NLP Basics

Ruihong Huang
Texas A&M University

many slides are from Dan Jurafsky and Chris Manning

Text Normalization

- Every NLP task needs to do text normalization:
  1. Segmenting/tokenizing words in running text
  2. Normalizing word formats
  3. Segmenting sentences in running text

How many words?

they lay back on the San Francisco grass and looked at the stars and their

- **Type**: an element of the vocabulary.
- **Token**: an instance of that type in running text.
- How many?
  - 15 tokens (or 14)
  - 13 types (or 12) (or 11?)

How many words?

N = number of tokens

V = vocabulary = set of types

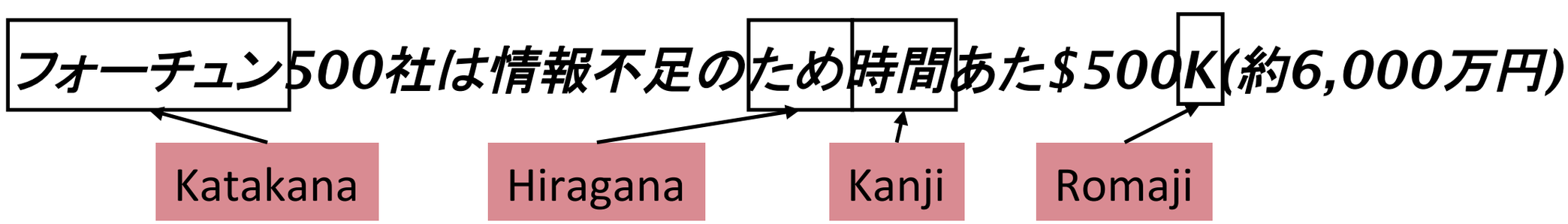
$|V|$ is the size of the vocabulary

Church and Gale (1990): $|V| > O(N^{½})$

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

Tokenization: language issues

- Chinese and Japanese no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova now lives in US southeastern Florida
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats



End-user can express query entirely in hiragana!

Normalization

- Need to "normalize" terms
  - Information Retrieval: indexed text & query terms must have same form.
    - We want to match ***U.S.A.*** and ***USA***
- We implicitly define equivalence classes of terms
  - e.g., deleting periods in a term
- Alternative: asymmetric expansion:
  - Enter: ***window*** Search: ***window, windows***
  - Enter: ***windows*** Search: ***Windows, windows, window***
  - Enter: ***Windows*** Search: ***Windows***
- Potentially more powerful, but less efficient

Case folding

- Applications like IR: reduce all letters to lower case
  - Since users tend to use lower case
  - Possible exception: upper case in mid-sentence?
    - e.g., ***General Motors***
    - ***Fed*** vs. ***fed***
    - ***SAIL*** vs. ***sail***
- For sentiment analysis, MT, Information extraction
  - Case is helpful (***US*** versus ***us*** is important)

Lemmatization

- Reduce inflections or variant forms to base form
  - *am, are, is → be*
  - *car, cars, car's, cars' → car*
- *the boy's cars are different colors → the boy car be different color*
- Lemmatization: have to find correct dictionary headword form
- Machine translation
  - Spanish quiero ('I want'), quieres ('you want') same lemma as querer 'want'

Morphology

- **Morphemes**:
  - The small meaningful units that make up words
  - **Stems**: The core meaning-bearing units
  - **Affixes**: Bits and pieces that adhere to stems
    - Often with grammatical functions

Stemming

- Reduce terms to their stems in information retrieval
- *Stemming* is crude chopping of affixes
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.

for exampl compress and compress ar both accept as equival to compress

Dan Jurafsky

Porter's algorithm
The most common English stemmer

Step 1a

sses → ss	caresses	→	caress
ies → i	ponies	→	poni
ss → ss	caress	→	caress
s → ø	cats	→	cat

Step 1b

(*v*)ing → ø	walking	→	walk
	sing	→	sing
(*v*)ed → ø	plastered	→	plaster

…

Step 2 (for long stems)

ational→ ate	relational	→	relate
izer→ ize	digitizer	→	digitize
ator→ ate	operator	→	operate

…

Step 3 (for longer stems)

al → ø	revival	→	reviv
able → ø	adjustable	→	adjust
ate → ø	activate	→	activ

…

Sentence Segmentation

- !, ? are relatively unambiguous
- Period "." is quite ambiguous
  - Sentence boundary
  - Abbreviations like Inc. or Dr.
  - Numbers like .02% or 4.3
- Build a binary classifier
  - Looks at a "."
  - Decides EndOfSentence/NotEndOfSentence
  - Classifiers: hand-written rules, regular expressions, or machine-learning

