.

Prepositions

Occur before noun phrases

- Closed-class
- Relate noun phrase to some higher-level constituent
  - I scattered the data **from** hell **to** breakfast.
  - He lingered **in** the depths **of** despair.
- It is actually not difficult to characterize prepositions formally, but they are very difficult to characterize semantically (a good argument not to introduce semantic considerations into PoS categories)
- Also, they are often identical in spelling and pronunciation to particles
- **Other examples:** at, by, about, near, between, before, after, across, along

Determiners

Determiners are words that come at the beginning of noun phrases in English

- The most recognizable determiners are probably **articles** like “the”, “a”, and “an”
  - **The** interpreter choked on **an** unknown identifier.
- Other determiners include:
  - Demonstratives like “this” and “that”.
    - **That** book is good.
  - Possessives like “my” and “their”
    - **My** book is good.
  - Quantifiers like “many” and “both”
    - **Both** books are good.

Pronouns

Pronouns replace noun phrases, acting as a sort of shorthand for them

- **You** are a good person.
- **Your** type system is not well-founded.
- **I**’m not going to give the password to **you**.
- **Who** knows Haskell, really?
- **He** thought **she** wouldn’t think to close **their** door.

Conjunctions

Conjunctions join phrases, clauses, or sentences.

- Typically, the conjuncts joined by a conjunction are of the same time
- **Coordinating conjunctions**
  - “and”, “or”, “but”…
- **Subordinating conjunctions**
  - “if”, “because”, “though”, “while”…

Auxiliary (helping) verbs

"Helping verbs" that occur before main verbs

- Some occur as main verbs as well
  - Be
    - I **am** the type system. (main verb)
    - I **am** working on my project, you insensitive clod. (aux. verb)
  - Have
    - I **have** no qualms about criticizing your choice of languages. (main verb)
    - I **have** written a brilliant function that **will** accomplish just that! (aux. verb)
- Others (e.g. modals) occur only as auxiliary verbs
  - would, will, could, can, might, must…

Particles

Particle is sometimes used as a grab-bag category for closed-class items that do not fit in another category

- “Somewhat nebulous term for a variety of small words that do not conveniently fit into other classes of words” (Wikipedia)
- Most often, in English, these resemble prepositions or adverbs and are used in combination with a verb
  - He tore **off** his shirt.
  - He took the dog **out**.

Numerals

Numerals have properties of both nouns and adjectives

- They can be the subject and object of verbs:
  - **Two** will enter but only **one** will leave.
  - I bought **twenty**.
- They can function both attributively and predicatively:
  - **Two** variables were undeclared.
  - We are **three**.
- When then are used attributively, they come before any adjectives:
  - The **two** undeclared variables were the cause of much consternation.
  - *The undeclared **two** variables were the cause of much consternation.
    - Not actually grammatical

