

# Natural Language Processing

Simone Teufel

Computer Laboratory  
University of Cambridge

January 2012

Lecture Materials created by Ann Copestake

## NLP and linguistics

NLP: the computational modelling of human language.

1. **Morphology** — the structure of words: lecture 2.
2. **Syntax** — the way words are used to form phrases: lectures 3, 4 and 5.
3. **Semantics**
  - ▶ **Compositional semantics** — the construction of meaning based on syntax: lecture 6.
  - ▶ **Lexical semantics** — the meaning of individual words: lecture 6.
4. **Pragmatics** — meaning in context: lecture 7.

## Outline of today's lecture

### Lecture 1: Introduction

- Overview of the course
- Why NLP is hard
- Scope of NLP
- A sample application: sentiment classification
- More NLP applications
- NLP components

## Also note:

- ▶ Exercises: pre-lecture and post-lecture
- ▶ Glossary
- ▶ **Recommended Book:** Jurafsky and Martin (2008).

## Querying a knowledge base

### User query:

- ▶ Has my order number 4291 been shipped yet?

### Database:

#### ORDER

Order number	Date ordered	Date shipped
4290	2/2/09	2/2/09
4291	2/2/09	2/2/09
4292	2/2/09	

**USER:** Has my order number 4291 been shipped yet?

**DB QUERY:** order(number=4291,date\_shipped=?)

**RESPONSE:** Order number 4291 was shipped on 2/2/09

## Why is this difficult?

Similar strings mean different things, different strings mean the same thing:

1. How fast is the TZ?
2. How fast will my TZ arrive?
3. Please tell me when I can expect the TZ I ordered.

Ambiguity:

- ▶ Do you sell Sony laptops and disk drives?
- ▶ Do you sell (Sony (laptops and disk drives))?
- ▶ Do you sell (Sony laptops) and disk drives?

## Why is this difficult?

Similar strings mean different things, different strings mean the same thing:

1. How fast is the TZ?
2. How fast will my TZ arrive?
3. Please tell me when I can expect the TZ I ordered.

Ambiguity:

- ▶ Do you sell Sony laptops and disk drives?
- ▶ Do you sell (Sony (laptops and disk drives))?
- ▶ Do you sell (Sony laptops) and disk drives?

## Why is this difficult?

Similar strings mean different things, different strings mean the same thing:

1. How fast is the TZ?
2. How fast will my TZ arrive?
3. Please tell me when I can expect the TZ I ordered.

Ambiguity:

- ▶ Do you sell Sony laptops and disk drives?
- ▶ Do you sell (Sony (laptops and disk drives))?
- ▶ Do you sell (Sony laptops) and disk drives?

## Why is this difficult?

Similar strings mean different things, different strings mean the same thing:

1. How fast is the TZ?
2. How fast will my TZ arrive?
3. Please tell me when I can expect the TZ I ordered.

Ambiguity:

- ▶ Do you sell Sony laptops and disk drives?
- ▶ Do you sell (Sony (laptops and disk drives))?
- ▶ Do you sell (Sony laptops) and disk drives?

## Why is this difficult?

Similar strings mean different things, different strings mean the same thing:

1. How fast is the TZ?
2. How fast will my TZ arrive?
3. Please tell me when I can expect the TZ I ordered.

Ambiguity:

- ▶ Do you sell Sony laptops and disk drives?
- ▶ Do you sell (Sony (laptops and disk drives))?
- ▶ Do you sell (Sony laptops) and disk drives?

## Why is this difficult?

Similar strings mean different things, different strings mean the same thing:

1. How fast is the TZ?
2. How fast will my TZ arrive?
3. Please tell me when I can expect the TZ I ordered.

Ambiguity:

- ▶ Do you sell Sony laptops and disk drives?
- ▶ Do you sell (Sony (laptops and disk drives))?
- ▶ Do you sell (Sony laptops) and disk drives?

## Wouldn't it be better if ... ?

The properties which make natural language difficult to process are essential to human communication:

- ▶ Flexible
- ▶ Learnable but compact
- ▶ Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- ▶ Ambiguity is mostly local (for humans)
- ▶ Semi-formal additions and conventions for different genres

## Wouldn't it be better if ... ?

The properties which make natural language difficult to process are essential to human communication:

- ▶ Flexible
- ▶ Learnable but compact
- ▶ Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- ▶ Ambiguity is mostly local (for humans)
- ▶ Semi-formal additions and conventions for different genres

## Some NLP applications

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>▶ spelling and grammar checking</li> <li>▶ optical character recognition (OCR)</li> <li>▶ screen readers</li> <li>▶ augmentative and alternative communication</li> <li>▶ machine aided translation</li> <li>▶ lexicographers' tools</li> <li>▶ information retrieval</li> <li>▶ document classification</li> <li>▶ document clustering</li> </ul> | <ul style="list-style-type: none"> <li>▶ information extraction</li> <li>▶ question answering</li> <li>▶ summarization</li> <li>▶ text segmentation</li> <li>▶ exam marking</li> <li>▶ report generation</li> <li>▶ machine translation</li> <li>▶ natural language interfaces to databases</li> <li>▶ email understanding</li> <li>▶ dialogue systems</li> </ul> |
|---|---|

## Sentiment classification: finding out what people think about you

- ▶ Task: scan documents for positive and negative opinions on people, products etc.
- ▶ Find all references to entity in some document collection: list as positive, negative (possibly with strength) or neutral.
- ▶ Summaries plus text snippets.
- ▶ Fine-grained classification: e.g., for phone, opinions about: overall design, keypad, camera.
- ▶ Still often done by humans ...

## Motorola KRZR (from the Guardian)

*Motorola has struggled to come up with a worthy successor to the RAZR, arguably the most influential phone of the past few years. Its latest attempt is the KRZR, which has the same clamshell design but has some additional features. It has a striking blue finish on the front and the back of the handset is very tactile brushed rubber. Like its predecessors, the KRZR has a laser-etched keypad, but in this instance Motorola has included ridges to make it easier to use. ... Overall there's not much to dislike about the phone, but its slightly quirky design means that it probably won't be as huge or as hot as the RAZR.*

## Sentiment classification: the research task

- ▶ Full task: information retrieval, cleaning up text structure, named entity recognition, identification of relevant parts of text. Evaluation by humans.
- ▶ Research task: preclassified documents, topic known, opinion in text along with some straightforwardly extractable score.
- ▶ Movie review **corpus**, with ratings.

## Bag of words technique

- ▶ Treat the reviews as collections of individual words.
- ▶ Classify reviews according to positive or negative words.
- ▶ Could use word lists prepared by humans, but machine learning based on a portion of the corpus (**training set**) is preferable.
- ▶ Use star rankings for training and evaluation.
- ▶ Pang et al, 2002: Chance success is 50% (movie database was artificially balanced), bag-of-words gives 80%.

## IMDb: An American Werewolf in London (1981)

Rating: 9/10

Ooooo. Scary.

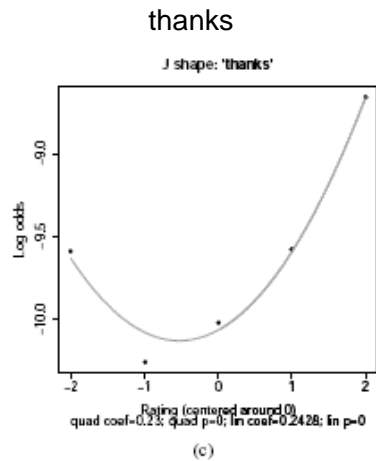
*The old adage of the simplest ideas being the best is once again demonstrated in this, one of the most entertaining films of the early 80's, and almost certainly Jon Landis' best work to date. The script is light and witty, the visuals are great and the atmosphere is top class. Plus there are some great freeze-frame moments to enjoy again and again. Not forgetting, of course, the great transformation scene which still impresses to this day.*

*In Summary: Top banana*

## Sentiment words

thanks

## Sentiment words



from Potts and Schwarz (2008)

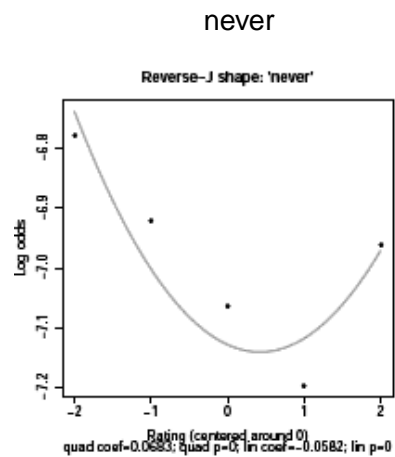


## Sentiment words

never



## Sentiment words



from Potts and Schwarz (2008)

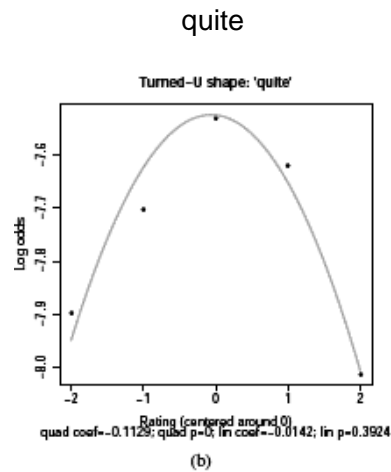


## Sentiment words

quite



## Sentiment words



from Potts and Schwarz (2008)

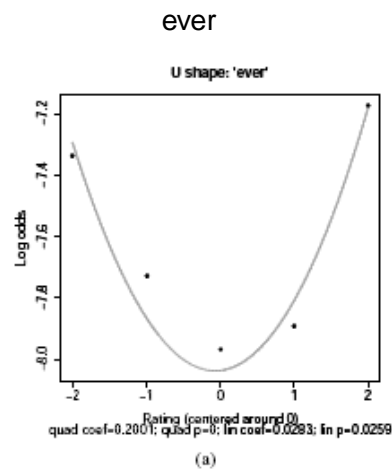


## Sentiment words: ever

ever



## Sentiment words: ever



from Potts and Schwarz (2008)



## Some sources of errors for bag-of-words

- ▶ Negation:
  - Ridley Scott has never directed a bad film.*
- ▶ Overfitting the training data:
  - e.g., if training set includes a lot of films from before 2005, *Ridley* may be a strong positive indicator, but then we test on reviews for 'Kingdom of Heaven'?
- ▶ Comparisons and contrasts.



## Contrasts in the discourse

*This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up.*

## Sample data

<http://www.cl.cam.ac.uk/~sht25/sentiment/>  
(linked from  
<http://www.cl.cam.ac.uk/~sht25/stuff.html>)  
See test data texts in:  
<http://www.cl.cam.ac.uk/~sht25/sentiment/test/>  
classified into positive/negative.

## More contrasts

*AN AMERICAN WEREWOLF IN PARIS is a failed attempt . . . Julie Delpy is far too good for this movie. She imbues Serafine with spirit, spunk, and humanity. This isn't necessarily a good thing, since it prevents us from relaxing and enjoying AN AMERICAN WEREWOLF IN PARIS as a completely mindless, campy entertainment experience. Delpy's injection of class into an otherwise classless production raises the specter of what this film could have been with a better script and a better cast . . . She was radiant, charismatic, and effective . . .*

## Doing sentiment classification 'properly'?

- ▶ Morphology, syntax and compositional semantics: who is talking about what, what terms are associated with what, tense . . .
- ▶ Lexical semantics: are words positive or negative **in this context**? Word senses (e.g., *spirit*)?
- ▶ Pragmatics and discourse structure: what is the topic of this section of text? Pronouns and definite references.
- ▶ But getting all this to work well on arbitrary text is very hard.
- ▶ Ultimately the problem is **AI-complete**, but can we do well enough for NLP to be useful?



## Doing sentiment classification ‘properly’?

- ▶ Morphology, syntax and compositional semantics: who is talking about what, what terms are associated with what, tense ...
- ▶ Lexical semantics: are words positive or negative **in this context**? Word senses (e.g., *spirit*)?
- ▶ Pragmatics and discourse structure: what is the topic of this section of text? Pronouns and definite references.
- ▶ But getting all this to work well on arbitrary text is very hard.
- ▶ Ultimately the problem is **AI-complete**, but can we do well enough for NLP to be useful?

## IR, IE and QA

- ▶ Information retrieval: return documents in response to a user query (Internet Search is a special case)
- ▶ Information extraction: discover specific information from a set of documents (e.g. company joint ventures)
- ▶ Question answering: answer a specific user question by returning a section of a document:

## What is the capital of France?

Paris has been the French capital for many centuries.

Much more about these in the IR course.

## MT

- ▶ Earliest attempted NLP application
- ▶ Quality depends on restricting the **domain**
- ▶ Utility greatly increased with increase in availability of electronic text
- ▶ Good applications for bad MT ...
- ▶ Spoken language translation is viable for limited domains

## Human translation?



## Human translation?



I am not in the office at the moment. Please send any work to be translated.

## Natural language interfaces and dialogue systems

All rely on a limited domain:

- ▶ LUNAR: classic example of a natural language interface to a database (NLID): 1970–1975
- ▶ SHRDLU: (text-based) dialogue system: 1973
- ▶ Current spoken dialogue systems

Limited domain allows disambiguation: e.g., in LUNAR, *rock* had one sense.

## Siri Example Dialogues

**Man (jogging):** Move my meeting with Kelly Altek to 12.

**Siri:** You already have a meeting about budgets at 12. Shall I schedule it anyway?

**Man:** Move it to 2. . . . Play my running mix.

\*\*\*

**Woman:** I am locked out.

**Siri:** I found three locksmiths fairly close to you (shows them on screen)

## Siri Example Dialogues

**Man:** What does my day look like?

**Siri:** Not too bad, only two meetings (shows them on screen)

\*\*\*

**Woman:** Do I need an umbrella tonight?

**Siri:** There is no rain in the forecast for tonight.

\*\*\*

And more requests to Siri:

**Man:** How do I tie a bowtie again?

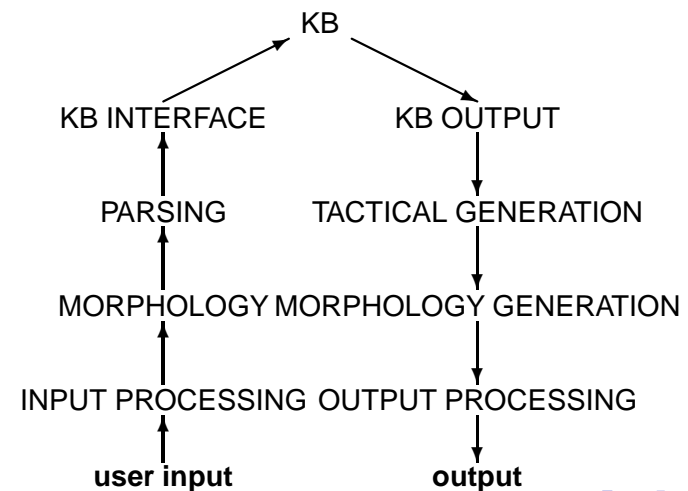
**Child:** What does a weasel look like?

**Woman:** We have a flat tire.

## Generic NLP modules

- ▶ input preprocessing: speech recogniser, text preprocessor or gesture recogniser.
- ▶ morphological analysis
- ▶ part of speech tagging
- ▶ parsing: this includes syntax and compositional semantics
- ▶ disambiguation
- ▶ context module
- ▶ text planning
- ▶ tactical generation
- ▶ morphological generation
- ▶ output processing: text-to-speech, text formatter, etc.