Sentence Splitters

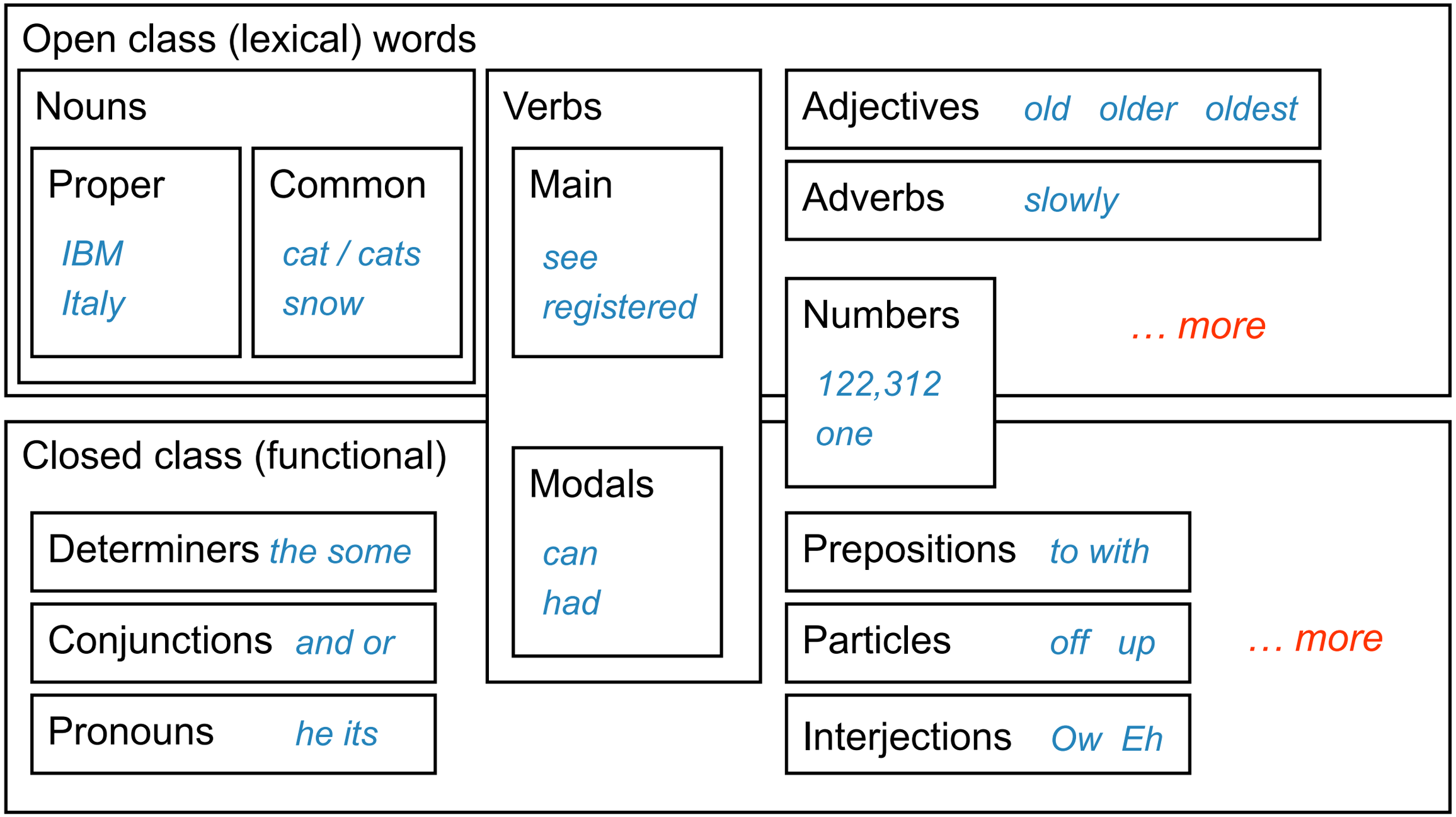
- Stanford tokenizer: http://nlp.stanford.edu/software/tokenizer.shtml
- UIUC sentence segmentation tool: https://cogcomp.cs.illinois.edu/page/tools_view/2

Syntax Processing Tasks

- Parts-of-speech Tagging
- Syntactic Parsing
- Dependency Parsing

Parts-of-speech Tagging

- Parts-of-speech: noun, verb, article, adverb, preposition, conjunction, participle, pronoun
- lexical categories
- word classes
- “tags”
- POS


Open class (lexical) words
Nouns
Proper
IBM
Italy
Common
cat / cats
snow
Verbs
Main
see
registered
Modals
can
had
Adjectives old older oldest
Adverbs slowly
Numbers
122,312
one
… more
Closed class (functional)
Determiners the some
Conjunctions and or
Pronouns he its
Prepositions to with
Particles off up
Interjections Ow Eh
… more

