Accelerated Natural Language Processing 2016

Lecture 17: Words in PCFGs, collocations and mutual information

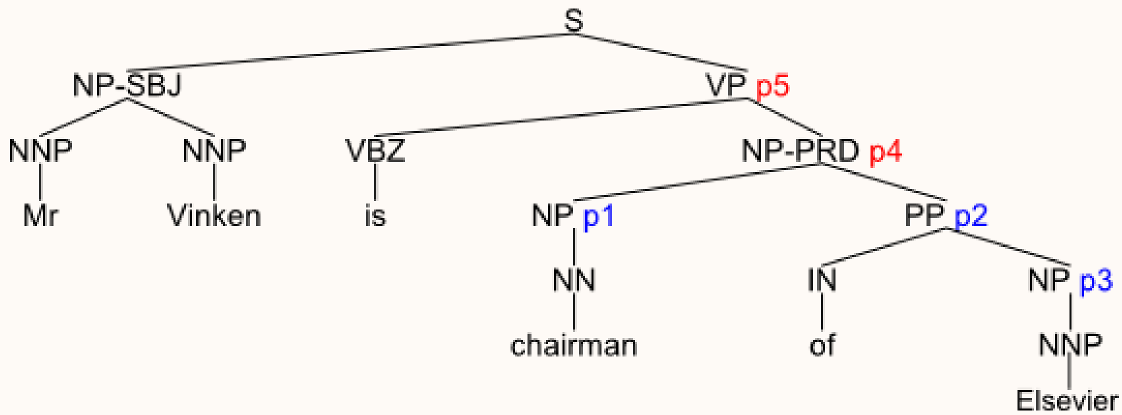
Henry S. Thompson
7 November 2016

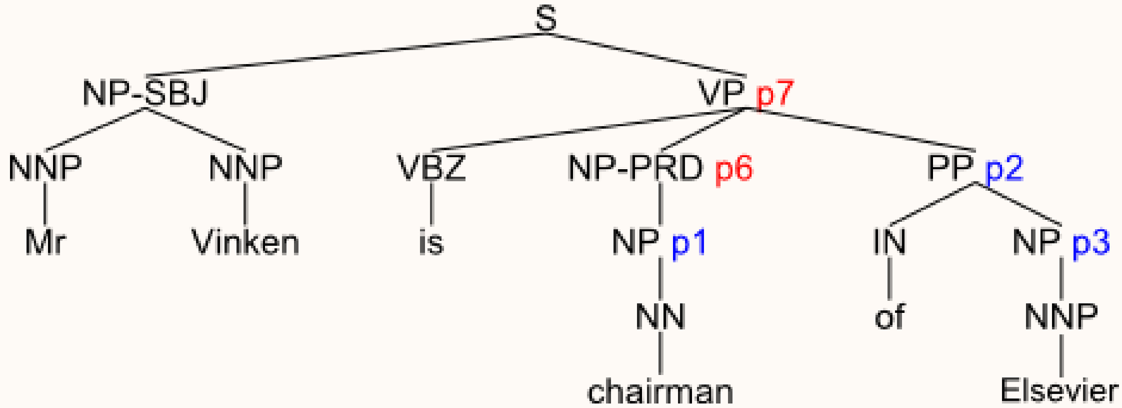
1. Another big problem with simple PCFGs

For a given structural ambiguity, say PP attachment

- A simple PCFG *always* makes the same choice

Consider the two alternative parses we would get from the Treebank grammar for *Mr Vinken is chairman of Elsevier*:





2. PCFG example, cont'd

How did we get those two analyses?

- Here's a top-down parse trace using Penn Treebank rules:

PP attached to NP-PRD	PP attached to VP
S → NP-SBJ VP	S → NP-SBJ VP
NP-SBJ → NNP NNP	NP-SBJ → NNP NNP
NNP → Mr	NNP → Mr
NNP → Vinken	NNP → Vinken
p5 **VP → VBZ NP-PRD**	p7 **VP → VBZ NP-PRD PP**
VBZ → is	VBZ → is
p4 **NP-PRD → NP PP**	p6 **NP-PRD → NP**
NP → NN	NP → NN
p1 NN → chairman	p1 NN → chairman
p3 PP → IN NP	p3 PP → IN NP
IN → of	IN → of
NP → NNP	NP → NNP
p2 NNP → Elsevier	p2 NNP → Elsevier

So the only difference is the probabilities of the rules highlighted in red

- p4 and p5 versus p6 and p7

And those will *always* give the same answer

- Depending on the relative magnitude of p4*p5 vs p6*p7
- Regardless of what the individual words and their probabilities are
- Because whatever the words
- p1, p2 and p3 are the *same* for both parses
- Compare, for instance, *chairman at the moment* with *chairman for life*