Other POS types

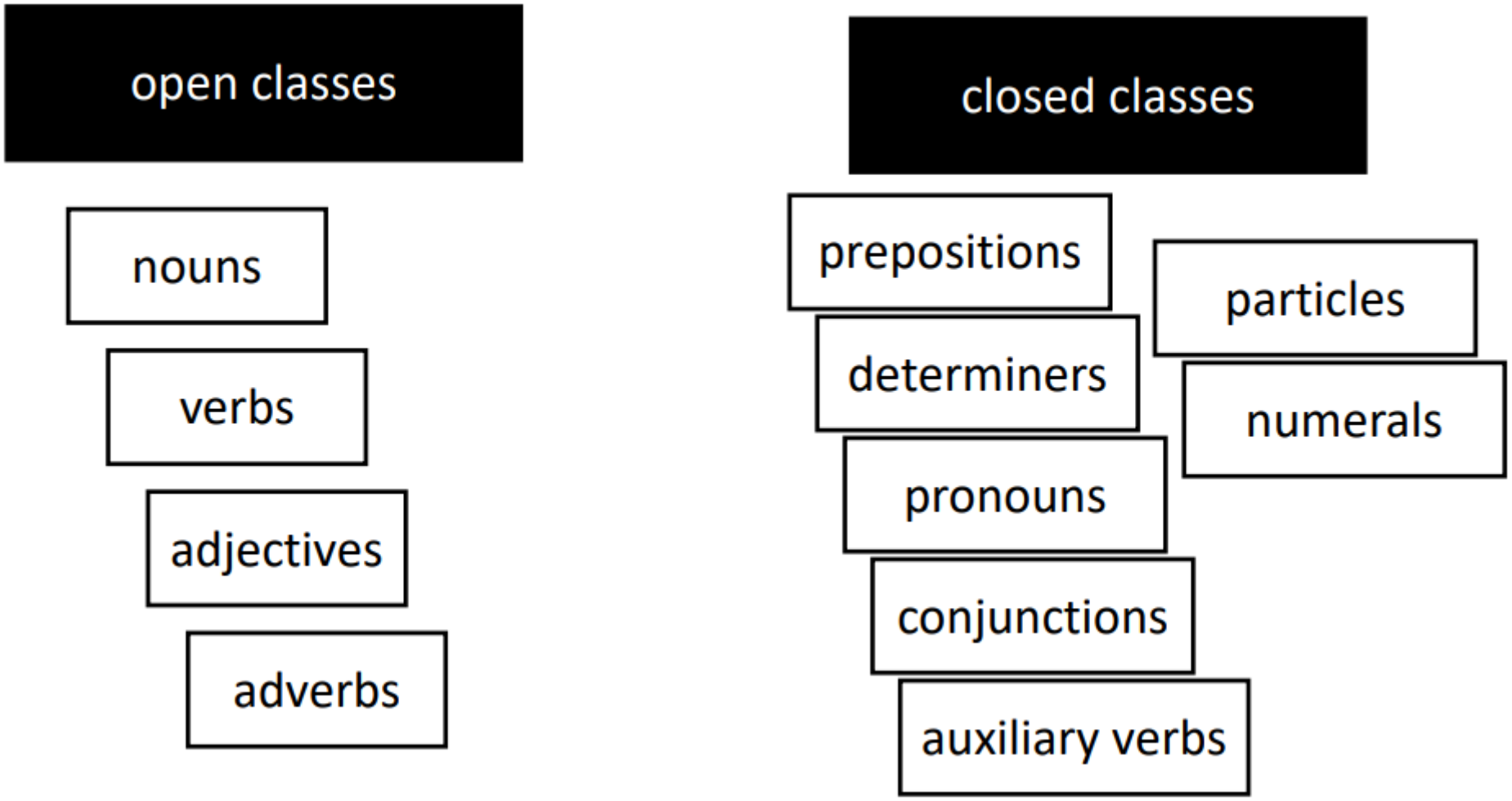
The exact set of POS types varies from theory to theory

- For example, some theorists don't consider pronouns their own class

These other categories will also sometimes show up in e.g. POS taggers

- Interjections
- Negatives
- Politeness markers
- Greetings
- Existential "there"
- Numbers, Symbols, Money, …
- Emoticon
- URL
- Hashtag

Broad POS categories



Word sense disambiguation

Important to understand meaning of text, but often bleeds into semantics

Sometimes POS is all you need:

Ex. Is “jerk” being used in a toxic way?

- “I love **jerk**[adjective] chicken”
- “You are a huge **jerk**[noun]”

But sometimes it isn’t enough

- “That guy is a real **prick**[noun]”
- “The flu shot isn’t so bad, just a little **prick**[noun]”

Syntax

Syntax

Deals with the structure of phrases and sentences

Not the same as morphology, which deals with the internal structure of individual words

Or lexical analysis which deals with the category and role of individual words

Also not semantics; deals with structure regardless of meaning

- Sentence can be syntactically valid but meaningless
  - *Colorless green ideas sleep furiously.*

Constituency

One way to view the structure of a sentence is as a collection of nested **constituents**

Constituent: a group of neighboring words that “go together”

A constituent larger than a single word is a **phrase**

Phrases can contain other phrases

Noun phrases (NP)

Generally consist of a single main noun, and some amount of other stuff that modifies it (determiners, adjectives, prepositional phrases, etc.)

- **The elephant** arrived.
- **It** arrived.
- **Elephants** arrived.
- **The big ugly elephant** arrived.
- **The elephant I love to hate** arrived.
- **The elephant and the hippo** arrived at the same time.