POS Tagging

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

POS tagging performance

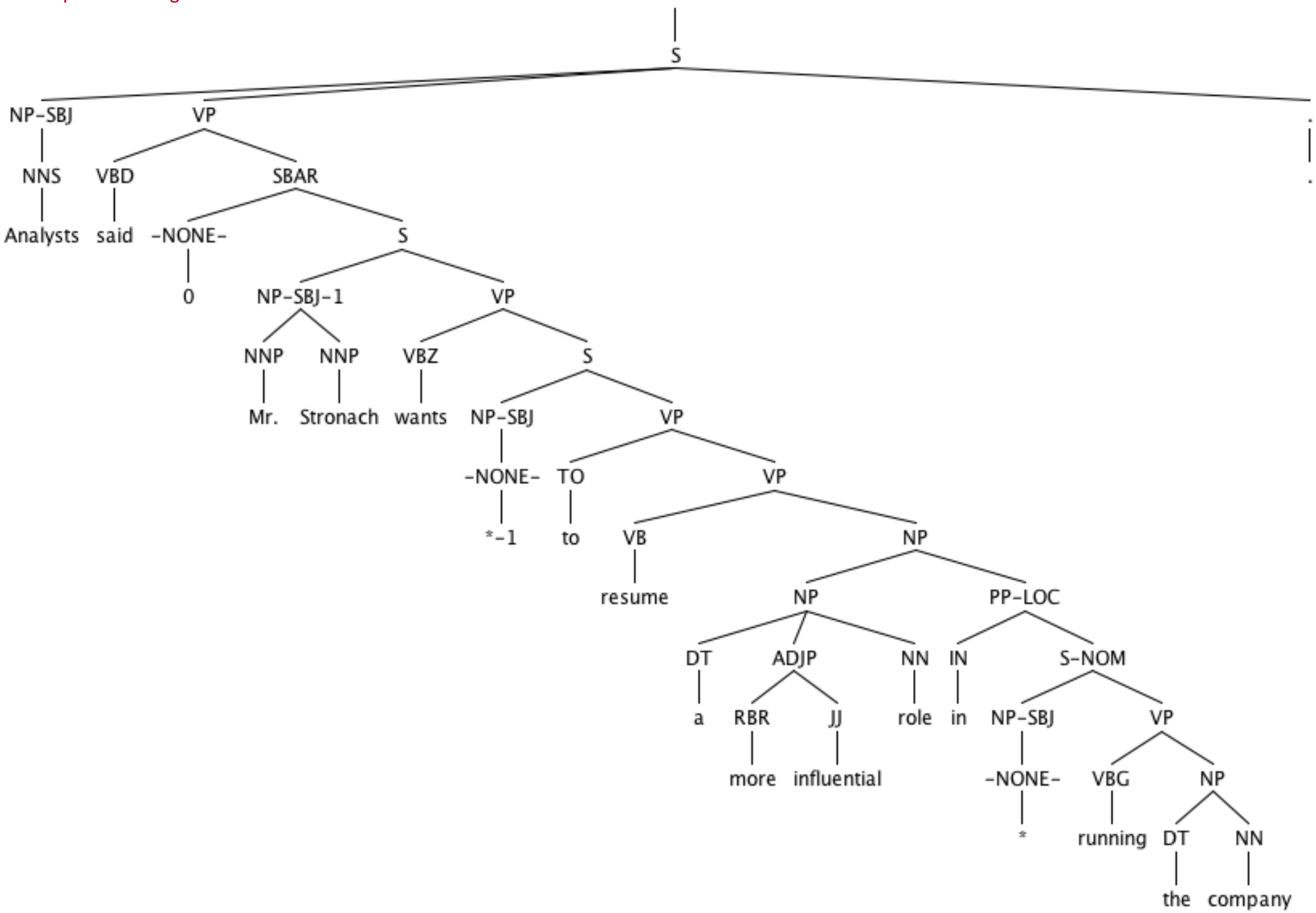
- How many tags are correct? (Tag accuracy)
  - About 97% currently
  - But baseline is already 90%
    - Baseline is performance of stupidest possible method
      - Tag every word with its most frequent tag
      - Tag unknown words as nouns
  - Partly easy because
    - Many words are unambiguous
    - You get points for them (*the*, *a*, etc.) and for punctuation marks!