We really need to pay attention to the probability of those words being tightly or loosely connected

3. Paying attention to words

Improving our approach to probabilistic grammar requires paying more attention to individual *words*

- Let's explore how we might do that a bit more

Back to bigrams, but in a little more detail

- It turns out that not all bigrams are created equal

Here are the top 10 bigrams from Herman Melville's famous American novel *Moby Dick*:

of the (1879)
in the (1182)
to the (731)
from the (440)
the whale (407)
of his (373)
and the (370)
on the (359)
of a (334)
at the (330)
to be (329)

Aside from *the whale*, these are all made up of very high-frequency closed-class items

The highest bigram of open-class items doesn't come until position 27: *sperm whale*, with frequency 182

None-the-less if feels as if there's something particularly interesting about that one. . .

4. Collocations

"You shall know a word by the company it keeps" (J. R. Firth)

One of the things we evidently know about our language is what words go with what

- *strong tea*, not *powerful tea*
- *powerful enemies*, not *strong enemies*
- *tall flagpole*, not *long flagpole*
- *long railing*, not *tall railing*

Choosing the right word from among a set of synonyms is a common problem for second-language learners

- *un froid vive* is not (per Microsoft translate) *a bright cold*, but rather *a piercing cold*
- Likewise *универсальный магазин* is not *universal store* but *department store*

Or consider the old (linguists') joke:

- *If a maternity dress is what a woman wears when she is pregnant, what is a paternity suit?*

5. What measure to find collocations?

The name for an 'interesting' pair is **collocation**

How can we separate the interesting pairs from the dull ones?

We could try just throwing out the 'little words'

- Typically called 'stop words', see e.g. `corpus.stopwords.word('english')` in NLTK
- But that still isn't quite getting at what we want:
  sperm whale (182)

white whale (106)
moby dick (84)
old man (81)
captain ahab (62)
right whale (57)
mast head (49)
mast heads (37)
ye see (35)
whale ship (34)

Some of these feel special (*right whale*, *moby dick*), but others (*old man* in particular) just seem ordinary

- They are **encoding transparent**: we would expect their meaning to be formed from those words
- They're not 'terms' in the sense of 'terminology'

6. Normalising by expection: mutual information

What we want is some way of factoring in frequency more generally

- To take care of more than just some pre-determined list of stop words
- and to downgrade *old man* and similar cases

Conditional and joint probability are the answer

- Remember that if two events are independent, the conditional probability is the same as the unconditional
  - **P(Y|X)** is just **P(Y)**
- so the joint probability is just the product of the two event probabilities
  - **P(X,Y)** is defined as **P(X)P(Y|X)**
  - So if **X** and **Y** are independent, that's **P(X)P(Y)**

One way of getting at our intuition might be to say we're looking for cases where the two probabilities are *not* independent

Now the bigram frequency gives us an MLE of the joint probability directly

So the *ratio* of that probability, to what it would be if they were independent, would be illuminating:

Pointwise mutual information

$$\log_2\left(\frac{P(X, Y)}{P(X)P(Y)}\right)$$

Terminology note: Strictly speaking we should distinguish between ***pointwise* mutual information** and **mutual information** as such. The latter is a measure over *distributions*, as opposed to individuals.

- But below, and often in the literature, we will use **mutual information** for the measure defined above, when it is clear from context that it's individuals we're concerned with.

7. Mutual information example

Let's compare the most frequent bigram (*of the*) with the first interesting one we saw, *sperm whale*

```
>>> f[('of', 'the')]
1879
>>> u['of']
6609
>>> u['the']
14431
>>> 1879.0/218360
0.0086050558710386513
>>> (6609.0 * 14431)/(218361*218361)
0.0020002396390988637
>>> log((1879.0/218360)/((6609.0 * 14431)/(218361*218361)),2)
2.105011706733956

>>> f[('sperm','whale')]
182
>>> u['sperm']
244
>>> u['whale']
1226
>>> 182.0/218361
0.00083348216943501818
>>> (244.0*1226)/(218361*218361)
6.2737924534150313e-06
>>> log((182.0/218361)/((244.0*1226)/(218361*218361)),2)
7.0536697225202696
```

Simply put, the mutual information between *sperm* and *whale* is 5 binary orders of magnitude greater than that between *of* and *the*

Why are we using log base 2?