Verb phrase (VP)

Made up of a verb and its dependents (objects, complements, and modifiers)

- Not its subject, however
- Examples
  - John **went for a walk**.
  - John **went for a walk in the park.**
  - John **went for a walk in the park with Mary, who was depressed because she had lost her job.**
  - John went for a walk in the park with Mary, who **was depressed because she had lost her job.**
  - John went for a walk in the park with Mary, who **was depressed** because she had lost her job.

Adjective phrases (AP)

Consists of one or more adjectives and modifiers

Examples

- John is **smart**.
- John is **smart and handsome**.
- John is **sharp as a tack and as handsome as the day is long.**
- John is sharp as a tack and **as handsome as the day is long.**

Prepositional phrase (PP)

A prepositional phrase consists of a preposition and another phrase, and usually modifies a sentence, verb phrase or noun phrase.

- I arrived **on Tuesday**.
- I arrived **in March**.
- I arrived **under the leaking roof**.

Sentences or clauses (S)

A sentence or clause is composed of a noun phrase (NP) and a verb phrase (VP) of which it is the subject

A full sentence can have multiple sentence clauses

Examples:

- *John* **went for a walk**
- *John* **went for a walk while he called his mother**
- John went for a walk while **he called his mother**

Context-free grammars (CFGs)

A **grammar** is a set of rules which defines how constituents relate to each other

- Grammar is to words as morphology is to morphemes
- e.g definition of a sentence: S → NP VP
- e.g. definitions of a noun phrase:
  - NP → N
  - NP → Det NP

A **context-free grammar** has only one symbol on the left—the identity of a constituent doesn't depend on its context