Deciding on the correct part of speech can be difficult even for people

- Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG
- All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN
- Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

Syntactic Parsing

Christopher Manning



The rise of annotated data: The Penn Treebank

[Marcus et al. 1993, *Computational Linguistics*]

```
( (S
    (NP-SBJ (DT The) (NN move))
    (VP (VBD followed)
      (NP
        (NP (DT a) (NN round))
        (PP (IN of)
          (NP
            (NP (JJ similar) (NNS increases))
            (PP (IN by)
              (NP (JJ other) (NNS lenders)))
            (PP (IN against)
              (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans))))))
      (, ,)
      (S-ADV
        (NP-SBJ (-NONE- *))
        (VP (VBG reflecting)
          (NP
            (NP (DT a) (VBG continuing) (NN decline))
            (PP-LOC (IN in)
              (NP (DT that) (NN market)))))))
    (. .)))
```

Attachment Ambiguities

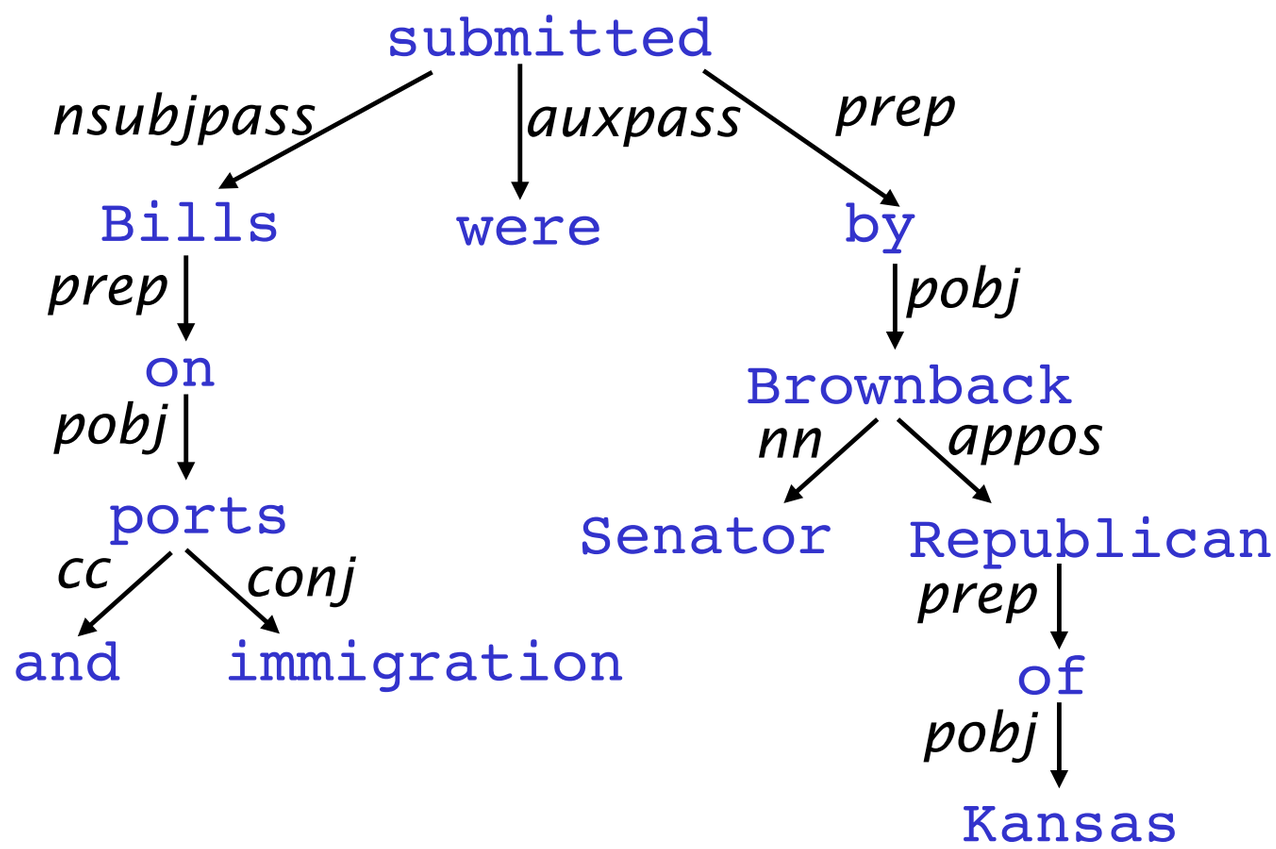
- Prepositional phrases, coordinations, etc.
- Example: She saw the man with a telescope.
- Example: I eat steak with a knife.
- Example: I eat steak with wine.