## Natural language interface to a knowledge base



## General comments

- ▶ Even 'simple' applications might need complex knowledge sources
- ▶ Applications cannot be 100% perfect
- ▶ Applications that are < 100% perfect can be useful
- ▶ Aids to humans are easier than replacements for humans
- ▶ NLP interfaces compete with non-language approaches
- ▶ Shallow processing on arbitrary input or deep processing on narrow domains
- ▶ Limited domain systems require extensive and expensive expertise to port

## Outline of the next lecture

### Lecture 2: Morphology and finite state techniques

A brief introduction to morphology

Using morphology

Spelling rules

Finite state techniques

More applications for finite state techniques

## Outline of today's lecture

### Lecture 2: Morphology and finite state techniques

A brief introduction to morphology

Using morphology

Spelling rules

Finite state techniques

More applications for finite state techniques

## Inflectional morphology

- ▶ e.g., plural suffix +s, past participle +ed
- ▶ sets slots in some **paradigm**
- ▶ e.g., tense, aspect, number, person, gender, case
- ▶ inflectional affixes are not combined in English
- ▶ generally fully productive (modulo irregular forms)

## Some terminology

- ▶ **morpheme**: the minimal information carrying unit
- ▶ **affix**: morpheme which only occurs in conjunction with other morphemes
- ▶ words are made up of a **stem** (more than one in the case of compounds) and zero or more affixes. e.g., *dog* plus plural suffix +s
- ▶ affixes: prefixes, suffixes, infixes and circumfixes
- ▶ in English: prefixes and suffixes (prefixes only for **derivational morphology**)
- ▶ **productivity**: whether affix applies generally, whether it applies to new words

## Derivational morphology

- ▶ e.g., *un-*, *re-*, *anti-*, *-ism*, *-ist* etc
- ▶ broad range of semantic possibilities, may change part of speech
- ▶ indefinite combinations  
e.g., *antiantidisestablishmentarianism*  
*anti-anti-dis-establish-ment-arian-ism*
- ▶ The case of **flammable**
- ▶ generally semi-productive
- ▶ zero-derivation (e.g. *tango*, *waltz*)

## Internal structure and ambiguity

**Morpheme ambiguity:** stems and affixes may be individually ambiguous: e.g. *dog* (noun or verb), +s (plural or 3persg-verb)

**Structural ambiguity:** e.g., *shorts/short -s*  
*unionised* could be *union -ise -ed* or *un- ion -ise -ed*

## Bracketing:

- ▶ *un-ion* is not a possible form
- ▶ *un-* is ambiguous:
  - ▶ with verbs: means 'reversal' (e.g., *untie*)
  - ▶ with adjectives: means 'not' (e.g., *unwise*)
- ▶ internal structure of *un-ion-ise-ed* has to be (*un- ((ion-ise) -ed)*)

## Internal structure and ambiguity

**Morpheme ambiguity:** stems and affixes may be individually ambiguous: e.g. *dog* (noun or verb), +s (plural or 3persg-verb)

**Structural ambiguity:** e.g., *shorts/short -s*  
*unionised* could be *union -ise -ed* or *un- ion -ise -ed*

## Bracketing:

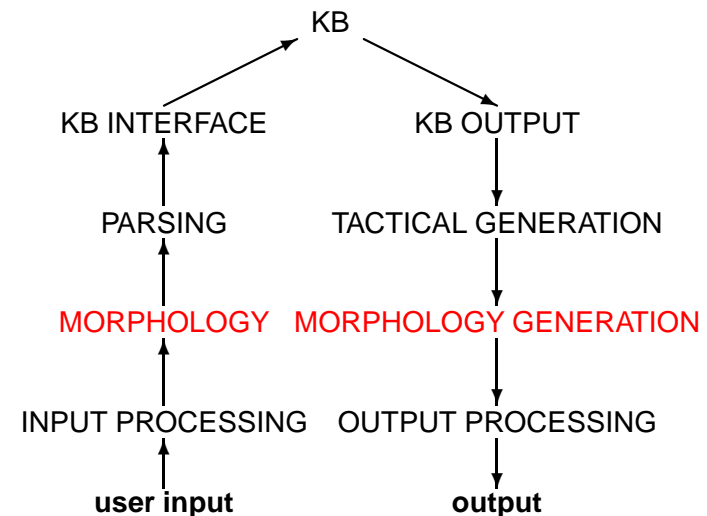
- ▶ *un- ion* is not a possible form
- ▶ *un-* is ambiguous:
  - ▶ with verbs: means 'reversal' (e.g., *untie*)
  - ▶ with adjectives: means 'not' (e.g., *unwise*)
- ▶ internal structure of *un- ion -ise -ed* has to be (*un- ((ion -ise) -ed)*)

## Temporarily skip 2.3

## Applications of morphological processing

- ▶ compiling a **full-form** lexicon
- ▶ **stemming** for IR (not linguistic stem)
- ▶ **lemmatization** (often inflections only): finding stems and affixes as a precursor to parsing  
NB: may use parsing to filter results (see lecture 5)  
e.g., *feed* analysed as *fee-ed* (as well as *feed*)  
but parser blocks (assuming lexicon does not have *fee* as a verb)
- ▶ generation  
Morphological processing may be **bidirectional**: i.e., parsing and generation.  
*sleep* + PAST\_VERB <-> *slept*

## Morphology in a deep processing system (cf lec 1)



## Lexical requirements for morphological processing

- ▶ affixes, plus the associated information conveyed by the affix
  - ed PAST\_VERB
  - ed PSP\_VERB
  - s PLURAL\_NOUN
- ▶ irregular forms, with associated information similar to that for affixes
  - began PAST\_VERB begin
  - begun PSP\_VERB begin
- ▶ stems with syntactic categories (plus more)

## Mongoose

A zookeeper was ordering extra animals for his zoo. He started the letter:

*“Dear Sir, I need two mongeese.”*

This didn’t sound right, so he tried again:

*“Dear Sir, I need two mongooses.”*

But this sounded terrible too. Finally, he ended up with:

*“Dear Sir, I need a mongoose, and while you’re at it, send me another one as well.”*

## Mongoose

A zookeeper was ordering extra animals for his zoo. He started the letter:

*“Dear Sir, I need two mongeese.”*

This didn’t sound right, so he tried again:

*“Dear Sir, I need two mongooses.”*

But this sounded terrible too. Finally, he ended up with:

*“Dear Sir, I need a mongoose, and while you’re at it, send me another one as well.”*

## Mongoose

A zookeeper was ordering extra animals for his zoo. He started the letter:

*“Dear Sir, I need two mongeese.”*

This didn’t sound right, so he tried again:

*“Dear Sir, I need two mongooses.”*

But this sounded terrible too. Finally, he ended up with:

*“Dear Sir, I need a mongoose, and while you’re at it, send me another one as well.”*

## Spelling rules (sec 2.3)

- ▶ English morphology is essentially concatenative
- ▶ irregular morphology — inflectional forms have to be listed
- ▶ regular phonological and spelling changes associated with affixation, e.g.
  - ▶ -s is pronounced differently with stem ending in s, x or z
  - ▶ spelling reflects this with the addition of an e (*boxes* etc)
- ▶ in English, description is independent of particular stems/affixes

## e-insertion

e.g.  $box^s$  to *boxes*

$$\varepsilon \rightarrow e / \left\{ \begin{array}{c} s \\ x \\ z \end{array} \right\} ^ \_ s$$

- ▶ map 'underlying' form to surface form
- ▶ mapping is left of the slash, context to the right
- ▶ notation:

— position of mapping  
 $\varepsilon$  empty string  
 $\wedge$  affix boundary — stem  $\wedge$  affix

- ▶ same rule for plural and 3sg verb
- ▶ formalisable/implementable as a finite state transducer

## e-insertion

e.g.  $box^s$  to *boxes*

$$\varepsilon \rightarrow e / \left\{ \begin{array}{c} s \\ x \\ z \end{array} \right\} ^ \_ s$$

- ▶ map 'underlying' form to surface form
- ▶ mapping is left of the slash, context to the right
- ▶ notation:

— position of mapping  
 $\varepsilon$  empty string  
 $\wedge$  affix boundary — stem  $\wedge$  affix

- ▶ same rule for plural and 3sg verb
- ▶ formalisable/implementable as a finite state transducer

## e-insertion

e.g.  $box^s$  to *boxes*

$$\varepsilon \rightarrow e / \left\{ \begin{array}{c} s \\ x \\ z \end{array} \right\} ^ \_ s$$

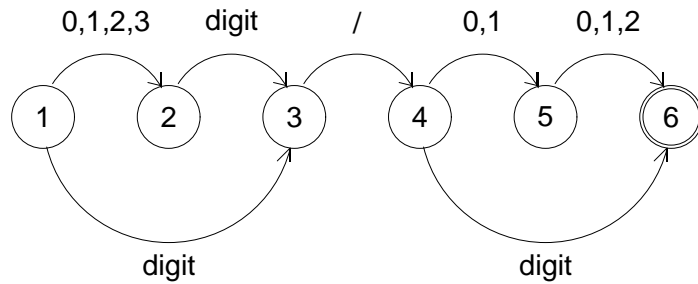
- ▶ map 'underlying' form to surface form
- ▶ mapping is left of the slash, context to the right
- ▶ notation:

— position of mapping  
 $\varepsilon$  empty string  
 $\wedge$  affix boundary — stem  $\wedge$  affix

- ▶ same rule for plural and 3sg verb
- ▶ formalisable/implementable as a finite state transducer

## Finite state automata for recognition

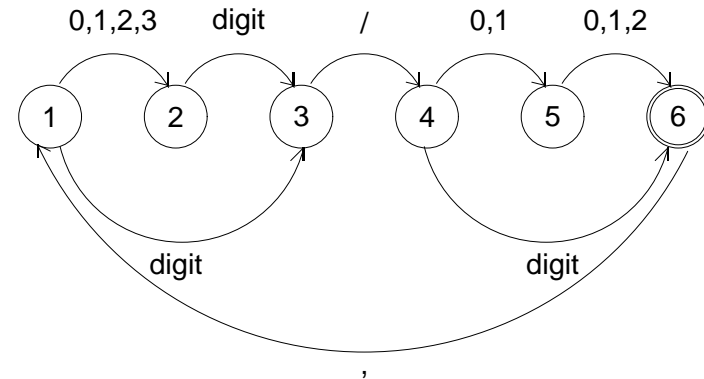
day/month pairs:



- ▶ non-deterministic — after input of '2', in state 2 and state 3.
- ▶ double circle indicates accept state
- ▶ accepts e.g., 11/3 and 3/12
- ▶ also accepts 37/00 — overgeneration

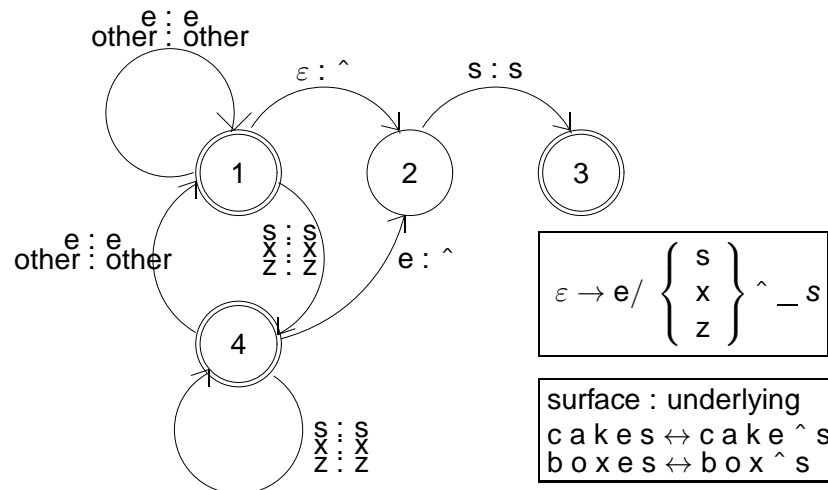
## Recursive FSA

comma-separated list of day/month pairs:

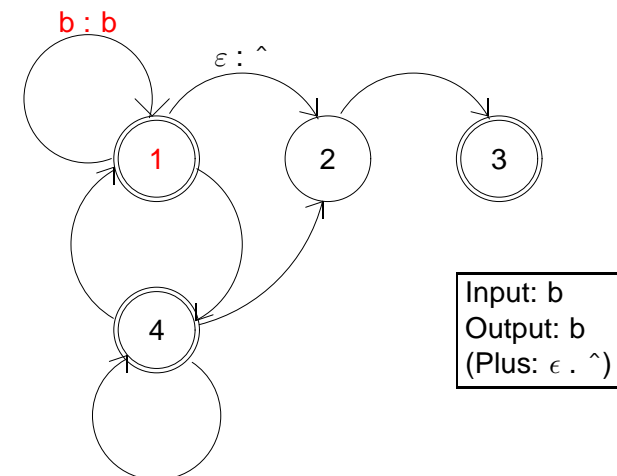


- ▶ list of indefinite length
- ▶ e.g., 11/3, 5/6, 12/04

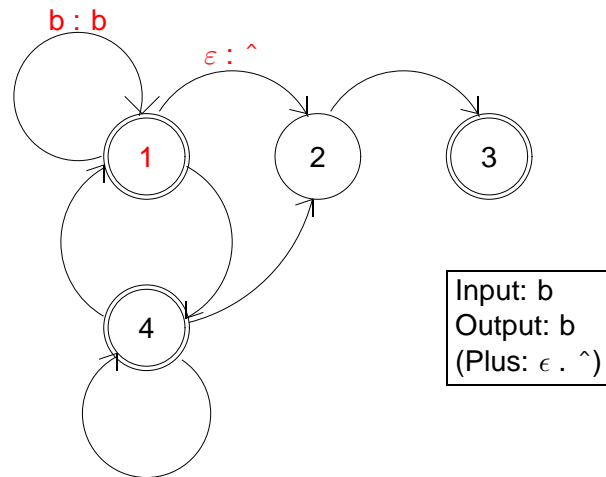
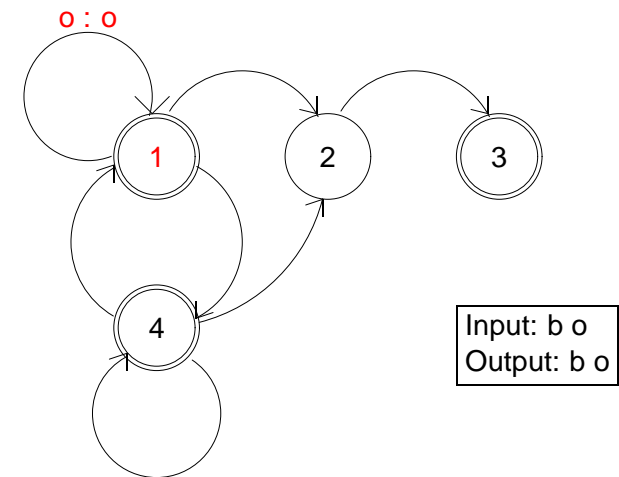
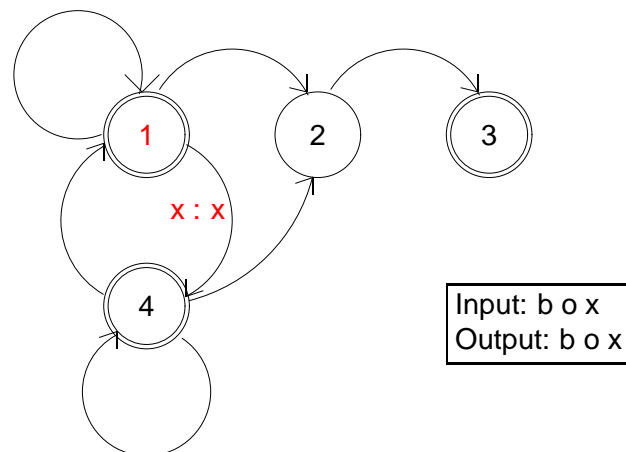
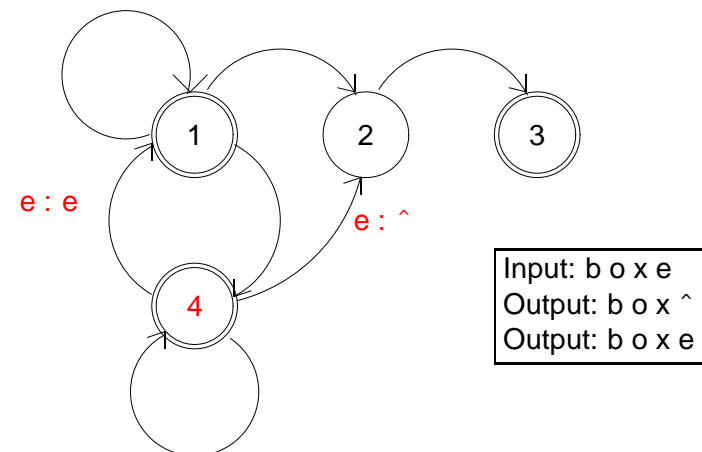
## Finite state transducer

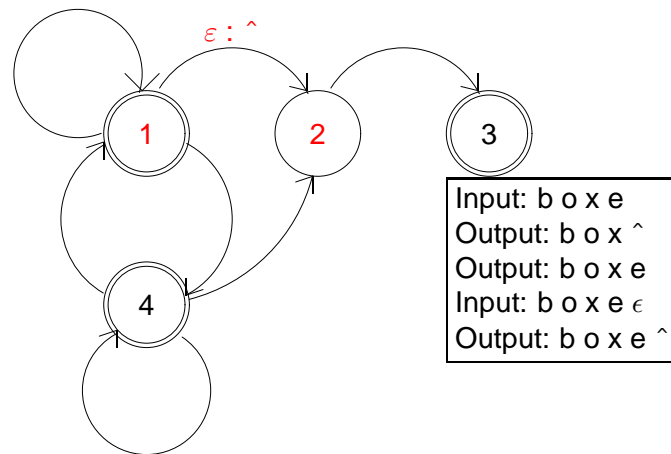
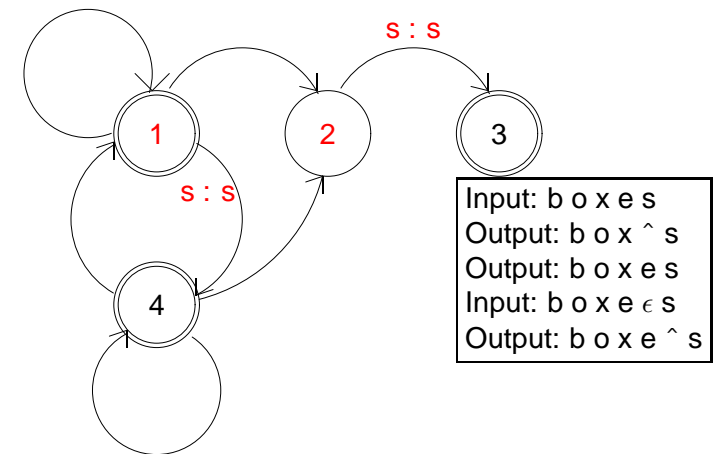
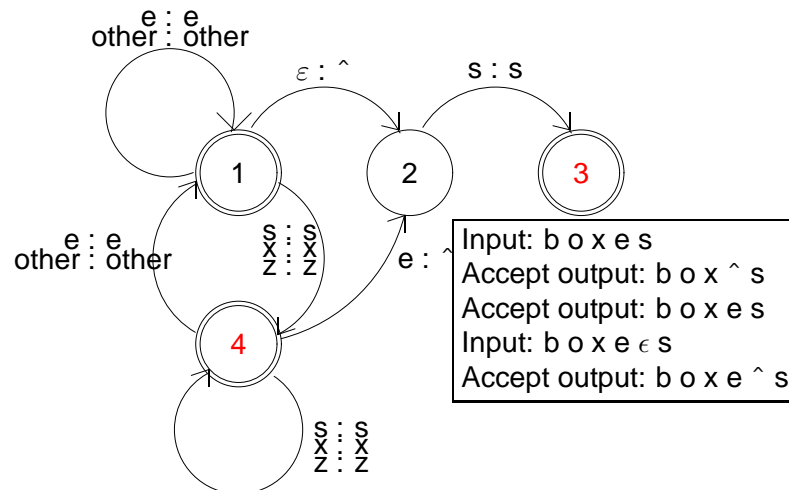