Example CFG

Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats

The dog eats a bone

Example CFG

Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats

DET NN V DET NN

The dog eats a bone

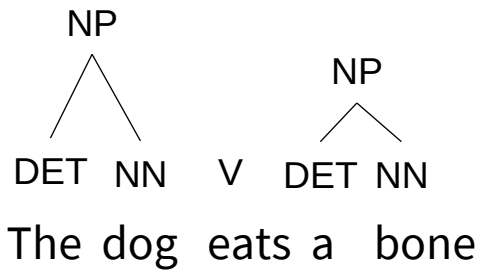
Example CFG

Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats



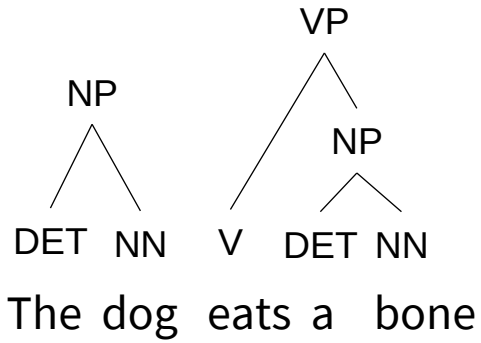
Example CFG

Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats



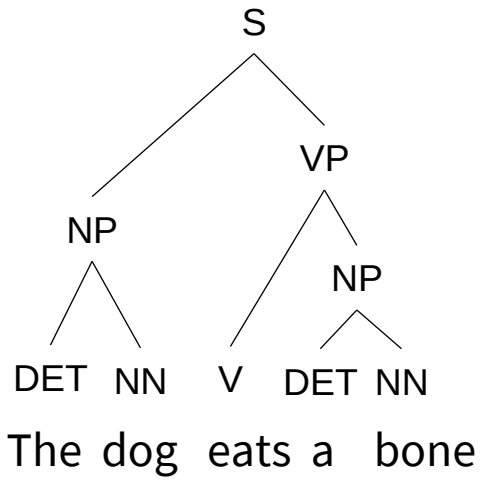
Example CFG

Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats



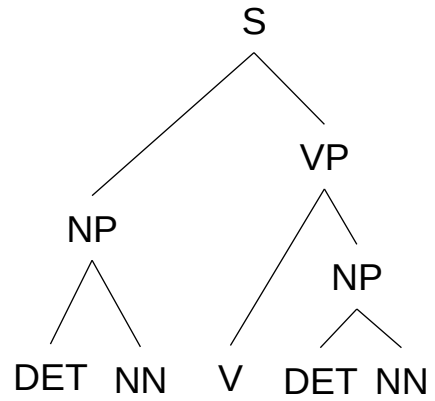
Example CFG

Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats



A dog hates cats

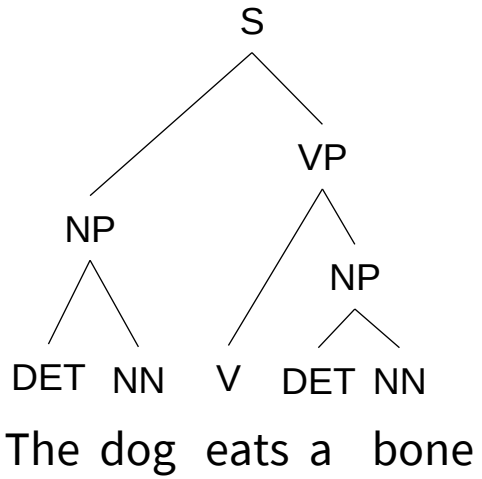
Example CFG

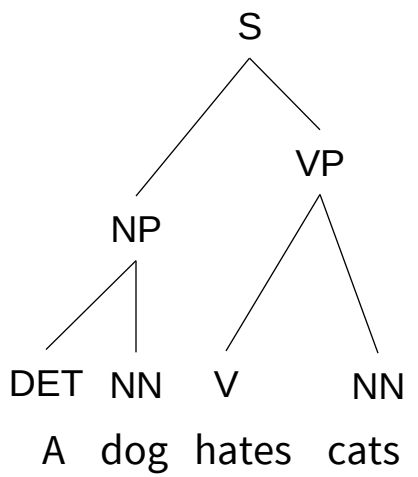
Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats





What is wrong here?

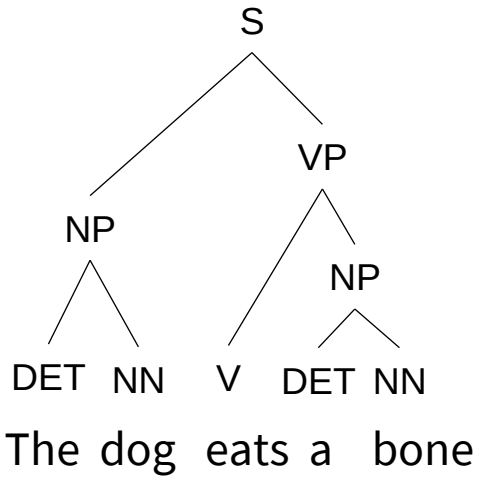
Example CFG

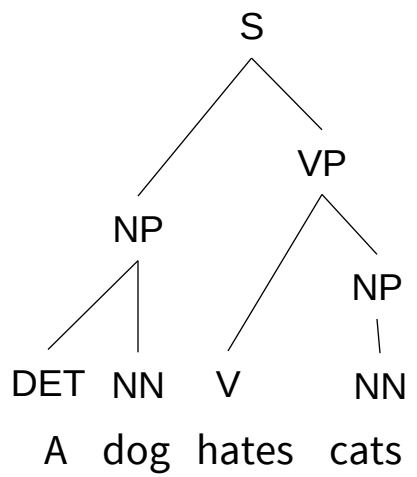
Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP
- NP → Noun