Dependency Grammar and Dependency Structure

Dependency syntax postulates that syntactic structure consists of lexical items linked by binary asymmetric relations ("arrows") called dependencies

The arrow connects a head (governor, superior, regent) with a dependent (modifier, inferior, subordinate)

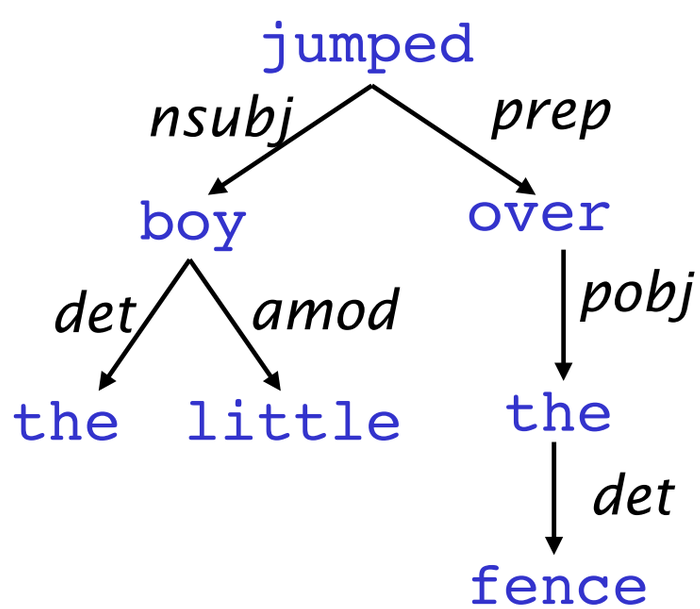
Usually, dependencies form a tree (connected, acyclic, single-head)



Stanford Dependencies

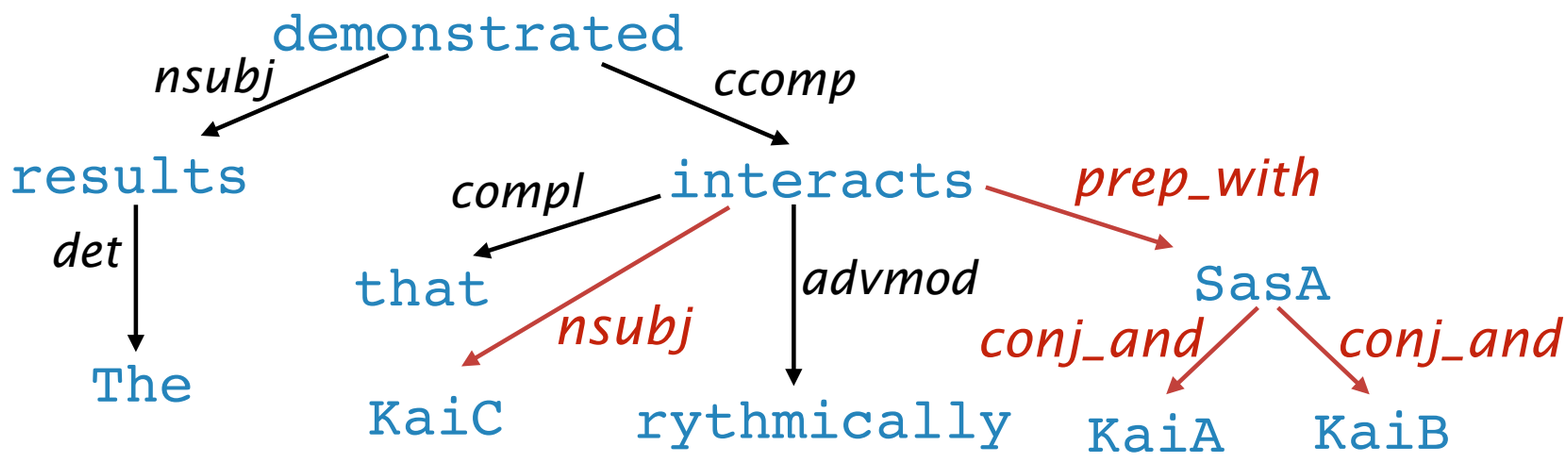
[de Marneffe et al. LREC 2006]

- The basic dependency representation is projective
- It can be generated by postprocessing headed phrase structure parses (Penn Treebank syntax)
- It can also be generated directly by dependency parsers, such as MaltParser, or the Easy-First Parser



Dependency paths identify relations like protein interaction

[Erkan et al. EMNLP 07, Fundel et al. 2007]