## Analysing *b o x e s*





Analysing **b** o x e sAnalysing **b** **o** x e sAnalysing **b** o **x** e sAnalysing **b** o x **e** s

Analysing *b o x e € s*Analysing *b o x e s*Analysing *b o x e s*

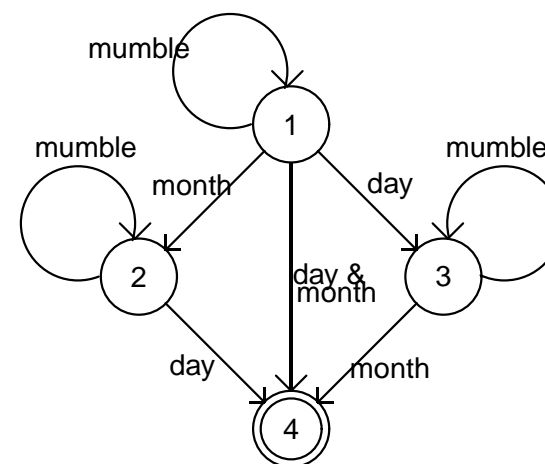
## Using FSTs

- ▶ FSTs assume **tokenization** (word boundaries) and words split into characters. One character pair per transition!
- ▶ Analysis: return character list with affix boundaries, so enabling lexical lookup.
- ▶ Generation: input comes from stem and affix lexicons.
- ▶ One FST per spelling rule: either compile to big FST or run in parallel.
- ▶ FSTs do not allow for internal structure:
  - ▶ can't model *un- ion -ize -d* bracketing.
  - ▶ can't condition on prior transitions, so potential redundancy (cf 2006/7 exam q)

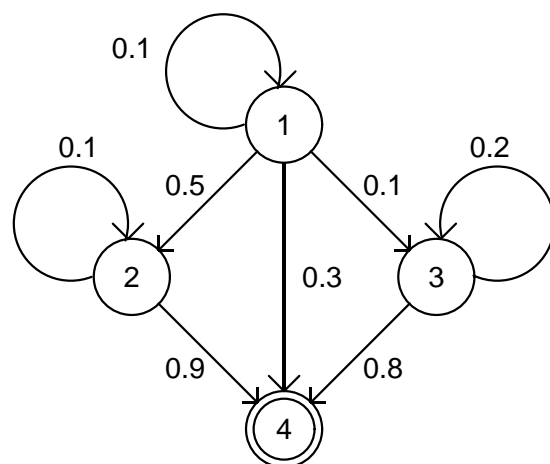
## Some other uses of finite state techniques in NLP

- ▶ Grammars for simple spoken dialogue systems (directly written or compiled)
- ▶ Partial grammars for named entity recognition
- ▶ Dialogue models for spoken dialogue systems (SDS)  
e.g. obtaining a date:
  1. No information. System prompts for month and day.
  2. Month only is known. System prompts for day.
  3. Day only is known. System prompts for month.
  4. Month and day known.

## Example FSA for dialogue



## Example of probabilistic FSA for dialogue



## Next lecture

### Lecture 3: Prediction and part-of-speech tagging

Corpora in NLP

Word prediction

Part-of-speech (POS) tagging

Evaluation in general, evaluation of POS tagging

## Outline of today's lecture

### Lecture 3: Prediction and part-of-speech tagging

Corpora in NLP

Word prediction

Part-of-speech (POS) tagging

Evaluation in general, evaluation of POS tagging

First of three lectures that concern **syntax** (i.e., how words fit together). This lecture: 'shallow' syntax: word sequences and POS tags. Next lectures: more detailed syntactic structures.

## Statistical techniques: NLP and linguistics

*But it must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term. (Chomsky 1969)*

*Whenever I fire a linguist our system performance improves. (Jelinek, 1988?)*

## Corpora

Changes in NLP research over the last 15-20 years are largely due to increased availability of electronic corpora.

- ▶ **corpus**: text that has been collected for some purpose.
- ▶ **balanced corpus**: texts representing different genres  
**genre** is a type of text (vs domain)
- ▶ **tagged corpus**: a corpus annotated with POS tags
- ▶ **treebank**: a corpus annotated with parse trees
- ▶ **specialist corpora** — e.g., collected to train or evaluate particular applications
  - ▶ Movie reviews for sentiment classification
  - ▶ Data collected from simulation of a dialogue system

## Statistical techniques: NLP and linguistics

*But it must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term. (Chomsky 1969)*

*Whenever I fire a linguist our system performance improves. (Jelinek, 1988?)*

## Prediction

Guess the missing words:

Illustrations produced by any package can be transferred with consummate \_\_\_\_\_ to another.

Wright tells her story with great \_\_\_\_\_.

## Prediction

Guess the missing words:

Illustrations produced by any package can be transferred with consummate ease to another.

Wright tells her story with great \_\_\_\_\_.

## Prediction

Guess the missing words:

Illustrations produced by any package can be transferred with consummate ease to another.

Wright tells her story with great professionalism.

## Prediction

Prediction is relevant for:

- ▶ language modelling for speech recognition to disambiguate results from signal processing: e.g., using **n-grams**. (Alternative to finite state grammars, suitable for large-scale recognition.)
- ▶ word prediction for communication aids (augmentative and alternative communication). e.g., to help enter text that's input to a synthesiser
- ▶ text entry on mobile phones and similar devices
- ▶ OCR, spelling correction, text segmentation
- ▶ estimation of entropy

## bigrams (n-gram with N=2)

A probability is assigned to a word based on the previous word:

$$P(w_n | w_{n-1})$$

where  $w_n$  is the  $n$ th word in a sentence.

Probability of a sequence of words (assuming independence):

$$P(W_1^n) \approx \prod_{k=1}^n P(w_k | w_{k-1})$$

Probability is estimated from counts in a training corpus:

$$\frac{C(w_{n-1} w_n)}{\sum_w C(w_{n-1} w)} \approx \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

i.e. count of a particular bigram in the corpus divided by the count of all bigrams starting with the prior word.

## Calculating bigrams

⟨s⟩ good morning ⟨/s⟩ ⟨s⟩ good afternoon ⟨/s⟩ ⟨s⟩ good afternoon ⟨/s⟩ ⟨s⟩ it is very good ⟨/s⟩ ⟨s⟩ it is good ⟨/s⟩

sequence	count	bigram probability
⟨s⟩	5	
⟨s⟩ good	3	.6
⟨s⟩ it	2	.4
good	5	
good morning	1	.2
good afternoon	2	.4
good ⟨/s⟩	2	.4
⟨/s⟩	5	
⟨/s⟩ ⟨s⟩	4	1

## Sentence probabilities

Probability of ⟨s⟩ it is good afternoon ⟨/s⟩ is estimated as:

$$P(it|\langle s \rangle)P(is|it)P(good|is)P(afternoon|good)P(\langle /s \rangle |afternoon) \\ = .4 \times 1 \times .5 \times .4 \times 1 = .08$$

Problems because of **sparse data** (cf Chomsky comment):

- ▶ **smoothing**: distribute 'extra' probability between rare and unseen events
- ▶ **backoff**: approximate unseen probabilities by a more general probability, e.g. unigrams

## Shakespeare, re-generated

**Unigram:** To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have. Every enter now...

**Bigram:** What means, sir. I confess she? then all sorts, he is trim, captain. What dost stand forth they canopy, forsooth; ...

**Trigram:** Sweet prince, Falstaff shall die. Harry of Monmouth's grave. This shall forbid it should be branded, if renown ...

**Quadrigram:** King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. It cannot be but so.

## Practical application

- ▶ Word prediction: guess the word from initial letters. User confirms each word, so we predict on the basis of individual bigrams consistent with letters.
- ▶ Speech recognition: given an input which is a lattice of possible words, we find the sequence with maximum likelihood.  
Implemented efficiently using dynamic programming (Viterbi algorithm).

## Part of speech tagging

They can fish .

- ▶ They\_PNP can\_VM0 fish\_VVI .\_PUN
- ▶ They\_PNP can\_VVB fish\_NN2 .\_PUN
- ▶ They\_PNP can\_VM0 fish\_NN2 .\_PUN **no full parse**

POS lexicon fragment:

```
they  PNP
can   VM0 VVB VVI NN1
fish  NN1 NN2 VVB VVI
```

tagset (CLAWS 5) includes:

NN1	singular noun	NN2	plural noun
PNP	personal pronoun	VM0	modal auxiliary verb
VVB	base form of verb	VVI	infinitive form of verb

## Part of speech tagging

- ▶ They\_PNP can\_VM0 fish\_VVI .\_PUN
- ▶ They\_PNP can\_VVB fish\_NN2 .\_PUN
- ▶ They\_PNP can\_VM0 fish\_NN2 .\_PUN **no full parse**

POS lexicon fragment:

```
they  PNP
can   VM0 VVB VVI NN1
fish  NN1 NN2 VVB VVI
```

tagset (CLAWS 5) includes:

NN1	singular noun	NN2	plural noun
PNP	personal pronoun	VM0	modal auxiliary verb
VVB	base form of verb	VVI	infinitive form of verb

## Part of speech tagging

- ▶ They\_PNP can\_VM0 fish\_VVI .\_PUN
- ▶ They\_PNP can\_VVB fish\_NN2 .\_PUN
- ▶ They\_PNP can\_VM0 fish\_NN2 .\_PUN **no full parse**

POS lexicon fragment:

```
they  PNP
can   VM0 VVB VVI NN1
fish  NN1 NN2 VVB VVI
```

tagset (CLAWS 5) includes:

NN1	singular noun	NN2	plural noun
PNP	personal pronoun	VM0	modal auxiliary verb
VVB	base form of verb	VVI	infinitive form of verb

## Why POS tag?

- ▶ Coarse-grained syntax / word sense disambiguation: fast, so applicable to very large corpora.
- ▶ Some linguistic research and lexicography: e.g., how often is *tango* used as a verb? *dog*?
- ▶ Named entity recognition and similar tasks (finite state patterns over POS tagged data).
- ▶ Features for machine learning e.g., sentiment classification. (e.g., *stink\_V* vs *stink\_N*)
- ▶ Preliminary processing for full parsing: cut down search space or provide guesses at unknown words.

Note: tags are more fine-grained than conventional part of speech. Different possible tagsets.

## Stochastic part of speech tagging using Hidden Markov Models (HMM)

1. Start with untagged text.
2. Assign all possible tags to each word in the text on the basis of a lexicon that associates words and tags.
3. Find the most probable sequence (or n-best sequences) of tags, based on probabilities from the training data.
  - ▶ lexical probability: e.g., is *can* most likely to be VM0, VVB, VVI or NN1?
  - ▶ and tag sequence probabilities: e.g., is VM0 or NN1 more likely after PNP?

## Training stochastic POS tagging

They\_PNP used\_VVD to\_TO0 can\_VVI fish\_NN2 in\_PRP  
those\_DT0 towns\_NN2 .\_PUN But\_CJC now\_AV0 few\_DT0  
people\_NN2 fish\_VVB in\_PRP these\_DT0 areas\_NN2  
.\_PUN

sequence	count	bigram probability
NN2	4	
NN2 PRP	1	0.25
NN2 PUN	2	0.5
NN2 VVB	1	0.25

Also lexicon: fish NN2 VVB

## Training stochastic POS tagging

They\_PNP used\_VVD to\_TO0 can\_VVI fish\_NN2 in\_PRP  
those\_DT0 towns\_NN2 .\_PUN But\_CJC now\_AV0 few\_DT0  
people\_NN2 fish\_VVB in\_PRP these\_DT0 areas\_NN2  
.\_PUN

sequence	count	bigram probability
NN2	4	
NN2 PRP	1	0.25
NN2 PUN	2	0.5
NN2 VVB	1	0.25

Also lexicon: fish NN2 VVB



## Training stochastic POS tagging

They\_PNP used\_VVD to\_T00 can\_VVI fish\_NN2 in\_PRP  
those\_DT0 towns\_NN2 .\_PUN But\_CJC now\_AV0 few\_DT0  
people\_NN2 fish\_VVB in\_PRP these\_DT0 areas\_NN2  
.\_PUN

sequence	count	bigram probability
NN2	4	
NN2 PRP	1	0.25
NN2 PUN	2	0.5
NN2 VVB	1	0.25

Also lexicon: fish NN2 VVB

## Assigning probabilities

Our estimate of the sequence of  $n$  tags is the sequence of  $n$  tags with the maximum probability, given the sequence of  $n$  words:

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n)$$

By Bayes theorem:

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)}$$

We're tagging a particular sequence of words so  $P(w_1^n)$  is constant, giving:

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(w_1^n | t_1^n) P(t_1^n)$$

## Assigning probabilities, continued

Bigram assumption: probability of a tag depends on the previous tag, hence approximate by the product of bigrams:

$$P(t_1^n) \approx \prod_{i=1}^n P(t_i | t_{i-1})$$

Probability of the word estimated on the basis of its own tag alone:

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^n P(w_i | t_i)$$

Hence:

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$

## Example

Tagging: *they fish*

Assume PNP is the only tag for *they*, and that *fish* could be NN2 or VVB.

Then the estimate for PNP NN2 will be:

$$P(\text{they} | \text{PNP}) P(\text{NN2} | \text{PNP}) P(\text{fish} | \text{NN2})$$

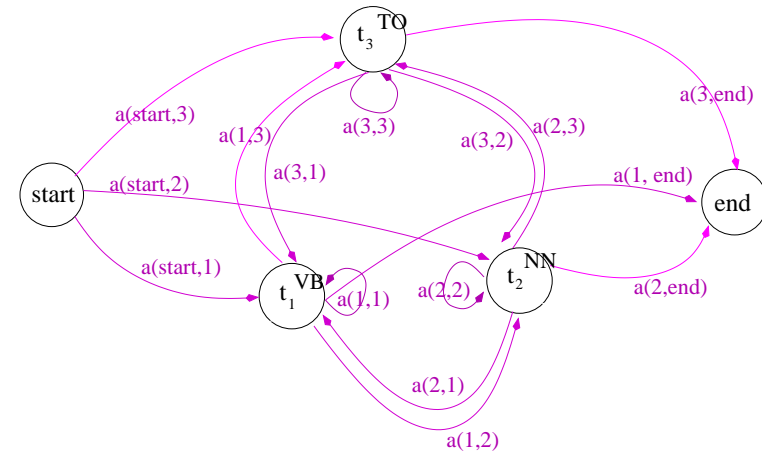
and for PNP VVB:

$$P(\text{they} | \text{PNP}) P(\text{VVB} | \text{PNP}) P(\text{fish} | \text{VVB})$$

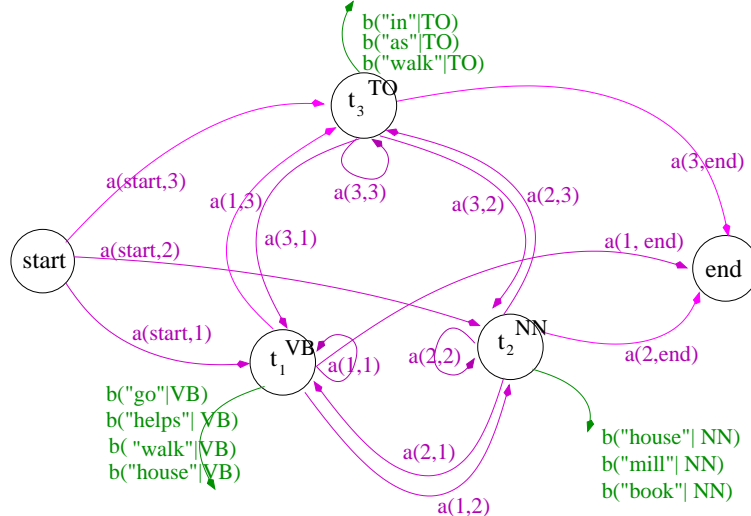
## Assigning probabilities, more details

- ▶ Maximise the overall tag sequence probability — e.g., use Viterbi.
- ▶ Actual systems use trigrams — smoothing and backoff are critical.
- ▶ Unseen words: these are not in the lexicon, so use all possible **open class** tags, possibly restricted by morphology.