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats





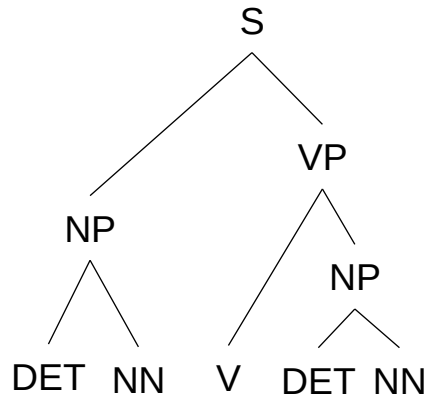
Example CFG

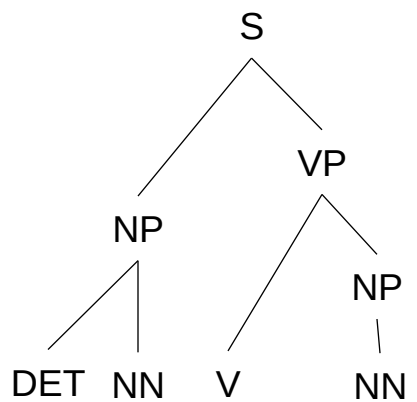
Phrases

- S → NP VP
- NP → Det Noun
- VP → Verb NP
- NP → Noun

Parts of speech

- Det → the, a
- Noun → boy, girl, hotdogs, bone
- Verb → likes, hates, eats





Other valid examples

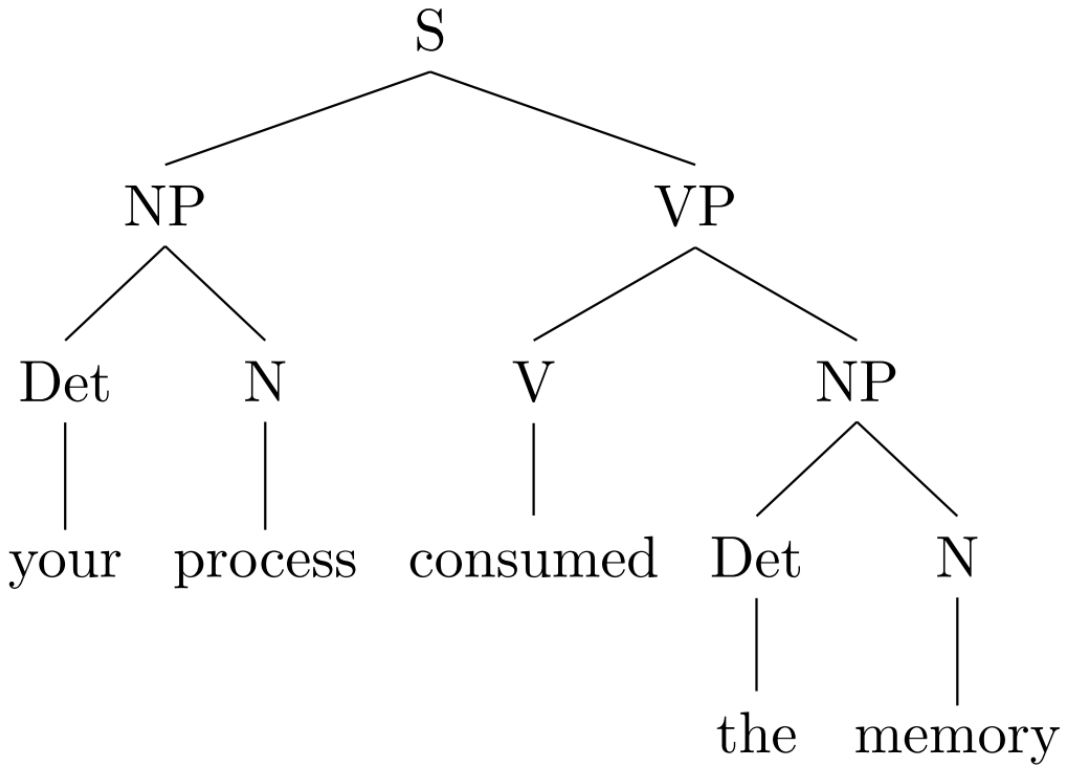
- "The hotdogs likes a hotdogs"
- "A hotdogs eats boy"

So, semantics not considered here—only grammar

Constituency (parse) trees

A pretty common task in linguistics (syntax) and older-style NLP is to **parse** text into a parse (constituency) tree

Goal is to learn/apply a good grammar for natural language in order to consistently parse natural text to a structured format



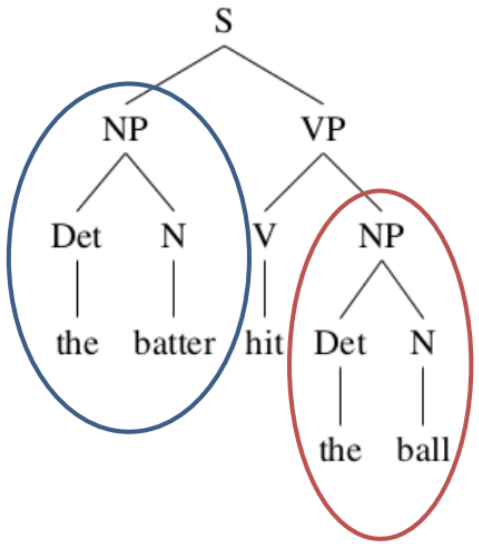
Constituency vs dependency trees

An alternative way to parse the structure of a sentence is to mark the relations of words to each other.