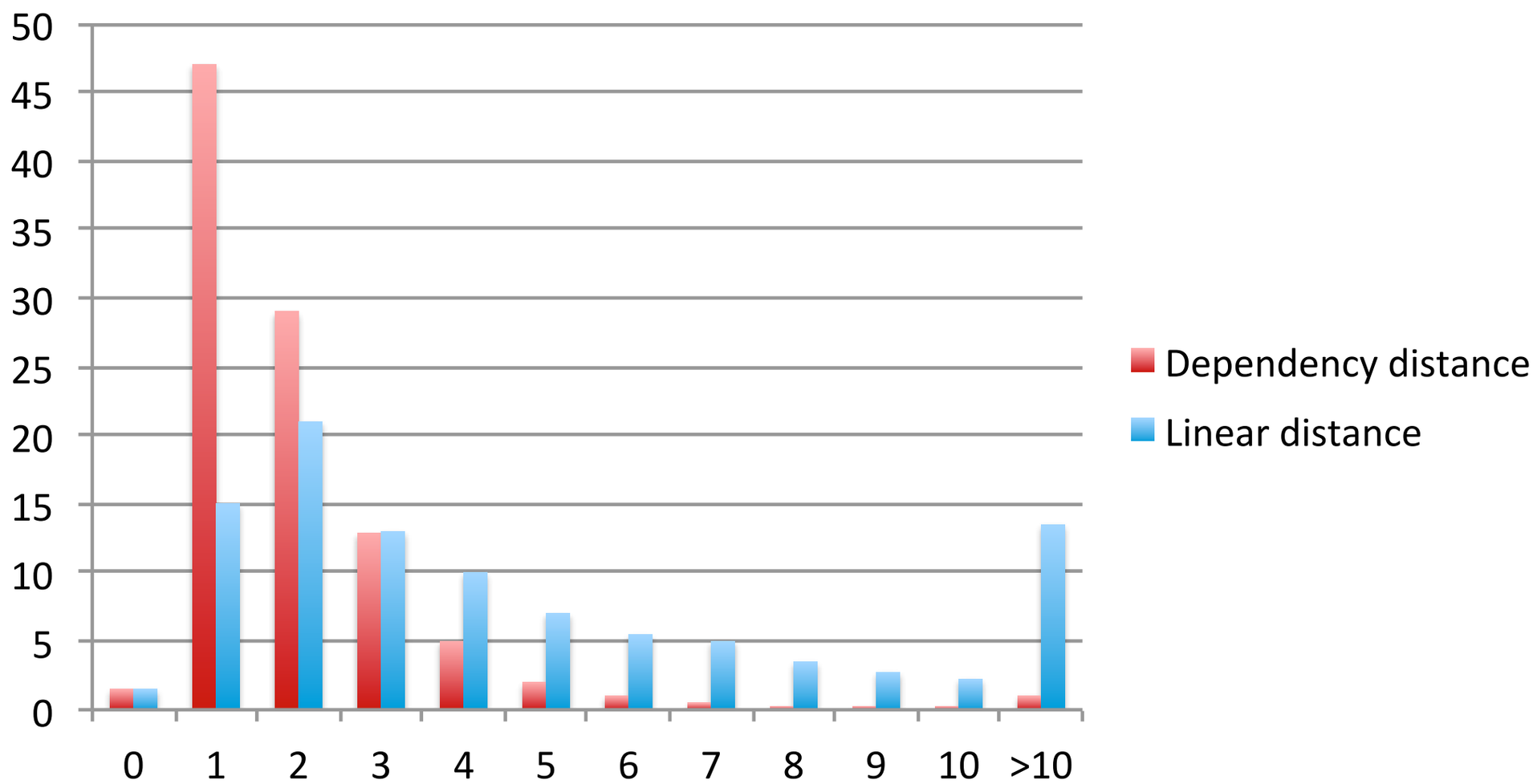


KaiC ←nsubj interacts prep_with→ SasA

KaiC ←nsubj interacts prep_with→ SasA conj_and→ KaiA

KaiC ←nsubj interacts prep_with→ SasA conj_and→ KaiB

BioNLP 2009/2011 relation extraction shared tasks [Björne et al. 2009]



Semantic Processing Tasks

- semantic similarity (at different levels: word, phrase, sentence)
- Entailment inference and paraphrasing
- Semantic role labeling (seen last time)
- Information extraction (seen last time)

Word Semantics

- Polysemy
- Synonym
- Antonym
- Hypernym
- Hyponym

Lemmas have senses

- One lemma "bank" can have many meanings:

Sense 1:
- …a **bank**[1] can hold the investments in a custodial account…

Sense 2:
- "…as agriculture burgeons on the east **bank**[2] the river will shrink even more"

- **Sense** (or **word sense**)
  - A discrete representation of an aspect of a word's meaning.
- The lemma **bank** here has two senses

Polysemy

- 1. The **bank** was constructed in 1875 out of local red brick.
- 2. I withdrew the money from the **bank**
- Are those the same sense?
  - Sense 2: "A financial institution"
  - Sense 1: "The building belonging to a financial institution"
- A **polysemous** word has **related** meanings
  - Most non-rare words have multiple meanings

Synonyms

- Word that have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H_2O
- Two lexemes are synonyms
  - if they can be substituted for each other in all situations
  - If so they have the same **propositional meaning**

Dan Jurafsky

Synonymy is a relation between senses rather than words

- Consider the words *big* and *large*
- Are they synonyms?
  - How **big** is that plane?
  - Would I be flying on a **large** or small plane?
- How about here:
  - Miss Nelson became a kind of **big** sister to Benjamin.
  - ?Miss Nelson became a kind of **large** sister to Benjamin.
- Why?
  - *big* has a sense that means being older, or grown up
  - *large* lacks this sense

Synonyms

- But there are few (or no) examples of perfect synonymy.
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.
- Example:
  - Water/$H_2$0
  - Big/large
  - Brave/courageous

Antonyms

- Senses that are opposites with respect to one feature of meaning
- Otherwise, they are very similar!