## A Hidden Markov Model



## A Hidden Markov Model



## Evaluation of POS tagging

- ▶ percentage of correct tags
- ▶ one tag per word (some systems give multiple tags when uncertain)
- ▶ over 95% for English on normal corpora (but note punctuation is unambiguous)
- ▶ **baseline** of taking the most common tag gives 90% accuracy
- ▶ different tagsets give slightly different results: utility of tag to end users vs predictive power (an open research issue)

## Evaluation in general

- ▶ **Training data and test data** Test data must be kept unseen, often 90% training and 10% test data.
- ▶ **Baseline**
- ▶ **Ceiling** Human performance on the task, where the ceiling is the percentage agreement found between two annotators (**interannotator agreement**)
- ▶ **Error analysis** Error rates are nearly always unevenly distributed.
- ▶ **Reproducibility**

## Representative corpora and data sparsity

- ▶ test corpora have to be representative of the actual application
- ▶ POS tagging and similar techniques are not always very robust to differences in genre
- ▶ balanced corpora may be better, but still don't cover all text types
- ▶ communication aids: extreme difficulty in obtaining data, text corpora don't give good prediction for real data

## Outline of next lecture

### Lecture 4: Parsing and generation

Generative grammar  
 Simple context free grammars  
 Random generation with a CFG  
 Simple chart parsing with CFGs  
 More advanced chart parsing  
 Formalism power requirements

## Parsing (and generation)

Syntactic structure in analysis:

- ▶ as a step in assigning semantics
- ▶ checking grammaticality
- ▶ corpus-based investigations, lexical acquisition etc

### Lecture 4: Parsing and generation

Generative grammar  
 Simple context free grammars  
 Random generation with a CFG  
 Simple chart parsing with CFGs  
 More advanced chart parsing  
 Formalism power requirements

Next lecture — beyond simple CFGs

## Generative grammar

a formally specified grammar that can generate all and only the acceptable sentences of a natural language

Internal structure:

the big dog slept

can be bracketed

((the (big dog)) slept)

constituent a phrase whose components 'go together' ...

weak equivalence grammars generate the same strings

strong equivalence grammars generate the same strings with same brackets

## A simple CFG for a fragment of English

### rules

```
S -> NP VP
VP -> VP PP
VP -> V
VP -> V NP
VP -> V VP
NP -> NP PP
PP -> P NP
```

### lexicon

```
V -> can
V -> fish
NP -> fish
NP -> rivers
NP -> pools
NP -> December
NP -> Scotland
NP -> it
NP -> they
P -> in
```

## Context free grammars

1. a set of non-terminal symbols (e.g., S, VP);
2. a set of terminal symbols (i.e., the words);
3. a set of rules (productions), where the LHS (**mother**) is a single non-terminal and the RHS is a sequence of one or more non-terminal or terminal symbols (**daughters**);

S -> NP VP

V -> fish

4. a start symbol, conventionally S, which is a non-terminal.

Exclude empty productions, NOT e.g.:

NP ->  $\epsilon$

## Analyses in the simple CFG

they fish

(S (NP they) (VP (V fish)))

they can fish

(S (NP they) (VP (V can) (VP (V fish))))

(S (NP they) (VP (V can) (NP fish)))

they fish in rivers

(S (NP they) (VP (VP (V fish))  
(PP (P in) (NP rivers)))))

## Analyses in the simple CFG

they fish

(S (NP they) (VP (V fish)))

they can fish

(S (NP they) (VP (V can) (VP (V fish))))

(S (NP they) (VP (V can) (NP fish)))

they fish in rivers

(S (NP they) (VP (VP (V fish))  
(PP (P in) (NP rivers)))))

## Structural ambiguity without lexical ambiguity

they fish in rivers in December

(S (NP they)  
(VP (VP (V fish))  
(PP (P in) (NP rivers))  
(PP (P in) (NP December)))))

(S (NP they)  
(VP (VP (VP (V fish))  
(PP (P in) (NP rivers)))  
(PP (P in) (NP December)))))

## Analyses in the simple CFG

they fish

(S (NP they) (VP (V fish)))

they can fish

(S (NP they) (VP (V can) (VP (V fish))))

(S (NP they) (VP (V can) (NP fish)))

they fish in rivers

(S (NP they) (VP (VP (V fish))  
(PP (P in) (NP rivers)))))

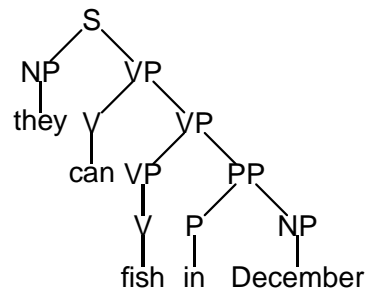
## Structural ambiguity without lexical ambiguity

they fish in rivers in December

(S (NP they)  
(VP (VP (V fish))  
(PP (P in) (NP rivers))  
(PP (P in) (NP December)))))

(S (NP they)  
(VP (VP (VP (V fish))  
(PP (P in) (NP rivers)))  
(PP (P in) (NP December)))))

## Parse trees



```

(S (NP they)
  (VP (V can)
    (VP (VP (V fish))
      (PP (P in)
        (NP December))))))
  
```

## Using a grammar as a random generator

### Expand cat *category sentence-record*:

Let *possibilities* be all lexical items matching *category*  
 and all rules with LHS *category*

If *possibilities* is empty,  
 then fail

else

Randomly select a possibility *chosen* from *possibilities*

If *chosen* is lexical,

then append it to *sentence-record*

else

expand cat on each rhs category in *chosen*

(left to right) with the updated *sentence-record*

return *sentence-record*

## Chart parsing

A dynamic programming algorithm (memoisation):

chart store partial results of parsing in a vector

edge representation of a rule application

Edge data structure:

[*id*, *left\_vtx*, *right\_vtx*, *mother\_category*, *dtrs*]

```

. they . can . fish .
0      1      2      3
  
```

Fragment of chart:

id	l	r	mo	dtrs
e	2	3	V	(fish)
f	2	3	VP	(e)
g	1	3	VP	(c f)

## A bottom-up passive chart parser

### Parse:

Initialize the chart

For each word *word*, let *from* be left vtx,

*to* right vtx and *dtrs* be (*word*)

For each category *category*

lexically associated with *word*

Add new edge *from*, *to*, *category*, *dtrs*

Output results for all spanning edges

## Inner function

**Add new edge** *from*, *to*, *category*, *dtrs*:

Put edge in chart: [*id*, *from*, *to*, *category*, *dtrs*]

For each *rule lhs*  $\rightarrow$  *cat*<sub>1</sub> ... *cat*<sub>*n*-1</sub>, *category*

Find sets of contiguous edges

[*id*<sub>1</sub>, *from*<sub>1</sub>, *to*<sub>1</sub>, *cat*<sub>1</sub>, *dtrs*<sub>1</sub>] ...

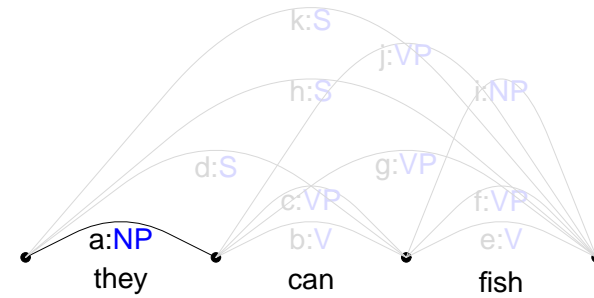
[*id*<sub>*n*-1</sub>, *from*<sub>*n*-1</sub>, *from*, *cat*<sub>*n*-1</sub>, *dtrs*<sub>*n*-1</sub>]

(such that *to*<sub>1</sub> = *from*<sub>2</sub> etc)

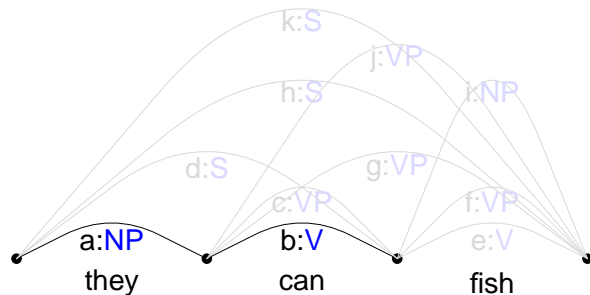
For each set of edges,

**Add new edge** *from*<sub>1</sub>, *to*, *lhs*, (*id*<sub>1</sub> ... *id*)

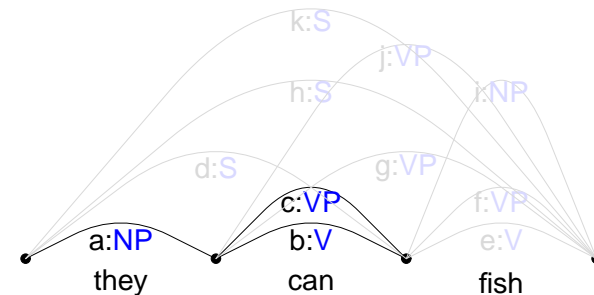
## Bottom up parsing: edges



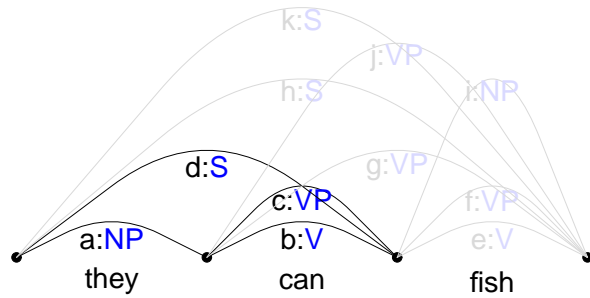
## Bottom up parsing: edges



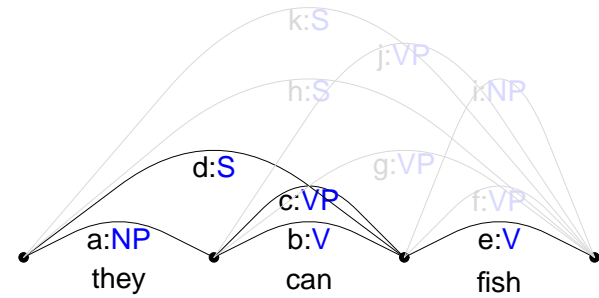
## Bottom up parsing: edges



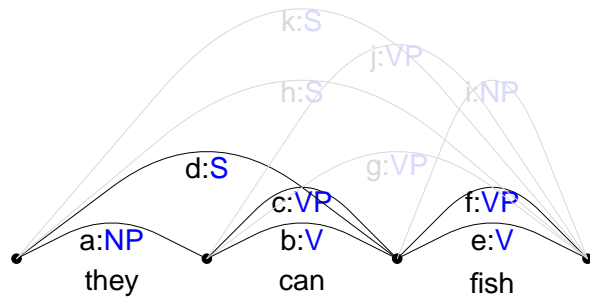
## Bottom up parsing: edges



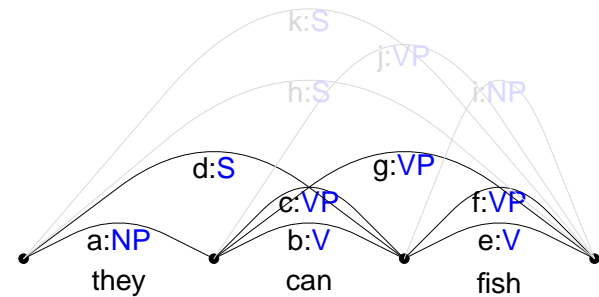
## Bottom up parsing: edges



## Bottom up parsing: edges

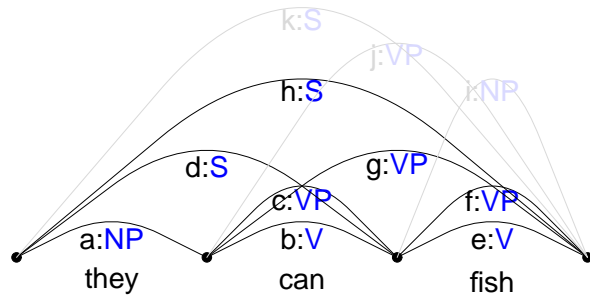


## Bottom up parsing: edges

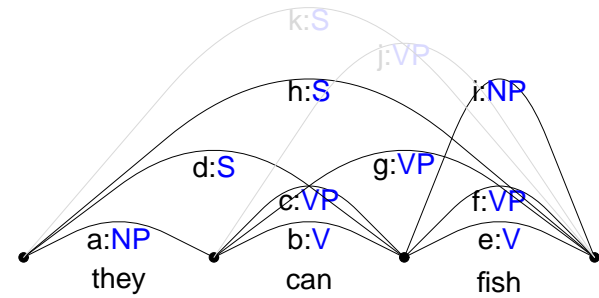




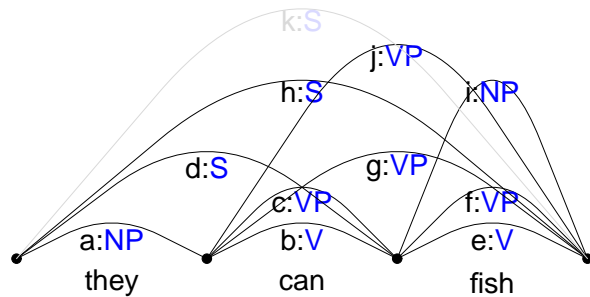
## Bottom up parsing: edges



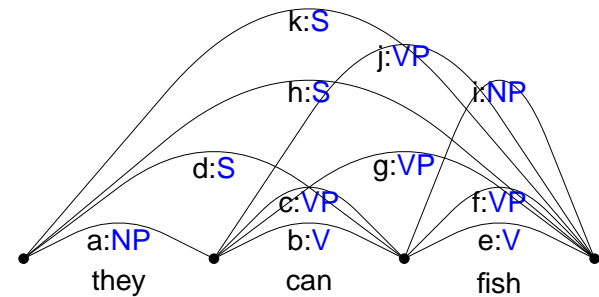
## Bottom up parsing: edges



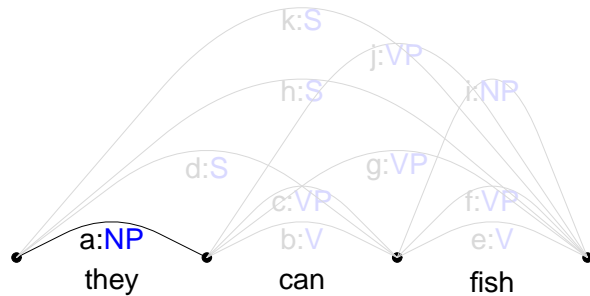
## Bottom up parsing: edges



## Bottom up parsing: edges



## Parse construction



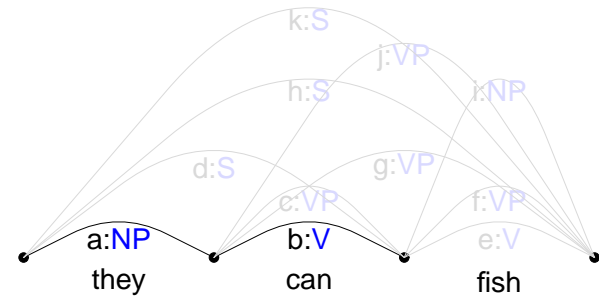
word = they, categories = {NP}

**Add new edge a** 0, 1, NP, (they)

Matching grammar rules: { $VP \rightarrow V \ NP$ ,  $PP \rightarrow P \ NP$ }

No matching edges corresponding to V or P

## Parse construction



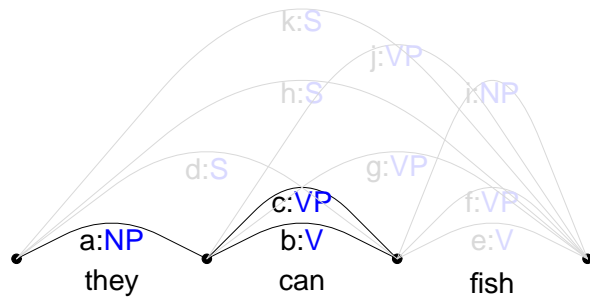
word = can, categories = {V}

**Add new edge b** 1, 2, V, (can)

Matching grammar rules: { $VP \rightarrow V$ }

recurse on edges {(b)}

## Parse construction

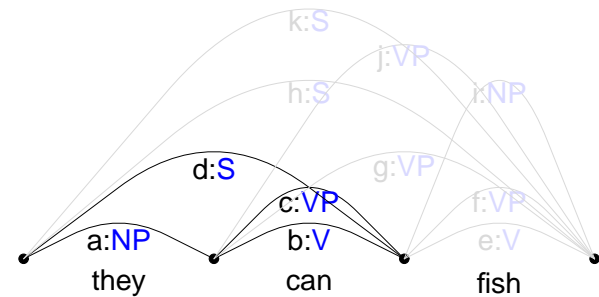


**Add new edge c** 1, 2, VP, (b)

Matching grammar rules: { $S \rightarrow NP \ VP$ ,  $VP \rightarrow V \ VP$ }

recurse on edges {(a,c)}

## Parse construction



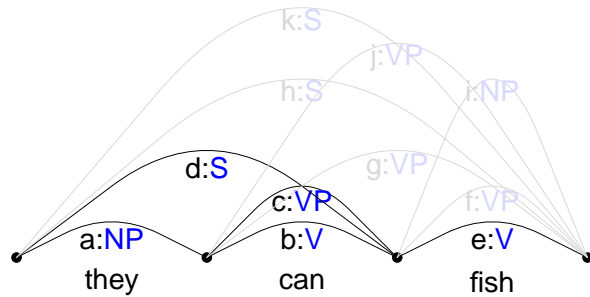
**Add new edge d** 0, 2, S, (a, c)