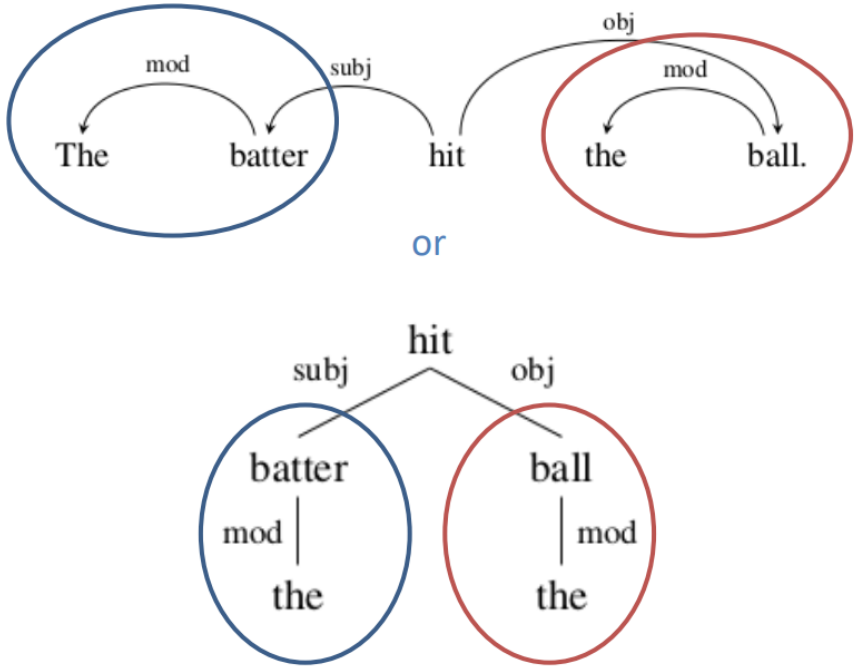
Dependency relations include:

- Subject
- Object
- Modifier

Constituency tree



Dependency tree



Closing thoughts on syntax

Very big topic

- More details: LING 405 or ENGL 405 and followups

This lecture based on so-called “standard theory”

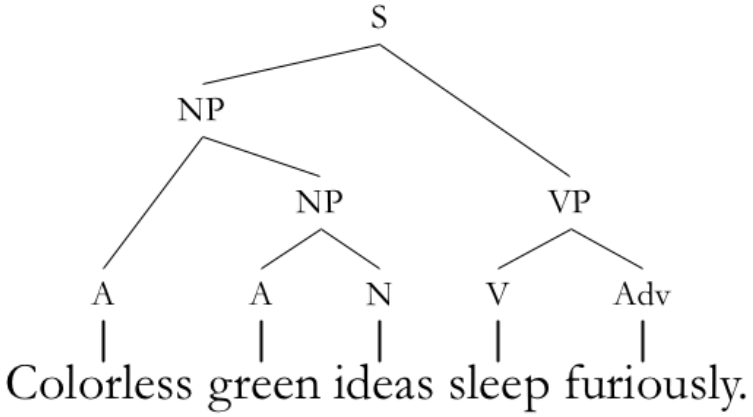
- https://en.wikipedia.org/wiki/Generative_grammar#Standard_theory_(1956%E2%80%931965)

Compare and contrast with “X-bar theory”