  `dark/light` `short/long` `fast/slow` `rise/fall`
  `hot/cold` `up/down` `in/out`
- More formally: antonyms can
  - define a binary opposition
    or be at opposite ends of a scale
    - `long/short, fast/slow`
  - Be **reversives**:
    - `rise/fall, up/down`

Hyponymy and Hypernymy

- One sense is a **hyponym** of another if the first sense is more specific, denoting a subclass of the other
  - *car* is a hyponym of *vehicle*
  - *mango* is a hyponym of *fruit*
- Conversely **hypernym/superordinate** ("hyper is super")
  - *vehicle* is a **hypernym** of *car*
  - *fruit* is a hypernym of *mango*

Superordinate/hyper	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

WordNet 3.0

- Where it is:
  - http://wordnetweb.princeton.edu/perl/webwn
- Libraries
  - Python: WordNet from NLTK
    - http://www.nltk.org/Home
  - Java:
    - JWNL, extJWNL on sourceforge

WordNet 3.0

- A hierarchically organized lexical database
- On-line thesaurus + aspects of a dictionary
  - Some other languages available or under development
    - (Arabic, Finnish, German, Portuguese…)

Category	Unique Strings
Noun	117,798
Verb	11,529
Adjective	22,479
Adverb	4,481

How is "sense" defined in WordNet?

- **The synset (synonym set),** the set of near-synonyms, instantiates a sense or concept, with a gloss
- Example: chump as a noun with the gloss:

  "a person who is gullible and easy to take advantage of"
- This sense of "chump" is shared by 9 words:

  `chump`^{1}, `fool`^{2}, `gull`^{1}, `mark`^{9}, `patsy`^{1}, `fall guy`^{1}, `sucker`^{1}, `soft touch`^{1}, `mug`^{2}
- Each of **these** senses have this same gloss
  - (Not **every** sense; sense 2 of gull is the aquatic bird)

Uses of the MeSH Ontology

- Provide synonyms (“entry terms”)
  - E.g., glucose and dextrose
- Provide hypernyms (from the hierarchy)
  - E.g., glucose ISA monosaccharide
- Indexing in MEDLINE/PubMED database
  - NLM’s bibliographic database:
    - 20 million journal articles
    - Each article hand-assigned 10-20 MeSH terms

Word Similarity

- **Synonymy**: a binary relation
  - Two words are either synonymous or not
- **Similarity** (or **distance**): a looser metric
  - Two words are more similar if they share more features of meaning
- Similarity is properly a relation between **senses**
  - The word "`bank`" is not similar to the word "`slope`"
  - Bank[1] is similar to fund[3]
  - Bank[2] is similar to slope[5]
- But we'll compute similarity over both words and senses