No matching grammar rules for S

Matching grammar rules: { $S \rightarrow NP \ VP$ ,  $VP \rightarrow V \ VP$ }

No edges for V VP

## Parse construction



word = fish, categories = {V, NP}

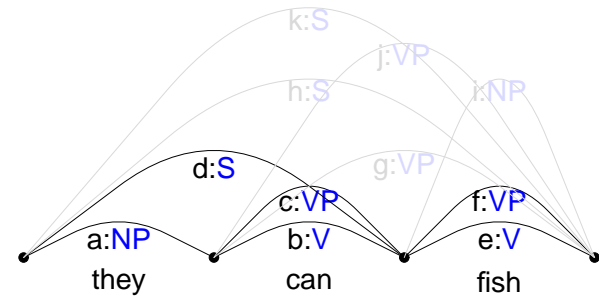
**Add new edge e** 2, 3, V, (fish)

Matching grammar rules: { $VP \rightarrow V$ }

recurse on edges {(e)}

NB: fish as V

## Parse construction



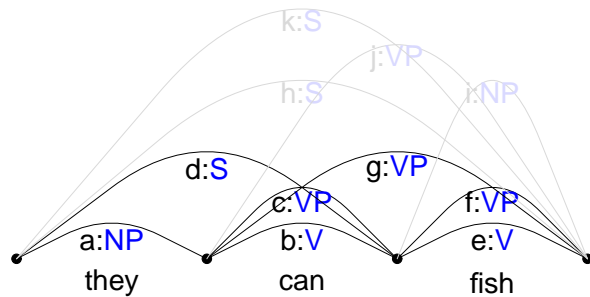
**Add new edge f** 2, 3, VP, (e)

Matching grammar rules: { $S \rightarrow NP VP$ ,  $VP \rightarrow V VP$ }

No edges match NP

recurse on edges for V VP: {(b,f)}

## Parse construction

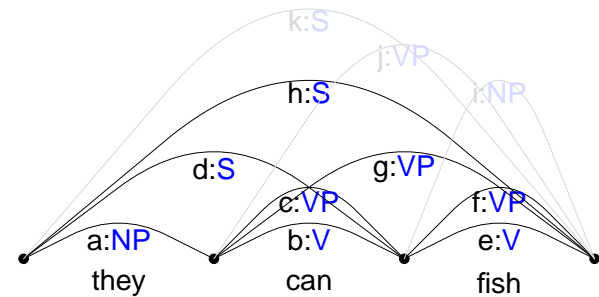


**Add new edge g** 1, 3, VP, (b, f)

Matching grammar rules: { $S \rightarrow NP VP$ ,  $VP \rightarrow V VP$ }

recurse on edges for NP VP: {(a,g)}

## Parse construction



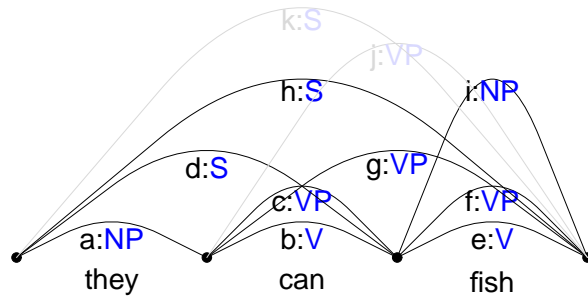
**Add new edge h** 0, 3, S, (a, g)

No matching grammar rules for S

Matching grammar rules: { $S \rightarrow NP VP$ ,  $VP \rightarrow V VP$ }

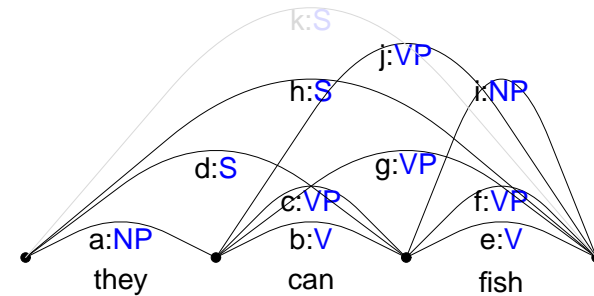
No edges matching V

## Parse construction



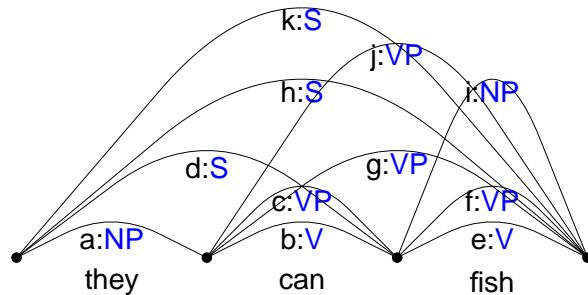
**Add new edge i** 2, 3, NP, (fish) NB: fish as NP  
 Matching grammar rules:  $\{VP \rightarrow V \text{ NP}, PP \rightarrow P \text{ NP}\}$   
 recurse on edges for V NP  $\{(b,i)\}$

## Parse construction



**Add new edge j** 1, 3, VP, (b,i)  
 Matching grammar rules:  $\{S \rightarrow NP \text{ VP}, VP \rightarrow V \text{ VP}\}$   
 recurse on edges for NP VP:  $\{(a,i)\}$

## Parse construction



**Add new edge k** 0, 3, S, (a,i)  
 No matching grammar rules for S  
 Matching grammar rules:  $\{S \rightarrow NP \text{ VP}, VP \rightarrow V \text{ VP}\}$   
 No edges corresponding to V VP  
 Matching grammar rules:  $\{VP \rightarrow V \text{ NP}, PP \rightarrow P \text{ NP}\}$   
 No edges corresponding to P NP

## Output results for spanning edges

Spanning edges are h and k:  
 Output results for h

$(S \text{ (NP they) (VP (V can) (VP (V fish))))$

Output results for k

$(S \text{ (NP they) (VP (V can) (NP fish)))$

Note: sample chart parsing code in Java is downloadable from the course web page.

## Packing

- ▶ exponential number of parses means exponential time
- ▶ body can be cubic time: don't add equivalent edges as whole new edges
- ▶ *dtrs* is a set of lists of edges (to allow for alternatives)

about to add: [*id*, *l\_vtx*, *right\_vtx*, *ma\_cat*, *dtrs*]  
and there is an existing edge:

[*id-old*, *l\_vtx*, *right\_vtx*, *ma\_cat*, *dtrs-old*]

we simply modify the old edge to record the new dtrs:

[*id-old*, *l\_vtx*, *right\_vtx*, *ma\_cat*, *dtrs-old*  $\cup$  *dtrs*]

and **do not recurse on it**: never need to continue computation with a packable edge.

## Packing example

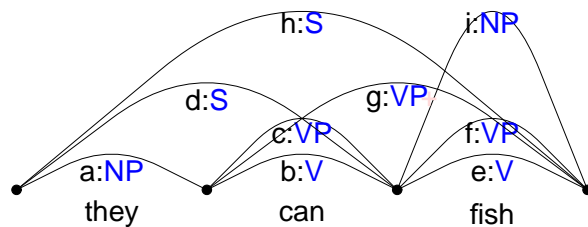
a	0	1	NP	{(they)}
b	1	2	V	{(can)}
c	1	2	VP	{(b)}
d	0	2	S	{(a c)}
e	2	3	V	{(fish)}
f	2	3	VP	{(e)}
g	1	3	VP	{(b f)}
h	0	3	S	{(a g)}
i	2	3	NP	{(fish)}

Instead of edge j 1 3 VP {(b,i)}

g 1 3 VP {(b,f), (b,i)}

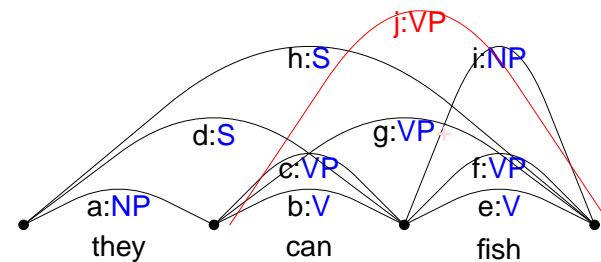
and we're done

## Packing example



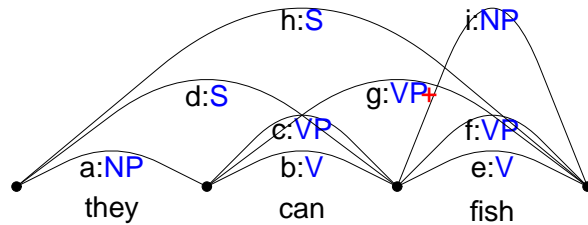
Both spanning results can now be extracted from edge h.

## Packing example



Both spanning results can now be extracted from edge h.

## Packing example



Both spanning results can now be extracted from edge h.

## Why not FSA?

centre-embedding:

$$A \rightarrow \alpha A \beta$$

generate grammars of the form  $a^n b^n$ .

For instance:

the students the police arrested complained

However, limits on human memory / processing ability:

? the students the police the journalists criticised arrested complained

More importantly:

1. FSM grammars are extremely redundant
2. FSM grammars don't support composition of semantics

## Ordering the search space

- ▶ agenda: order edges in chart by priority
- ▶ top-down parsing: predict possible edges

Producing n-best parses:

- ▶ manual weight assignment
- ▶ probabilistic CFG — trained on a treebank
  - ▶ automatic grammar induction
  - ▶ automatic weight assignment to existing grammar
- ▶ beam-search

## Why not FSA?

centre-embedding:

$$A \rightarrow \alpha A \beta$$

generate grammars of the form  $a^n b^n$ .

For instance:

the students the police arrested complained

However, limits on human memory / processing ability:

? the students the police the journalists criticised arrested complained

More importantly:

1. FSM grammars are extremely redundant
2. FSM grammars don't support composition of semantics

## Overgeneration in atomic category CFGs

- ▶ **agreement**: subject verb agreement. e.g., *they fish, it fishes, \*it fish, \*they fishes*. \* means ungrammatical
- ▶ **case**: pronouns (and maybe *who/whom*) e.g., *they like them, \*they like they*

## Overgeneration in atomic category CFGs

- ▶ **agreement**: subject verb agreement. e.g., *they fish, it fishes, \*it fish, \*they fishes*. \* means ungrammatical
- ▶ **case**: pronouns (and maybe *who/whom*) e.g., *they like them, \*they like they*

S -> NP-sg-nom VP-sg	NP-sg-nom -> he
S -> NP-pl-nom VP-pl	NP-sg-acc -> him
VP-sg -> V-sg NP-sg-acc	NP-sg-nom -> fish
VP-sg -> V-sg NP-pl-acc	NP-pl-nom -> fish
VP-pl -> V-pl NP-sg-acc	NP-sg-acc -> fish
VP-pl -> V-pl NP-pl-acc	NP-pl-acc -> fish

BUT: very large grammar, misses generalizations, no way of saying when we don't care about agreement.

## Subcategorization

- ▶ intransitive vs transitive etc
- ▶ verbs (and other types of words) have different numbers and types of syntactic arguments:
  - \**Kim adored*
  - \**Kim gave Sandy*
  - \**Kim adored to sleep*
  - Kim liked to sleep*
  - \**Kim devoured*
  - Kim ate*
- ▶ Subcategorization is correlated with semantics, but not determined by it.

## Overgeneration because of missing subcategorization

Overgeneration:

they fish fish it  
(S (NP they) (VP (V fish) (VP (V fish) (NP it))))

- ▶ Informally: need slots on the verbs for their syntactic arguments.
  - ▶ intransitive takes no following arguments (complements)
  - ▶ simple transitive takes one NP complement
  - ▶ *like* may be a simple transitive or take an infinitival complement, etc

## Outline of next lecture

Providing a more adequate treatment of syntax than simple CFGs: replacing the atomic categories by more complex data structures.

### Lecture 5: Parsing with constraint-based grammars

- Beyond simple CFGs
- Feature structures
- Encoding agreement
- Parsing with feature structures
- Encoding subcategorisation
- Interface to morphology

## Long-distance dependencies

1. which problem did you say you don't understand?
2. who do you think Kim asked Sandy to hit?
3. which kids did you say were making all that noise?

'gaps' (underscores below)

1. which problem did you say you don't understand \_?
2. who do you think Kim asked Sandy to hit \_?
3. which kids did you say \_ were making all that noise?

In 3, the verb *were* shows plural agreement.

\* what kid did you say \_ were making all that noise?

The gap filler has to be plural.

- Informally: need a 'gap' slot which is to be filled by something that itself has features.

## Outline of today's lecture

### Lecture 5: Parsing with constraint-based grammars

- Beyond simple CFGs
- Feature structures
- Encoding agreement
- Parsing with feature structures
- Encoding subcategorisation
- Interface to morphology

## Context-free grammar and language phenomena

- CFGs can encode long-distance dependencies
- Language phenomena that CFGs cannot model (without a bound) are unusual — probably none in English.
- BUT: CFG modelling for English or another NL could be trillions of rules
- Enriched formalisms: CFG equivalent (today) or greater power (more usual)
- Human processing vs linguistic generalisations.



## Constraint-based grammar (feature structures)

Providing a more adequate treatment of syntax than simple CFGs by replacing the atomic categories by more complex data structures.

- ▶ Feature structure formalisms give good linguistic accounts for many languages
- ▶ Reasonably computationally tractable
- ▶ Bidirectional (parse and generate)
- ▶ Used in LFG and HPSG formalisms

## Expanded CFG (from last time)

S -> NP-sg-nom VP-sg	NP-sg-nom -> he
S -> NP-pl-nom VP-pl	NP-sg-acc -> him
VP-sg -> V-sg NP-sg-acc	NP-sg-nom -> fish
VP-sg -> V-sg NP-pl-acc	NP-pl-nom -> fish
VP-pl -> V-pl NP-sg-acc	NP-sg-acc -> fish
VP-pl -> V-pl NP-pl-acc	NP-pl-acc -> fish

## Intuitive solution for case and agreement

- ▶ Separate slots (features) for CASE and AGR
- ▶ Slot values for CASE may be **nom** (e.g., *they*), **acc** (e.g., *them*) or unspecified (i.e., don't care)
- ▶ Slot values for AGR may be **sg**, **pl** or unspecified
- ▶ Subjects have the same value for AGR as their verbs
- ▶ Subjects have CASE **nom**, objects have CASE **acc**

can (n)	<div> <div>CASE [ ]</div> <div>AGR <b>sg</b></div> </div>	fish (n)	<div> <div>CASE [ ]</div> <div>AGR [ ]</div> </div>
she	<div> <div>CASE <b>nom</b></div> <div>AGR <b>sg</b></div> </div>	them	<div> <div>CASE <b>acc</b></div> <div>AGR <b>pl</b></div> </div>

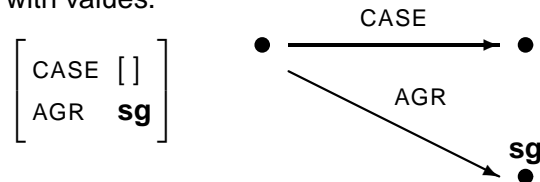
## Feature structures

$$\begin{bmatrix} \text{CASE} & [] \\ \text{AGR} & \textbf{sg} \end{bmatrix}$$

1. **Features** like AGR with simple values: **atomic-valued**
2. Unspecified values possible on features: compatible with any value.
3. Values for features for subcat and gap themselves have features: **complex-valued**
4. **path**: a sequence of features
5. Method of specifying two paths are the same: **reenentrancy**

## Feature structures, continued

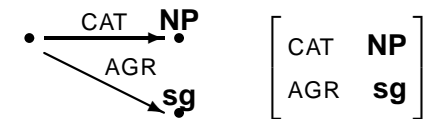
- Feature structures are singly-rooted directed acyclic graphs, with arcs labelled by features and terminal nodes associated with values.



- In grammars, rules relate FSs — i.e. lexical entries and phrases are represented as FSs
- Rule application by unification

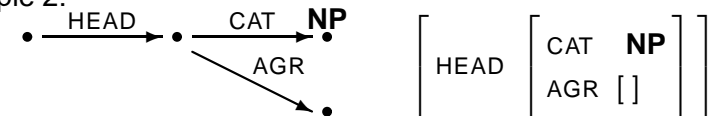
## Graphs and AVMs

Example 1:



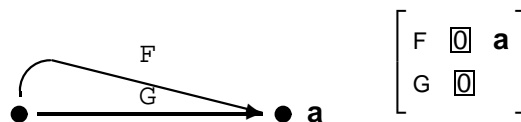
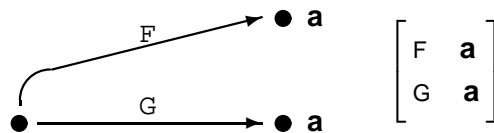
Here, CAT and AGR are atomic-valued features. **NP** and **sg** are values.

Example 2:



HEAD is complex-valued, AGR is unspecified.

## Reentrancy



Reentrancy indicated by boxed integer in AVM diagram: indicates path goes to the same node.

## Properties of FSs

**Connectedness and unique root** A FS must have a unique root node: apart from the root node, all nodes have one or more parent nodes.

**Unique features** Any node may have zero or more arcs leading out of it, but the label on each (that is, the feature) must be unique.

**No cycles** No node may have an arc that points back to the root node or to a node that intervenes between it and the root node.

**Values** A node which does not have any arcs leading out of it may have an associated atomic value.

**Finiteness** A FS must have a finite number of nodes.

## Unification

**Unification** is an operation that combines two feature structures, retaining all information from each, or failing if information is incompatible.