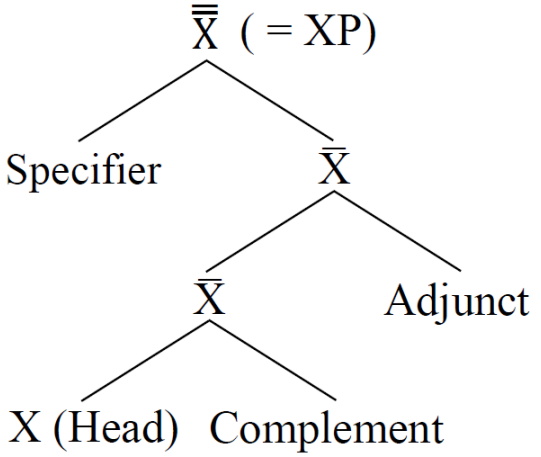
- https://en.wikipedia.org/wiki/X-bar_theory

In NLP: worth knowing about but doesn’t really help you do anything

Standard constituency tree



X-bar constituency tree



Applications (Content warning)

Non-applicability

Generally speaking, we're not (anymore) using linguistic theory to inform NLP.

- Not using morphology for tokenization https://huggingface.co/learn/nlp-course/en/chapter6/6
- Not using syntax for anything much

Constituency parsing **is** a classic NLP task

- …but to what end?

Contemporary models learn what they need to learn, directly from the data, to do what they need to do

GPT-4 moderation prompt on made-up Reddit comment

```
Decide whether the following Reddit
comment has violated the AskReddit
subreddit rules, and if so, which part of
the comment violated which rule. Return
your decision in JSON format.

Rules:
1. Slur use is not allowed in AskReddit.
2. Don't be abusive or harass other
users.
3. Do not make blatant statements of
bigotry.
4. Do not make jokes or statements that
sexualize minors.
5. Do not encourage suicide in any form,
through humour or otherwise.

[[few-shot exemplars]]

Comment:
I can't believe you don't like this
movie. Maybe you shouldn't be around
anymore.
Decision:
{
  "decision":"violation",
  "rules violated":[5],
  "violating snippets":["Maybe you
shouldn't be around anymore."]
}
```

Feature engineering

Sometimes helpful for downstream tasks to use linguistics to generate intermediate representations

Especially in non-neural NLP

Example:

Is “jerk” being used in a toxic way?

- “I love jerk[adjective] chicken”
- “You are a huge jerk[noun]”

But neural NLP tends to just skip the intermediate steps

Human-centered NLP

People tend to think and read in phrases, so phrases may be a good way to communicate with people.

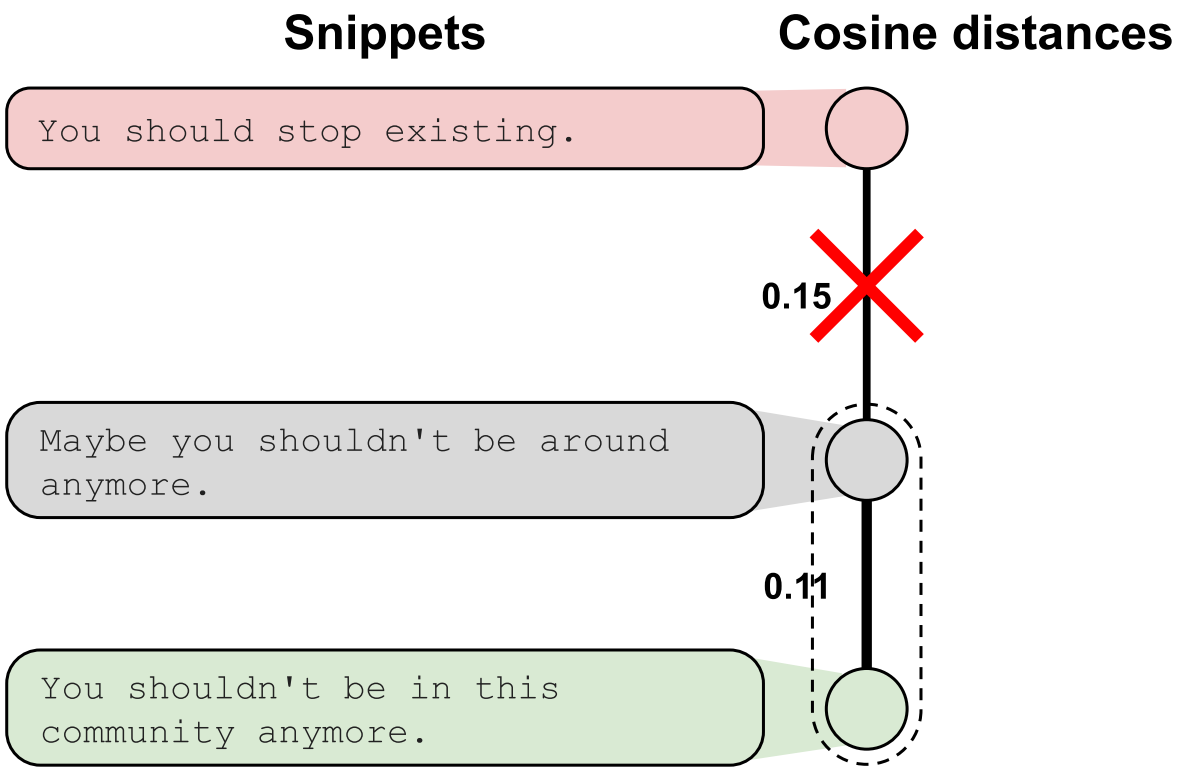
Example: rationalizing a toxicity prediction