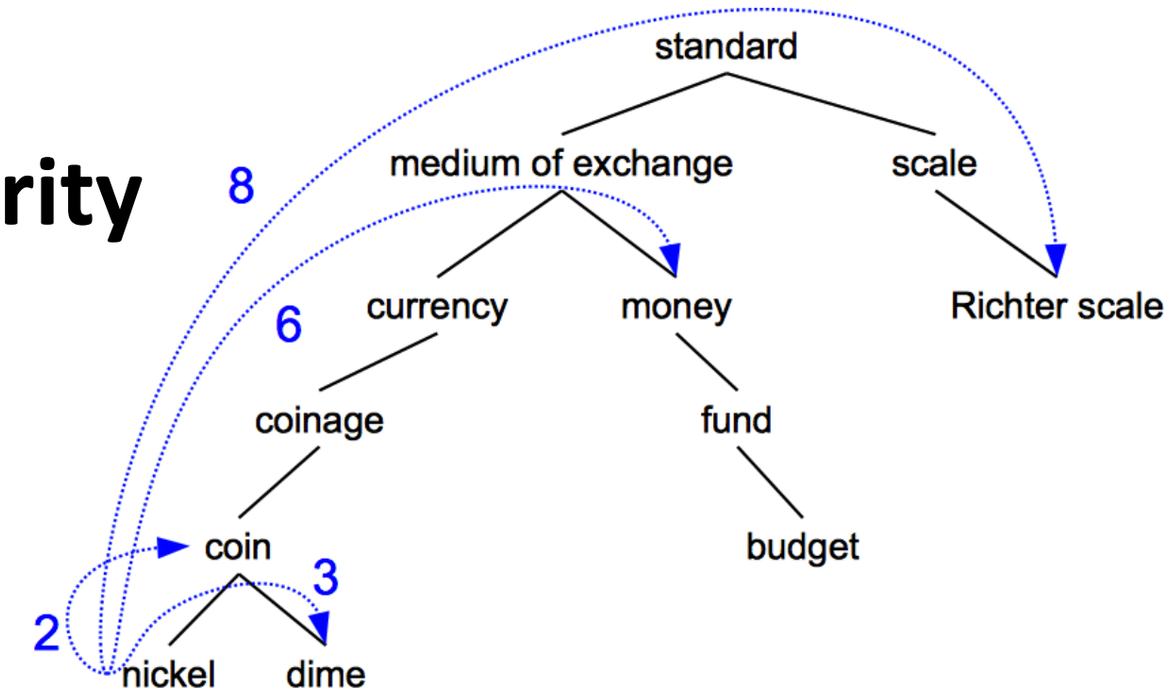
Word similarity and word relatedness

- We often distinguish **word similarity** from **word relatedness**
  - **Similar words**: near-synonyms
  - **Related words**: can be related any way
    - `car, bicycle:` **similar**
    - `car, gasoline:` **related**, not similar

Two classes of similarity algorithms

- Thesaurus-based algorithms
  - Are words "nearby" in hypernym hierarchy?
  - Do words have similar glosses (definitions)?
- Distributional algorithms
  - Do words have similar distributional contexts?

Path based similarity



- Two concepts (senses/synsets) are similar if they are near each other in the thesaurus hierarchy
  - =have a short path between them
  - concepts have path 1 to themselves

Problems with thesaurus-based meaning

- We don't have a thesaurus for every language
- Even if we do, they have problems with **recall**
  - Many words are missing
  - Most (if not all) phrases are missing
  - Some connections between senses are missing
  - Thesauri work less well for verbs, adjectives
    - Adjectives and verbs have less structured hyponymy relations

Distributional models of meaning

- Also called vector-space models of meaning
- Offer much higher recall than hand-built thesauri
  - Although they tend to have lower precision
- Zellig Harris (1954): "**oculist** and **eye-doctor** … occur in almost the same environments…. **If A and B have almost identical environments we say that they are synonyms**.
- Firth (1957): "You shall know a word by the company it keeps!"

Intuition of distributional word similarity

- Nida example:

  A bottle of ***tesgüino*** is on the table
  Everybody likes ***tesgüino***
  Tesgüino makes you drunk
  We make ***tesgüino*** out of corn.

- From context words humans can guess ***tesgüino*** means
  - an alcoholic beverage like **beer**
- Intuition for algorithm:
  - Two words are similar if they have similar word contexts.

Reminder: Term-document matrix

- Each cell: count of term *t* in a document *d*: $tf_{t,d}$:
  - Each document is a count vector in $\mathbb{N}^v$: a column below

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

The Term-Context matrix

- Instead of using entire documents, use smaller contexts
  - Paragraph
  - Window of 10 words
- A word is now defined by a vector over counts of context words

Sample contexts: 20 words (Brown corpus)

- equal amount of sugar, a sliced lemon, a tablespoonful of **apricot** preserve or jam, a pinch each of clove and nutmeg,
- on board for their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened to that of
- of a recursive type well suited to programming on the **digital** computer. In finding the optimal R-stage policy from that of
- substantially affect commerce, for the purpose of gathering data and **information** necessary for the study authorized in the first section of this

Term-context matrix for word similarity

- Two **words** are similar in meaning if their context vectors are similar

	aardvark	computer	data	pinch	result	sugar	…
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

Should we use raw counts?

- For the term-document matrix
  - We used tf-idf instead of raw term counts
- For the term-context matrix
  - Positive Pointwise Mutual Information (PPMI) is common

The third class of similarity algorithms

- Distributed similarities, based on word embeddings
- No counting, but predicting
- word2vec, glove
- Dense vectors, capture many linguistic regularies.
- vector('Paris') - vector('France') + vector('Italy') => vector('Rome')
- vector('king') - vector('man') + vector('woman') => vector('queen')

Entailment, Paraphrase, Hyponymy & Synonymy

- The relationship between **entailment and paraphrase** is **parallel** to the relationship **between hyponymy and synonymy**

- Synonymy is **symmetric (i.e. two-way)** hyponymy
- paraphrase is **symmetric (i.e. two-way)** entailment

113

Entailment v.s Paraphrasing

- beat (TeamA, TeamB) => play (TeamA, TeamB)
- (4) X wrote Y.
- (5) Y was written by X.
- (6) X is the writer of Y.