Some simple examples:

1.  $\begin{bmatrix} \text{CASE} & [] \\ \text{AGR} & \text{sg} \end{bmatrix} \sqcap \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & [] \end{bmatrix} = \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \text{sg} \end{bmatrix}$
2.  $\begin{bmatrix} \text{CASE} & [] \\ \text{AGR} & \text{sg} \end{bmatrix} \sqcap \begin{bmatrix} \text{AGR} & [] \end{bmatrix} = \begin{bmatrix} \text{CASE} & [] \\ \text{AGR} & \text{sg} \end{bmatrix}$
3.  $\begin{bmatrix} \text{CASE} & [] \\ \text{AGR} & \text{sg} \end{bmatrix} \sqcap \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{AGR} & \text{pl} \end{bmatrix} = \text{fail}$

## Subsumption and Unification

Feature structures are ordered by information content — FS1 **subsumes** FS2 if FS2 carries extra information.

FS1 subsumes FS2 if and only if the following conditions hold:

**Path values** For every path P in FS1 there is a path P in FS2. If P has a value t in FS1, then P also has value t in FS2.

**Path equivalences** Every pair of paths P and Q which are reentrant in FS1 (i.e., which lead to the same node in the graph) are also reentrant in FS2.

The **unification** of two FSs FS1 and FS2 is then defined as the most general FS which is subsumed by both FS1 and FS2, if it exists.

## CFG with agreement

```

S -> NP-sg VP-sg
S -> NP-pl VP-pl
VP-sg -> V-sg NP-sg
VP-sg -> V-sg NP-pl
VP-pl -> V-pl NP-sg
VP-pl -> V-pl NP-pl
V-pl -> like
V-sg -> likes
NP-sg -> it
NP-pl -> they
NP-sg -> fish
NP-pl -> fish

```

## FS grammar fragment encoding agreement

subj-verb rule  $\begin{bmatrix} \text{CAT} & \mathbf{S} \\ \text{AGR} & \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \end{bmatrix}$

verb-obj rule  $\begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & [] \end{bmatrix}$

Root structure:  $\begin{bmatrix} \text{CAT} & \mathbf{S} \end{bmatrix}$

they  $\begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \mathbf{pl} \end{bmatrix}$  it  $\begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \mathbf{sg} \end{bmatrix}$  likes  $\begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \mathbf{sg} \end{bmatrix}$

fish  $\begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & [] \end{bmatrix}$  like  $\begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \mathbf{pl} \end{bmatrix}$

## Parsing 'they like it'

- ▶ The lexical structures for *like* and *it* are unified with the corresponding structures on the right hand side of the verb-obj rule (unifications succeed).
- ▶ The structure corresponding to the mother of the rule is then:

$$\begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \mathbf{pl} \end{bmatrix}$$

- ▶ This unifies with the rightmost daughter position of the subj-verb rule.
- ▶ The structure for *they* is unified with the leftmost daughter.
- ▶ The result unifies with root structure.

## Rules as FSs

But what does the coindexation of parts of the rule mean? Treat rule as a FS: e.g., rule features MOTHER, DTR1, DTR2 ... DTRN.

$$\text{informally: } \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & [] \end{bmatrix}$$

$$\text{actually: } \begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR1} & \begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR2} & \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & [] \end{bmatrix} \end{bmatrix}$$

## Verb-obj rule application

Feature structure for *like* unified with the value of DTR1:

$$\begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \text{ pl} \end{bmatrix} \\ \text{DTR1} & \begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR2} & \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & [] \end{bmatrix} \end{bmatrix}$$

Feature structure for *it* unified with the value for DTR2:

$$\begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \text{ pl} \end{bmatrix} \\ \text{DTR1} & \begin{bmatrix} \text{CAT} & \mathbf{V} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR2} & \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \mathbf{sg} \end{bmatrix} \end{bmatrix}$$

## Subject-verb rule application 1

MOTHER value from the verb-object rule acts as the DTR2 of the subject-verb rule:

$$\begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \mathbf{pl} \end{bmatrix} \text{ unified with the DTR2 of: } \begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{CAT} & \mathbf{S} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR1} & \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR2} & \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Gives:

$$\begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{CAT} & \mathbf{S} \\ \text{AGR} & \boxed{1} \text{ pl} \end{bmatrix} \\ \text{DTR1} & \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{DTR2} & \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

## Subject rule application 2

FS for *they*:  $\begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \mathbf{pl} \end{bmatrix}$

Unification of this with the value of DTR1 succeeds (but adds no new information):

$$\begin{bmatrix} \text{MOTHER} & \begin{bmatrix} \text{CAT} & \mathbf{S} \\ \text{AGR} & \mathbf{1} \text{ pl} \end{bmatrix} \\ \text{DTR1} & \begin{bmatrix} \text{CAT} & \mathbf{NP} \\ \text{AGR} & \mathbf{1} \end{bmatrix} \\ \text{DTR2} & \begin{bmatrix} \text{CAT} & \mathbf{VP} \\ \text{AGR} & \mathbf{1} \end{bmatrix} \end{bmatrix}$$

Final structure unifies with the root structure:  $\begin{bmatrix} \text{CAT} & \mathbf{S} \end{bmatrix}$

## Grammar with subcategorisation

Verb-obj rule:  $\begin{bmatrix} \text{HEAD} & \mathbf{1} \\ \text{OBJ} & \mathbf{filled} \\ \text{SUBJ} & \mathbf{3} \end{bmatrix} \rightarrow \begin{bmatrix} \text{HEAD} & \mathbf{1} \\ \text{OBJ} & \mathbf{2} \\ \text{SUBJ} & \mathbf{3} \end{bmatrix}, \mathbf{2} [\text{OBJ} \mathbf{filled}]$

can (transitive verb):  $\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \mathbf{verb} \\ \text{AGR} & \mathbf{pl} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \mathbf{noun} \end{bmatrix} \\ \text{OBJ} & \mathbf{filled} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \mathbf{noun} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

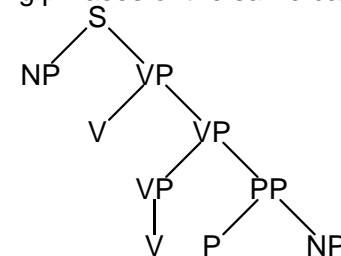
## Grammar with subcategorisation (abbrev for slides)

Verb-obj rule:  $\begin{bmatrix} \text{HEAD} & \mathbf{1} \\ \text{OBJ} & \mathbf{fld} \\ \text{SUBJ} & \mathbf{3} \end{bmatrix} \rightarrow \begin{bmatrix} \text{HEAD} & \mathbf{1} \\ \text{OBJ} & \mathbf{2} \\ \text{SUBJ} & \mathbf{3} \end{bmatrix}, \mathbf{2} [\text{OBJ} \mathbf{fld}]$

can (transitive verb):  $\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \mathbf{v} \\ \text{AGR} & \mathbf{pl} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \mathbf{n} \end{bmatrix} \\ \text{OBJ} & \mathbf{fld} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \mathbf{n} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

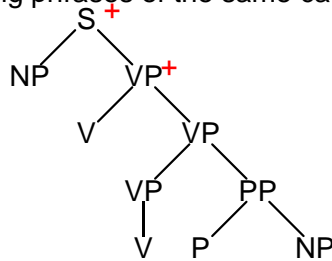
## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category



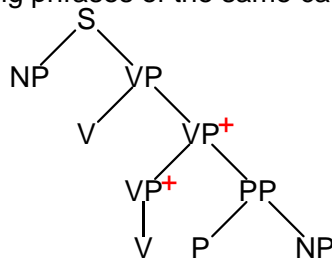
## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category



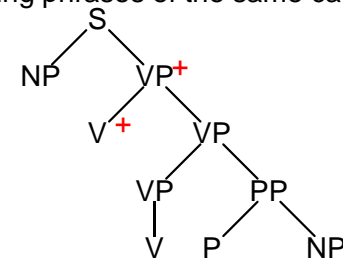
## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category



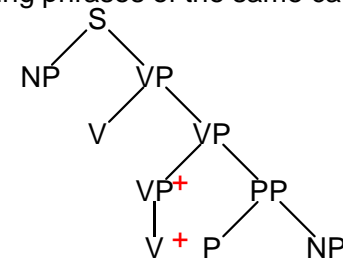
## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category



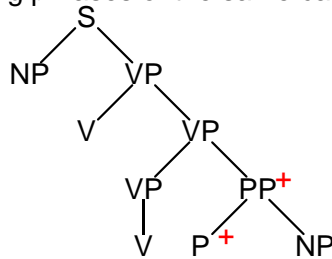
## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category



## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category



## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category

- **SUBJ**:

The subject-verb rule unifies the first daughter of the rule with the SUBJ value of the second. ('the first dtr fills the SUBJ slot of the second dtr in the rule')

## Concepts for subcategorisation

- **HEAD**: information shared between a lexical entry and the dominating phrases of the same category

- **SUBJ**:

The subject-verb rule unifies the first daughter of the rule with the SUBJ value of the second. ('the first dtr fills the SUBJ slot of the second dtr in the rule')

- **OBJ**:

The verb-object rule unifies the second dtr with the OBJ value of the first. ('the second dtr fills the OBJ slot of the first dtr in the rule')

## Example rule application: *they fish* 1

Lexical entry for fish:

$$\left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{v} \\ \text{AGR } \mathbf{pl} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ} \left[ \text{HEAD} \left[ \text{CAT } \mathbf{n} \right] \right] \end{array} \right]$$

subject-verb rule:

$$\left[ \begin{array}{l} \text{HEAD } \boxed{1} \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right] \rightarrow \boxed{2} \left[ \begin{array}{l} \text{HEAD} \left[ \text{AGR } \boxed{3} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right], \left[ \begin{array}{l} \text{HEAD } \boxed{1} \left[ \text{AGR } \boxed{3} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \boxed{2} \end{array} \right]$$

unification with second dtr position gives:

$$\left[ \begin{array}{l} \text{HEAD } \boxed{1} \left[ \begin{array}{l} \text{CAT } \mathbf{v} \\ \text{AGR } \boxed{3} \mathbf{pl} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right] \rightarrow \boxed{2} \left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{n} \\ \text{AGR } \boxed{3} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right], \left[ \begin{array}{l} \text{HEAD } \boxed{1} \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \boxed{2} \end{array} \right]$$

Lexical entry for *they*:

$$\left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{n} \\ \text{AGR } \mathbf{pl} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right]$$

unify this with first dtr position:

$$\left[ \begin{array}{l} \text{HEAD } \boxed{1} \left[ \begin{array}{l} \text{CAT } \mathbf{v} \\ \text{AGR } \boxed{3} \mathbf{pl} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right] \rightarrow \boxed{2} \left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{n} \\ \text{AGR } \boxed{3} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right], \left[ \begin{array}{l} \text{HEAD } \boxed{1} \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \boxed{2} \end{array} \right]$$

Root is:

$$\left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{v} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right]$$

Mother structure unifies with root, so valid.

## Parsing with feature structure grammars

- ▶ Naive algorithm: standard chart parser with modified rule application
- ▶ Rule application:
  1. copy rule
  2. copy daughters (lexical entries or FSs associated with edges)
  3. unify rule and daughters
  4. if successful, add new edge to chart with rule FS as category
- ▶ Efficient algorithms reduce copying.
- ▶ Packing involves subsumption.
- ▶ Probabilistic FS grammars are complex.

## Templates

Capture generalizations in the lexicon:

fish INTRANS\_VERB

sleep INTRANS\_VERB

snore INTRANS\_VERB

INTRANS\_VERB

$$\left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{v} \\ \text{AGR } \mathbf{pl} \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ} \left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{n} \end{array} \right] \end{array} \right] \end{array} \right]$$

## Interface to morphology: inflectional affixes as FSs

s

$$\left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{n} \\ \text{AGR } \mathbf{pl} \end{array} \right] \end{array} \right]$$

if stem is:

$$\left[ \begin{array}{l} \text{HEAD} \left[ \begin{array}{l} \text{CAT } \mathbf{n} \\ \text{AGR } [ ] \end{array} \right] \\ \text{OBJ } \mathbf{fld} \\ \text{SUBJ } \mathbf{fld} \end{array} \right]$$

stem unifies with affix template.

But unification failure would occur with verbs etc, so we get filtering (lecture 2).



## Outline of next lecture

Compositional semantics: the construction of meaning (generally expressed as logic) based on syntax.

Lexical semantics: the meaning of individual words.

### Lecture 6: Compositional and lexical semantics

Compositional semantics in feature structures

Logical forms

Meaning postulates

Lexical semantics: semantic relations

Polysemy

Word sense disambiguation

## Outline of today's lecture

Compositional semantics: the construction of meaning (generally expressed as logic) based on syntax.

Lexical semantics: the meaning of individual words.

### Lecture 6: Compositional and lexical semantics

Compositional semantics in feature structures

Logical forms

Meaning postulates

Lexical semantics: semantic relations

Polysemy

Word sense disambiguation

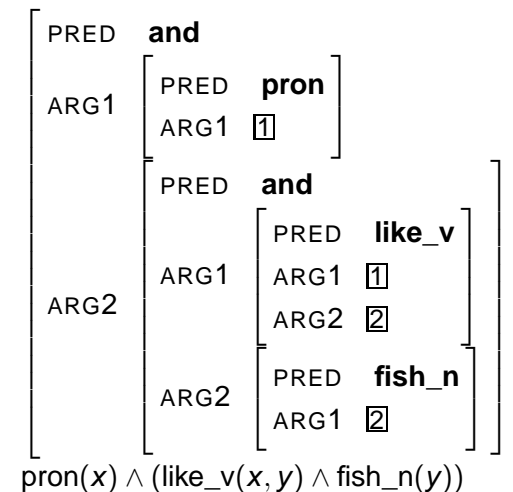
## Simple compositional semantics in feature structures

- ▶ Semantics is built up along with syntax
- ▶ Subcategorization 'slot' filling instantiates syntax
- ▶ Formally equivalent to logical representations (below: predicate calculus with no quantifiers)
- ▶ Alternative FS encodings possible

Objective: obtain the following semantics for *they like fish*:

$\text{pron}(x) \wedge (\text{like\_v}(x, y) \wedge \text{fish\_n}(y))$

## Feature structure encoding of semantics



## Noun entry

fish	HEAD	[ CAT <b>n</b> AGR [ ] ]	
	OBJ	<b>fld</b>	
	SUBJ	<b>fld</b>	
	SEM	[ INDEX ① PRED <b>fish_n</b> ARG1 ① ]	

- Corresponds to fish(x) where the INDEX points to the characteristic variable of the noun (that is x).

The INDEX is unambiguous here, but  
e.g., picture(x, y) ∧ sheep(y)  
picture of sheep

## Noun entry

fish	HEAD	[ CAT <b>n</b> AGR [ ] ]	
	OBJ	<b>fld</b>	
	SUBJ	<b>fld</b>	
	SEM	[ INDEX ① PRED <b>fish_n</b> ARG1 ① ]	

- Corresponds to fish(x) where the INDEX points to the characteristic variable of the noun (that is x).

The INDEX is unambiguous here, but  
e.g., picture(x, y) ∧ sheep(y)  
picture of sheep

## Verb entry

like	HEAD	[ CAT <b>v</b> AGR <b>pl</b> ]	
	OBJ	[ HEAD [ CAT <b>n</b> ] OBJ <b>fld</b> SEM [ INDEX ② ] ]	
	SUBJ	[ HEAD [ CAT <b>n</b> ] SEM [ INDEX ① ] ]	
	SEM	[ PRED <b>like_v</b> ARG1 ① ARG2 ② ]	

## Verb-object rule

[ HEAD ① OBJ <b>fld</b> SUBJ ③ SEM [ PRED <b>and</b> ARG1 ④ ARG2 ⑤ ] ]	→	[ HEAD ① OBJ ② SUBJ ③ SEM ④ ]	,	②	[ OBJ <b>fld</b> SEM ⑤ ]

- As last time: object of the verb (DTR2) 'fills' the OBJ slot
- New: semantics on the mother is the 'and' of the semantics of the dtrs

## Logic in semantic representation

- ▶ Meaning representation for a sentence is called the **logical form**
- ▶ Standard approach to composition in theoretical linguistics is lambda calculus, building FOPC or higher order representation.
- ▶ Representation in notes is quantifier-free predicate calculus but possible to build FOPC or higher-order representation in FSs.
- ▶ Theorem proving.
- ▶ Generation: starting point is logical form, not string.

## Meaning postulates

- ▶ e.g.,

$$\forall x[\text{bachelor}'(x) \rightarrow \text{man}'(x) \wedge \text{unmarried}'(x)]$$

- ▶ usable with compositional semantics and theorem provers
- ▶ e.g. from 'Kim is a bachelor', we can construct the LF

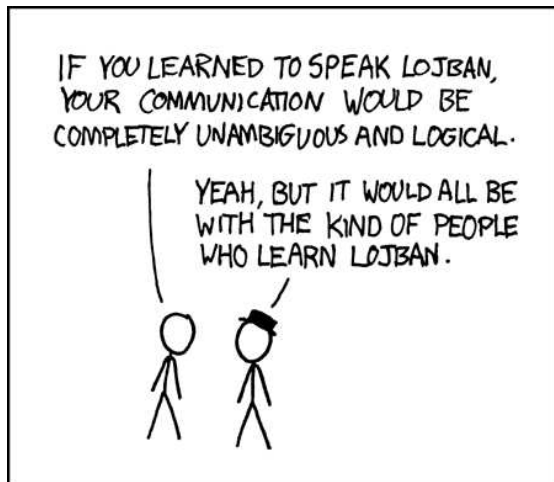
bachelor' (Kim)

and then deduce

unmarried'(Kim)

- ▶ OK for narrow domains or micro-worlds.