“You are a real piece of crap, and I wish I had never met you!” → Predicted toxic

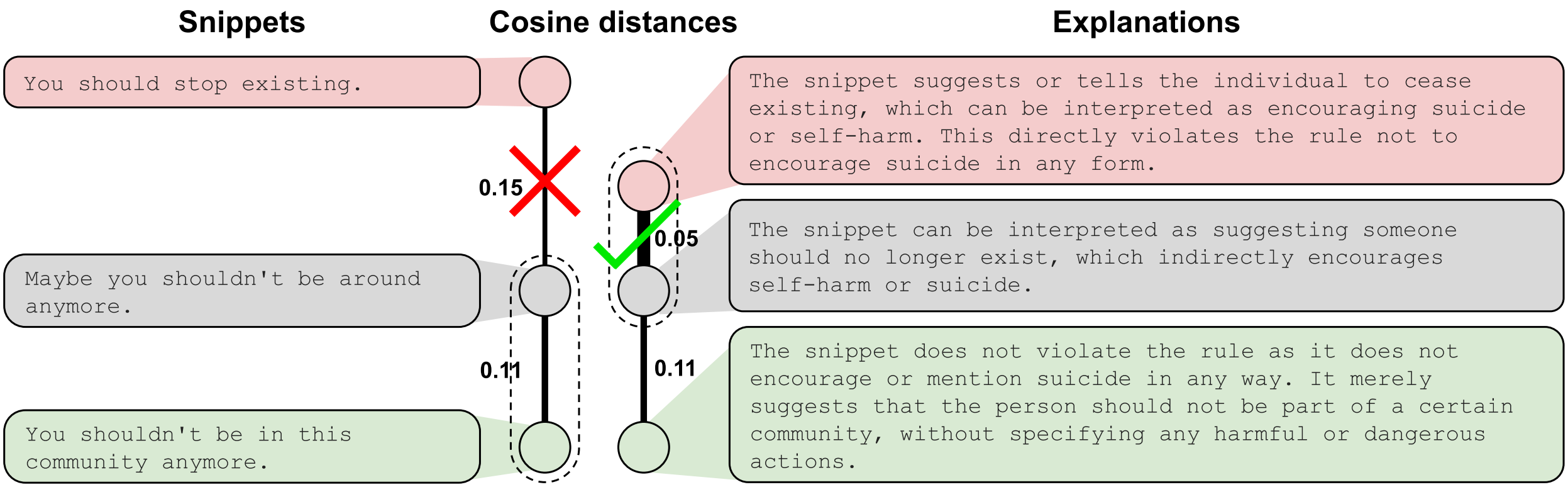
“You are a real piece of crap, and I wish I had never met you!” → Predicted toxic

“You are a real piece of crap, and I wish I had never met you!” → Predicted toxic

_____ vs _____ similarity



Lexical vs semantic similarity



rule.

Conclusion

Things you should know:

- Definitions of major linguistic categories
- Major parts of speech
- How to read/apply an FSA and FST
- How to read/apply a CFG
- Relationship between linguistics and NLP

Teaching philosophy:

- Things you know you know
- **Things you know you don't know**
- Things you don't know you don't know

See textbook links for more info/details