## Unambiguous and logical?



## Lexical semantic relations

**Hyponymy: IS-A:**

- ▶ (a sense of) *dog* is a **hyponym** of (a sense of) *animal*
- ▶ *animal* is a **hypernym** of *dog*
- ▶ hyponymy relationships form a **taxonomy**
- ▶ works best for concrete nouns

Meronymy: PART-OF e.g., *arm* is a **meronym** of *body*, *steering wheel* is a meronym of *car* (piece vs part)

**Synonymy** e.g., *aubergine/eggplant*

**Antonymy** e.g., *big/little*

## WordNet

- ▶ large scale, open source resource for English
- ▶ hand-constructed
- ▶ wordnets being built for other languages
- ▶ organized into **synsets**: synonym sets (near-synonyms)

Overview of adj red:

1. (43) red, reddish, ruddy, blood-red, carmine, cerise, cherry, cherry-red, crimson, ruby, ruby-red, scarlet - (having any of numerous bright or strong colors reminiscent of the color of blood or cherries or tomatoes or rubies)
2. (8) red, reddish - ((used of hair or fur) of a reddish brown color; "red deer"; reddish hair")

## Hyponymy in WordNet

Sense 6

big cat, cat

=> leopard, *Panthera pardus*

=> leopar~~d~~ess

=> panther

=> snow leopard, ounce, *Panthera uncia*

=> jaguar, panther, Panthera onca,

Felis onca

=> lion, king of beasts, Panthera leo

=> lioness

=> lionet

=> tiger, *Panthera tigris*

=> Bengal tiger

=> tigress

## Some uses of lexical semantics

- ▶ Semantic classification: e.g., for selectional restrictions (e.g., the object of *eat* has to be something edible) and for named entity recognition
- ▶ Shallow inference: 'X murdered Y' implies 'X killed Y' etc
- ▶ Back-off to semantic classes in some statistical approaches
- ▶ Word-sense disambiguation
- ▶ Query expansion: if a search doesn't return enough results, one option is to replace an over-specific term with a hypernym

## Lexical Relations in Compounds



## X-proofing

acid-proof, affair-proof, air-proof, ant-proof, baby-proof, bat-proof, bear-proof, bite-proof, bomb-proof, bullet-proof, burglar-proof, cat-proof, cannon-proof, claw-proof, coyote-proof, crash-proof, crush-proof, deer-proof, disaster-proof, dust-proof, dog-proof, elephant-proof, escape-proof, explosion-proof, fade-proof, fire-proof, flame-proof, flood-proof, fool-proof, fox-proof, frost-proof, fume-proof, gas-proof, germ-proof, glare-proof, goof-proof, gorilla-proof, grease-proof, hail-proof, heat-proof, high-proof (110-proof, 80-proof), hurricane-proof, ice-proof, idiot-proof, jam-proof, leak-proof, leopard-proof, lice-proof, light-proof, mole-proof, moth-proof, mouse-proof, nematode-proof, noise-proof, oil-proof, oven-proof, pet-proof, pilfer-proof, porcupine-proof, possum-proof, puncture-proof, quake-proof, rabbit-proof, raccoon-proof, radiation-proof, rain-proof, rat-proof, rattle-proof, recession-proof, rip-proof, roach-proof, rub-proof, rust-proof, sand-proof, scatter-proof, scratch-proof, shark-proof, shatter-proof, shell-proof, shock-proof, shot-proof, skid-proof, slash-proof, sleet-proof, slip-proof, smear-proof, smell-proof, smudge-proof, snag-proof, snail-proof, snake-proof, snow-proof, sound-proof, stain-proof, steam-proof, sun-proof, tamper-proof, tear-proof, teenager-proof, tick-proof, tornado-proof, trample-proof, varmint-proof, veto-proof, vibration-proof, water-proof, weasel-proof, weather-proof, wind-proof, wolf-proof, wrinkle-proof, x-ray-proof, zap-proof

source: [www.wordnik.com/lists/heres-your-proof](http://www.wordnik.com/lists/heres-your-proof)

## Polysemy

- ▶ **homonymy**: unrelated word senses. *bank* (raised land) vs *bank* (financial institution)
- ▶ *bank* (financial institution) vs *bank* (in a casino): related but distinct senses.
- ▶ *bank* (N) (raised land) vs *bank* (V) (to create some raised land): **regular polysemy**. Compare *pile*, *heap* etc

No clearcut distinctions.

Dictionaries are not consistent.

## Word sense disambiguation

Needed for many applications, problematic for large domains.

Assumes that we have a standard set of word senses (e.g., WordNet)

- ▶ frequency: e.g., *diet*: the food sense (or senses) is much more frequent than the parliament sense (Diet of Worms)
- ▶ collocations: e.g. *striped bass* (the fish) vs *bass guitar*: syntactically related or in a window of words (latter sometimes called 'cooccurrence'). Generally 'one sense per collocation'.
- ▶ selectional restrictions/preferences (e.g., *Kim eats bass*, must refer to fish)

## WSD techniques

- ▶ supervised learning: cf. POS tagging from lecture 3. But sense-tagged corpora are difficult to construct, algorithms need far more data than POS tagging
- ▶ unsupervised learning (see below)
- ▶ Machine readable dictionaries (MRDs): e.g., look at overlap with words in definitions and example sentences
- ▶ selectional preferences: don't work very well by themselves, useful in combination with other techniques

## WSD by (almost) unsupervised learning

Disambiguating *plant* (factory vs vegetation senses):

1. Find contexts in training corpus:

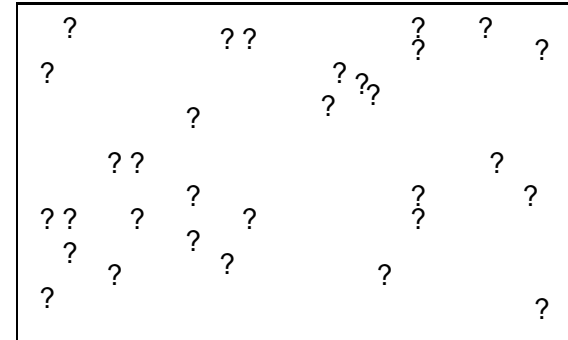
sense	training example
?	company said that the <i>plant</i> is still operating
?	although thousands of <i>plant</i> and animal species
?	zonal distribution of <i>plant</i> life
?	company manufacturing <i>plant</i> is in Orlando etc

2. Identify some seeds to disambiguate a few uses. e.g., 'plant life' for vegetation use (A) 'manufacturing plant' for factory use (B):

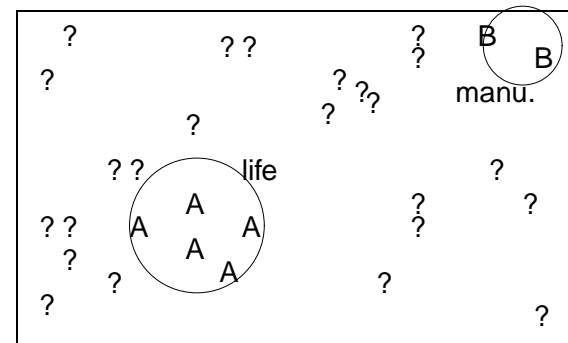
sense	training example
?	company said that the <i>plant</i> is still operating
?	although thousands of <i>plant</i> and animal species
A	zonal distribution of <i>plant</i> life
B	company manufacturing <i>plant</i> is in Orlando etc

Yarowsky (1995): schematically

Initial state



## Seeds



3. Train a **decision list** classifier on the Sense A/Sense B examples.

reliability	criterion	sense
8.10	<i>plant</i> life	A
7.58	manufacturing <i>plant</i>	B
6.27	<i>animal</i> within 10 words of <i>plant</i>	A
	etc	

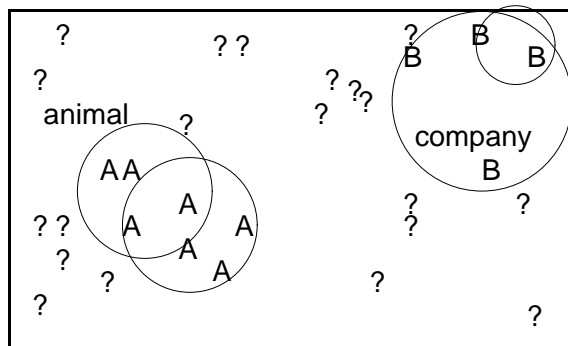
Decision list classifier: automatically trained if/then statements. Experimenter decides on classes of test by providing definitions of features of interest: system builds specific tests and provides reliability metrics.

4. Apply the classifier to the training set and add reliable examples to A and B sets.

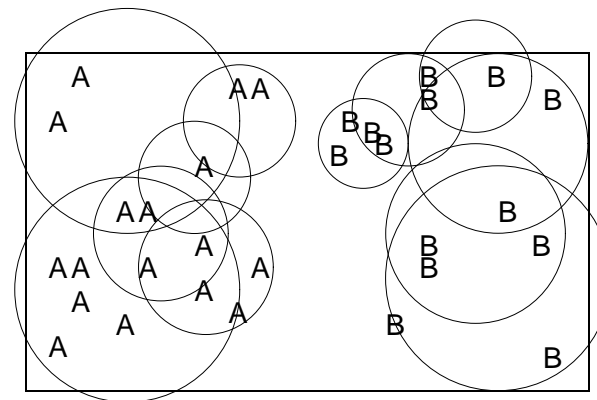
sense	training example
?	company said that the <i>plant</i> is still operating
A	although thousands of <i>plant</i> and animal species
A	zonal distribution of <i>plant</i> life
B	company manufacturing <i>plant</i> is in Orlando
	etc

5. Iterate the previous steps 3 and 4 until convergence

Iterating:



Final:



## 6. Apply the classifier to the unseen test data

‘one sense per discourse’: can be used as an additional refinement

e.g., once you’ve disambiguated *plant* one way in a particular text/section of text, you can assign all the instances of *plant* to that sense

## Outline of next lecture

Putting sentences together (in text).

### Lecture 7: Discourse

Relationships between sentences

Coherence

Anaphora (pronouns etc)

Algorithms for anaphora resolution

## Evaluation of WSD

- ▶ SENSEVAL competitions
- ▶ evaluate against WordNet
- ▶ baseline: pick most frequent sense — hard to beat (but don’t always know most frequent sense)
- ▶ human ceiling varies with words
- ▶ MT task: more objective but sometimes doesn’t correspond to polysemy in source language

## Outline of today’s lecture

Putting sentences together (in text).

### Lecture 7: Discourse

Relationships between sentences

Coherence

Anaphora (pronouns etc)

Algorithms for anaphora resolution



## Document structure and discourse structure

- ▶ Most types of document are highly structured, implicitly or explicitly:
  - ▶ Scientific papers: conventional structure (differences between disciplines).
  - ▶ News stories: first sentence is a summary.
  - ▶ Blogs, etc etc
- ▶ Topics within documents.
- ▶ Relationships between sentences.

## Coherence

Discourses have to have connectivity to be coherent:

Kim got into her car. Sandy likes apples.

Can be OK in context:

## Rhetorical relations

Max fell. John pushed him.

can be interpreted as:

1. Max fell because John pushed him.

EXPLANATION

or

- 2 Max fell and then John pushed him.

## NARRATION

Implicit relationship: **discourse relation** or **rhetorical relation**  
*because, and then* are examples of **cue phrases**

## Coherence

Discourses have to have connectivity to be coherent:

Kim got into her car. Sandy likes apples.

Can be OK in context:

Kim got into her car. Sandy likes apples, so Kim thought she'd go to the farm shop and see if she could get some.

## Coherence in generation

Strategic generation: constructing the logical form. Tactical generation: logical form to string.

Strategic generation needs to maintain coherence.

In trading yesterday: Dell was up 4.2%, Safeway was down 3.2%, HP was up 3.1%.

Better:

Computer manufacturers gained in trading yesterday: Dell was up 4.2% and HP was up 3.1%. But retail stocks suffered: Safeway was down 3.2%.

So far this has only been attempted for limited domains: e.g. tutorial dialogues.

## Coherence in interpretation

Discourse coherence assumptions can affect interpretation:

Kim's bike got a puncture. She phoned the AA.

Assumption of coherence (and knowledge about the AA) leads to *bike* interpreted as motorbike rather than pedal cycle.

John likes Bill. He gave him an expensive Christmas present.

If EXPLANATION - 'he' is probably Bill.

If JUSTIFICATION (supplying evidence for first sentence), 'he' is John.

## Factors influencing discourse interpretation

1. Cue phrases.
2. Punctuation (also prosody) and text structure.

Max fell (John pushed him) and Kim laughed.  
Max fell, John pushed him and Kim laughed.

3. Real world content:

Max fell. John pushed him as he lay on the ground.

4. Tense and aspect.

Max fell. John had pushed him.  
Max was falling. John pushed him.

Hard problem, but 'surfacy techniques' (punctuation and cue phrases) work to some extent.

## Rhetorical relations and summarization

Analysis of text with rhetorical relations generally gives a binary branching structure:

- ▶ **nucleus** and **satellite**: e.g., EXPLANATION, JUSTIFICATION
- ▶ equal weight: e.g., NARRATION

Max fell because John pushed him.

## Rhetorical relations and summarization

Analysis of text with rhetorical relations generally gives a binary branching structure:

- ▶ **nucleus** and **satellite**: e.g., EXPLANATION, JUSTIFICATION
- ▶ equal weight: e.g., NARRATION

Max fell because John pushed him.

## Summarisation by satellite removal

If we consider a discourse relation as a relationship between two phrases, we get a binary branching tree structure for the discourse. In many relationships, such as Explanation, one phrase depends on the other: e.g., the phrase being explained is the main one and the other is subsidiary. In fact we can get rid of the subsidiary phrases and still have a reasonably coherent discourse.

## Referring expressions

Niall Ferguson is prolific, well-paid and a snappy dresser.  
Stephen Moss hated him — at least until he spent an hour  
being charmed in the historian's Oxford study.

**referent** a real world entity that some piece of text (or speech) refers to. **the actual Prof. Ferguson**

**referring expressions** bits of language used to perform reference by a speaker. 'Niall Ferguson', 'he', 'him'

**antecedent** the text initially evoking a referent. 'Niall Ferguson'

**anaphora** the phenomenon of referring to an antecedent.

## Pronoun resolution

Pronouns: a type of anaphor.

Pronoun resolution: generally only consider cases which refer to antecedent noun phrases.

Niall Ferguson is prolific, well-paid and a snappy dresser.  
Stephen Moss hated him — at least until he spent an hour  
being charmed in the historian's Oxford study.

## Pronoun resolution

Pronouns: a type of anaphor.

Pronoun resolution: generally only consider cases which refer to antecedent noun phrases.

Niall Ferguson is prolific, well-paid and a snappy dresser. Stephen Moss hated him — at least until he spent an hour being charmed in the historian's Oxford study.

## Pronoun resolution

Pronouns: a type of anaphor.

Pronoun resolution: generally only consider cases which refer to antecedent noun phrases.

Niall Ferguson is prolific, well-paid and a snappy dresser. Stephen Moss hated him — at least until he spent an hour being charmed in the historian's Oxford study.

## Hard constraints: Pronoun agreement

- ▶ My dog has hurt his foot — he is in a lot of pain.
- ▶ \* My dog has hurt his foot — it is in a lot of pain.

Complications:

- ▶ The team played really well, but now they are all very tired.
- ▶ Kim and Sandy are asleep: they are very tired.
- ▶ Kim is snoring and Sandy can't keep her eyes open: they are both exhausted.

## Hard constraints: Reflexives

- ▶ John<sub>i</sub> washes himself<sub>i</sub>. (himself = John, subscript notation used to indicate this)
- ▶ \* John<sub>i</sub> washes him<sub>j</sub>.

Reflexive pronouns must be coreferential with a preceding argument of the same verb, non-reflexive pronouns cannot be.



## Anaphora resolution as supervised classification

- ▶ Classification: training data labelled with class and features, derive class for test data based on features.
- ▶ For potential pronoun/antecedent pairings, class is TRUE/FALSE.
- ▶ Assume candidate antecedents are all NPs in current sentence and preceeding 5 sentences (excluding pleonastic pronouns)

## Example

Niall Ferguson is prolific, well-paid and a snappy dresser.  
Stephen Moss hated him  
— at least until he spent an hour being charmed in the  
historian's Oxford study.

Issues: detecting pleonastic pronouns and predicative NPs, deciding on treatment of possessives (*the historian* and *the historian's Oxford study*), named entities (e.g., *Stephen Moss*, not *Stephen* and *Moss*), allowing for cataphora, ...

## Example

Niall Ferguson is prolific, well-paid and a snappy dresser. Stephen Moss hated him — at least until he spent an hour being charmed in the historian's Oxford study.

Issues: detecting pleonastic pronouns and predicative NPs, deciding on treatment of possessives (*the historian* and *the historian's Oxford study*), named entities (e.g., *Stephen Moss*, not *Stephen* and *Moss*), allowing for cataphora, ...

## Features

**Cataphoric** Binary: t if pronoun before antecedent.

**Number agreement** Binary: t if pronoun compatible with antecedent.

Gender agreement Binary: t if gender agreement.

**Same verb** Binary: t if the pronoun and the candidate antecedent are arguments of the same verb.

Sentence distance Discrete:  $\{0, 1, 2 \dots\}$

**Grammatical role** Discrete: { subject, object, other } The role of the potential antecedent.

**Parallel Binary:** t if the potential antecedent and the pronoun share the same grammatical role.

**Linguistic form** Discrete: { proper, definite, indefinite, pronoun }

## Feature vectors

pron	ante	cat	num	gen	same	dist	role	par	form
<i>him</i>	<i>Niall F.</i>	f	t	t	f	1	subj	f	prop
<i>him</i>	<i>Ste. M.</i>	f	t	t	t	0	subj	f	prop
<i>him</i>	<i>he</i>	t	t	t	f	0	subj	f	pron
<i>he</i>	<i>Niall F.</i>	f	t	t	f	1	subj	t	prop
<i>he</i>	<i>Ste. M.</i>	f	t	t	f	0	subj	t	prop
<i>he</i>	<i>him</i>	f	t	t	f	0	obj	f	pron

## Training data, from human annotation

class	pron	ante	cat	num	gen	same	dist	role	par	form
TRUE	<i>him</i>	<i>Niall F.</i>	f	t	t	f	1	subj	f	prop
FALSE	<i>him</i>	<i>Ste. M.</i>	f	t	t	t	0	subj	f	prop
FALSE	<i>him</i>	<i>he</i>	t	t	t	f	0	subj	f	pron
FALSE	<i>he</i>	<i>Niall F.</i>	f	t	t	f	1	subj	t	prop
TRUE	<i>he</i>	<i>Ste. M.</i>	f	t	t	f	0	subj	t	prop
FALSE	<i>he</i>	<i>him</i>	f	t	t	f	0	obj	f	pron

## Naive Bayes Classifier

Choose most probable class given a feature vector  $\vec{f}$ :

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c|\vec{f})$$

Apply Bayes Theorem:

$$P(c|\vec{f}) = \frac{P(\vec{f}|c)P(c)}{P(\vec{f})}$$

Constant denominator:

$$\hat{c} = \operatorname{argmax}_{c \in C} P(\vec{f}|c)P(c)$$

Independent feature assumption ('naive'):

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c) \prod_{i=1}^n P(f_i|c)$$

## Problems with simple classification model

- ▶ The problem is not really described well by pairs of "pron-ant"
- ▶ Cannot model interdependencies between features.
- ▶ Cannot implement 'repeated mention' effect.
- ▶ Cannot use information from previous links:

*Sturt think they can perform better in Twenty20 cricket. It requires additional skills compared with older forms of the limited over game.*

*it* should refer to Twenty20 cricket, but looked at in isolation could get resolved to *Sturt*. If linkage between *they* and *Sturt*, then number agreement is pl.

Would need **discourse model** with real world entities corresponding to clusters of referring expressions.

## Evaluation

Simple approach is link accuracy. Assume the data is previously marked-up with pronouns and possible antecedents, each pronoun is linked to an antecedent, measure percentage correct. But:

- ▶ Identification of non-pleonastic pronouns and antecedent NPs should be part of the evaluation.
- ▶ Binary linkages don't allow for chains:

*Sally met Andrew in town and took him to the new restaurant. He was impressed.*

Multiple evaluation metrics exist because of such problems.

## Outline of (rest of) today's lecture

### Lecture 8: An application – The FUSE project

Approach  
Example case RNAi  
(Some) Processing Steps  
Discourse Analysis  
A quick glance under the hood

## Classification in NLP

- ▶ Also sentiment classification, word sense disambiguation and many others. POS tagging (sequences).
- ▶ Feature sets vary in complexity and processing needed to obtain features. Statistical classifier allows some robustness to imperfect feature determination.
- ▶ Acquiring training data is expensive.
- ▶ Few hard rules for selecting a classifier: e.g., Naive Bayes often works even when independence assumption is clearly wrong (as with pronouns). Experimentation, e.g., with WEKA toolkit.

## The FUSE project – Foresight and Understanding from Scientific Exposition

- ▶ Funded by IARPA
- ▶ **Task:** Identify emerging ideas in the literature (given a certain time frame, and a set of scientific articles)
  - ▶ Example: Genetic Algorithms
  - ▶ Example: RNA interference
  - ▶ Counterexample: hot fusion
- ▶ Required: objective evidence collected from the articles



## Overview

- ▶ Team of 5 Universities—Columbia, Maryland, Washington State, Michigan, Cambridge
- ▶ 5 years, \$20 Million, roughly 30 people involved.
- ▶ Columbia: Project Management, Named Entity Recognition, Semantic Frames, Linguistic processing, Summarisation
- ▶ Maryland: Time Series Analysis, Machine Learning
- ▶ **Cambridge team: Discourse analysis, Sentiment Analysis**
- ▶ Washington: Chinese Processing
- ▶ Michigan: Citation Processing
- ▶ Showcase NLP technology, lower to higher level analysis

## Example sentences, case study “RNAi”

- ▶ 2001 article: Researchers have been critical of RNAi technology because of the concentration of RNAi necessary to have a therapeutic effect.
- ▶ 2004 article: Kits and reagents for making RNAi constructs are now widely available.
- ▶ 2010 **Press Release**: Collecta announces the DECIPHER project, an open-source platform for genome-wide RNAi screening and analysis.
- ▶ 2001 article: RNAi is now employed routinely across phyla, systematically analysing gene function in most organisms with complete genomic sequences.
- ▶ 2004 article: RNAi has quickly become one of the most powerful and indispensable tools in the molecular biologist's toolbox.
- ▶ 2003 article: cheaper and faster than knockout mice
- ▶ 2004 article: Previous RNA-based technologies mentioned are antisense and ribozymes. Advantages of RNAi: greater potency; taps into already-existing control system within the cell (i.e., is natural).
- ▶ 2009 article: “RNAi has rapidly become a standard method for experimental and therapeutic gene silencing, and has moved from bench to bedside at unprecedented speed.”

## Approach

- ▶ Apply Citation Analysis to find clusters of papers interested in one topic
- ▶ Process citations, surprising noun phrases around citations
- ▶ Understand important relationships the noun phrases participate in
- ▶ Sentiment towards ideas
- ▶ Collect Indicators: Frequency, type of statement, length of statement
- ▶ Machine learn “emergence events”
- ▶ Express justification based on indicators in a summary

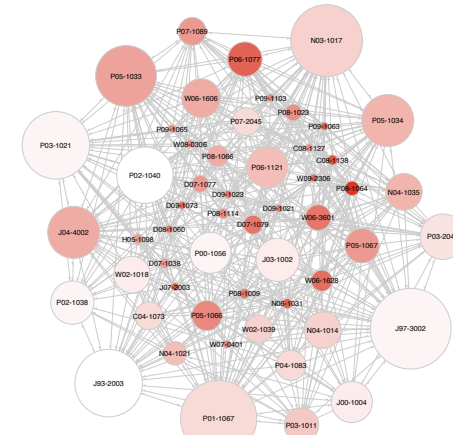
## Titles expressing sentiment towards RNAi

- ▶ “Unlocking the money-making potential of RNAi”
- ▶ “Drugmakers’ fever for the power of RNA interference has cooled”
- ▶ “Are early clinical successes enough to bring RNAi back from the brink?”
- ▶ “The promises and pitfalls of RNA-interference-based therapeutics”

## Processing

- ▶ Tokenisation, Citation Detection, Sentence Boundary Detection
- ▶ Morphological Analysis
- ▶ Named Entity Detection
- ▶ POS tagging
- ▶ Parsing (Stanford Parser)
- ▶ Pronoun Resolution
- ▶ Sentiment Analysis of Citations
- ▶ Lexical Similarity between Verb Semantics
- ▶ Coherent Functional Segments in Text

## Citation Analysis



## Discourse Analysis (chemistry)

# Synthesis of pyrazole and pyrimidine Troeger's base analogues

1  
P  
E  
K  
I  
N

Rodrigo Abonia, Andrea Albernez, Hector Larrabondo, Jairo Quiroga, Braulio Isuasty, Henry Isuasty, Angelina Hormaza Adolfo Sanchez, and Manuel Nogueiras

Troeger's base analogues bearing fused pyrazolic or pyrimidinic rings were prepared in acceptable to good yields through the reaction of 3-alkyl-5-amino-1-arylpyrazoles and 6-amino-pyrimidin-4(3H)-ones with formaldehyde under mild conditions (i.e. in ethanol at 50°C in the presence of catalytic amounts of acetic acid). Two key intermediates were isolated from the reaction mixtures, which helped us to suggest a sequence of steps for the formation of the Troeger's bases obtained. The structures of the products were assigned by <sup>1</sup>H and <sup>13</sup>C NMR, mass spectra and elemental analysis and confirmed by X-ray diffraction for one of the obtained compounds.

## Introduction

Although the first Troeger's base 1 was obtained more than a century ago from the reaction of p-toluidine and formaldehyde [1], recently the study of these compounds has gained importance due to their potential applications. They possess a relatively rigid chiral structure which makes them suitable for the development of possible synthetic enzymes and artificial receptor systems [2], chelating and biomimetic systems [3] and transition metal complexes for regio- and stereoselective catalytic reactions [4]. For these reasons, numerous Troeger's base derivatives have been prepared bearing different types of substituents and structures (i.e. 2-5 Scheme 1), with the purpose of increasing their potential applications [2,3,5].

Scheme 1 The original Troeger's base 1 and some interesting derivatives and analogues.

However, some of the above methodologies possess tedious work-up procedures or include relatively strong reaction conditions, such as treatment of the starting materials for several hours with an ethanolic solution of conc. hydrochloric acid or TFA solution, with poor to moderate yields, as is the case for analogues 4 and 5 [5].

Considering these potential applications, we now report a simple synthetic method for the preparation of 5,12-di-alkyl-3,10-di-aryl-1,3,4,8,10,11-hexa-azetetracyclo[6.6.1.0.2,6.0.9,13]heptaecata-(2-6),4,9(13)-11-tetraenes 8a-e and 4,12-dimethoxy-1,3,5,9,11,13-hexaazetetracyclo[7.7.1.0.2,7.0.10,15.1]heptaecata(2,7),3,10(15)-11-tetraenes-6,14-diones 10a,b based on the reaction of 3-alkyl-5-amino-1-arylpyrazoles and 6-amino-pyrimidin-4(3H)-ones with formaldehyde in ethanol and catalytic amounts of acetic acid. Compounds 8 and 10 are new Troeger's base analogues bearing heterocyclic rings instead of the usual phenyl rings in their aromatic parts.

## Results and discussion

In an attempt to prepare the benzotriazolyl derivative 7a, which could be used as an intermediate in the synthesis of new hydroquinolines of interest, [6], a mixture of 5-amino-3-methyl-1-phenylpyrazole 6a, formaldehyde and benzotriazole in 10 ml of ethanol, with catalytic amounts of acetic acid, was heated at 50°C for 5 minutes. A solid precipitated from the solution, while it was still hot. However, no consumption of benzotriazole was observed at TLC.

The reaction conditions were modified and the same product was obtained when the reaction was carried out without using benzotriazole, as shown in Scheme 12. On the basis of NMR and mass spectra and X-ray crystallographic analysis we established that the structure is 5,12-di-alkyl-3,10-di-aryl-1,3,4,8,10,11-hexapentacyclic Troeger's base analogue.

Co\_Gro

Other

Aim

Gap/Weak

Own\_Mhd

Own\_Res

Own\_Conc

## Discourse Analysis (comp. ling.)

# Distributional Clustering of English Words

Fernando Pereira

Naftali Tishby

Lillian Lee

Abstract

We describe and experimentally evaluate a method for automatically clustering words according to their distribution in particular syntactic contexts. Deterministic annealing is used to find lowest distortion sets of clusters. As the annealing parameter increases, existing clusters become unstable and subdivide, yielding a hierarchical "soft" clustering of the data. Clusters are used as the basis for class models of word occurrence, and the models evaluated with respect to held-out test data.

Introduction

Methods for automatically classifying words according to their contexts of use have both scientific and practical interest. The scientific questions arise in connection to distributional views of linguistic (particularly lexical) structure and also in relation to test question of lexical acquisition both from psychological and computational learning perspectives. From the practical point of view, word classification addresses questions of data sparseness and generalization in statistical language models, particularly models for deciding among alternatives analyses proposed by a grammar.

It is well-known that a simple tabulation of frequencies of certain words participating in certain configurations, for example the frequencies of pairs of a transitive main verb and the head noun of its direct object, cannot be reliably used for comparing the likelihood of different alternative configurations. The problem is that for large enough corpora the number of joint events is much larger than the number of event occurrences in the corpus, so many events are seen rarely or never, making their frequency counts unreliable estimates of their probabilities.

Hindle (1999) proposed dealing with the sparseness problem by estimating the likelihood of unseen events from that of "similar" events that have been seen. For instance, one may estimate the likelihood of a particular direct object of a verb from the likelihoods of that direct object for similar verbs. This requires a reasonable definition of verb similarity and a similarity estimation method. In Hindle's proposal, words are similar if we have strong

statistical evidence that they tend to participate in the same events. His notion of similarity seems to agree with our intuitions in many cases, but it is not clear how it can be used directly to construct word classes and corresponding models of association.

Our research addresses some of the same questions and uses similar raw data, but we investigate how to factor word association tendencies into associations of words to certain hidden base classes and associations between the classes themselves. While it may be worthwhile to base such a model on preexisting classes (Resnik, 1992), in the work described here we look at how to derive the classes directly from distributional data. More specifically, we model nouns as probabilistic concepts or clusters with corresponding cluster membership probabilities  $p(c|w)$  for each word  $w$ . Most other class-based modeling techniques for natural language rely on "hard" Boolean classes (Brown et al., 1990). Class construction is then combinatorially very demanding and depends on frequency counts for joint events involving particular words, a potentially unreliable source of information as we noted above. Our approach avoids both problems.

Problem Setting

In what follows, we will consider two major word classes,  $V$  and  $N$ , for the verbs and nouns in our experiments, and a single relation between them, in our experiments the relation between a transitive main verb and the head noun of its direct object. Our raw knowledge about the relation consists of the frequencies  $f(v, n)$  of occurrence of particular pairs  $(v, n)$  in the required configuration in our corpus. Some form of text analysis is required to collect such a collection of pairs. The corpus used in our first experiment was derived from newswire text automatically parsed by Hindle's parser Fiddish (Hindle, 1996). More recently, we have constructed similar tables with the help of a statistical part-of-speech tagger (Church, 1988) and of tools for regular expression pattern matching on tagged corpora (Yarowsky, 1992). We have not yet compared the accuracy and coverage of the two methods, or what systematic biases they might introduce, although we took care to filter out certain systematic errors, for instance the mis-parsing of the subject of a complement clause as the direct object of a main verb for our test verbs like "say".

We will consider here only the problem of classifying nouns according to their distribution as direct objects of verbs; the converse problem is formally similar. More generally, the theoretical bias for our methods supports the use of clustering to build models for any  $n$ -ary

## Drilling down: Agents

(we/I)  
 (we/I) also  
 (we/I) now  
 (we/I) here  
 (our/my) JJ\* (account/ algorithm/ analysis/ analyses/ approach/ application/ architecture...)  
 (our/my) JJ\* (article/ draft/ paper/ project/ report/ study)  
 (our/my) JJ\* (assumption/ hypothesis/ hypotheses/ claim/ conclusion/ opinion/ view)  
 (our/my) JJ\* (answer/ accomplishment/ achievement/ advantage/ benefit...)  
 (account/ ...) (noted/ mentioned/ addressed/ illustrated ...) (here/below)  
 (answer/ ...) given (here/below)  
 (answer/ ...) given in this (article/ ...)  
 (first/second/third) author  
 one of us  
 one of the authors

## Drilling down: Verb Semantics

Action Type	Example	Others
AFFECT	we <i>hope</i> to improve our results	feel, trust...
ARGUMENTATION	we <i>argue</i> against a model of	advocate, defend...
AWARENESS	we <i>are not aware of</i> attempts	do not know of
BETTER_SOLUTION	our system <i>outperforms</i> ...	defeat, surpass...
CHANGE	we <i>extend</i> CITE's algorithm	adapt, expand...
COMPARISON	we <i>tested</i> our system against...	compete, evaluate...
CONTINUATION	we <i>follow</i> Sag (1976) ...	borrow from, build on...
CONTRAST	our approach <i>differs from</i> ...	distinguish, contrast...
FUTURE_INTEREST	we <i>intend</i> to improve ...	plan, expect...
INTEREST	we <i>are concerned with</i> ...	focus on, be motivated by...
NEED	this approach, however, <i>lacks</i> ...	needs, requires, be reliant on...
PRESENTATION	we <i>present</i> here a method for...	point out, recapitulate
PROBLEM	this approach <i>fails</i> ...	is troubled by, degrade...
RESEARCH	we <i>collected</i> our data from...	measure, calculate...
SIMILAR	our approach <i>resembles</i> that of	bear comparison, have much in common with...
SOLUTION	we <i>solve</i> this problem by...	alleviate, circumvent...
TEXTSTRUCTURE	the paper <i>is organized</i> ...	begin by, outline
USE	we <i>employ</i> Suzuki's method...	employ, apply, make use of...