- Because we traditionally measure information in bits

8. Collocations and machine translation

We can use the same approach to build a translation lexicon

Instead of bigrams within a single text

- Co-occurence in aligned units of a bilingual text
- We treat sentences as sets of words
- And construct a distribution of pairs of words, one from each set for each pair of sentences
- Here are the top 30 pairs from a corpus of 4600 sentence pairs from the European Parliament debates, French and English versions:

```
[((u'commission', u'commission'), 113),
 ((u'rapport', u'report'), 84),
 ((u'régions', u'regions'), 71),
 ((u'parlement', u'parliament'), 66),
 ((u'politique', u'policy'), 62),
 ((u'voudrais', u'like'), 58),
 ((u'président', u'president'), 57),
 ((u'fonds', u'funds'), 52),
 ((u'monsieur', u'president'), 50),
 ((u'union', u'union'), 48),
 ((u'états', u'states'), 46),
 ((u'membres', u'member'), 46),
 ((u'états', u'member'), 46),
 ((u'développement', u'development'), 44),
 ((u'membres', u'states'), 43),
 ((u'également', u'also'), 42),
 ((u'structurels', u'structural'), 41),
 ((u'fonds', u'structural'), 41),
 ((u'structurels', u'funds'), 40),
 ((u'cohésion', u'cohesion'), 38),
 ((u'voudrais', u'would'), 38),
 ((u'européenne', u'european'), 37),
 ((u'orientations', u'guidelines'), 37),
 ((u'commission', u'would'), 36),
 ((u'madame', u'president'), 34),
 ((u'groupe', u'group'), 33),
 ((u'commissaire', u'commissioner'), 33),
 ((u'présidente', u'president'), 32),
 ((u'sécurité', u'safety'), 32),
 ((u'transports', u'transport'), 30)]
```

Pretty good

And would be better if we had done monolingual collocation detection first!

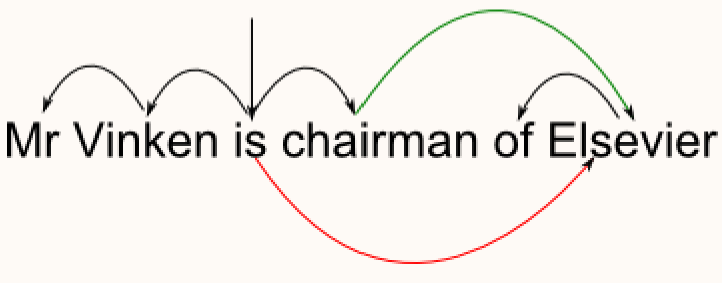
9. Words, heads and grammar

We mentioned the idea of the **head** of a constituent in earlier lectures

- Maybe organising our grammar in a completely different way would help with the attachment problem
- Because something like mutual information between *heads* might be just what we need

Approaches to grammar which focus on heads are called **dependency** grammars

The standard form of diagram shows where this name comes from:



(Green for the preferred attachment, red for the less likely one)

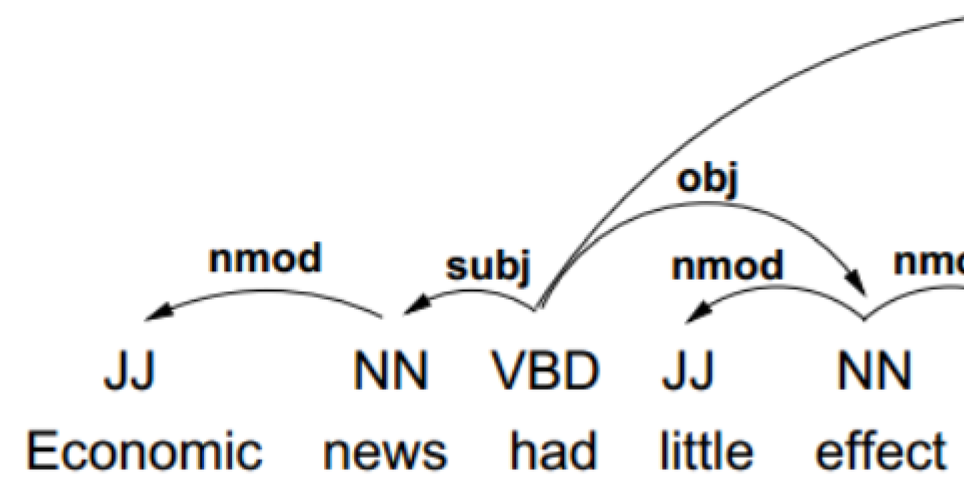
10. Dependency grammar

Dependency grammars don't have rules in the way that a CFG does

- But they do have categories
- And category-specific ways of identifying heads

A given approach to dependency grammar will also involve an *inventory* of relations

- Not just 'depends'
- But e.g. subj, obj, nmod, pmod:



[From Joakim Nivre, Dependency Grammar and Dependency Parsing]

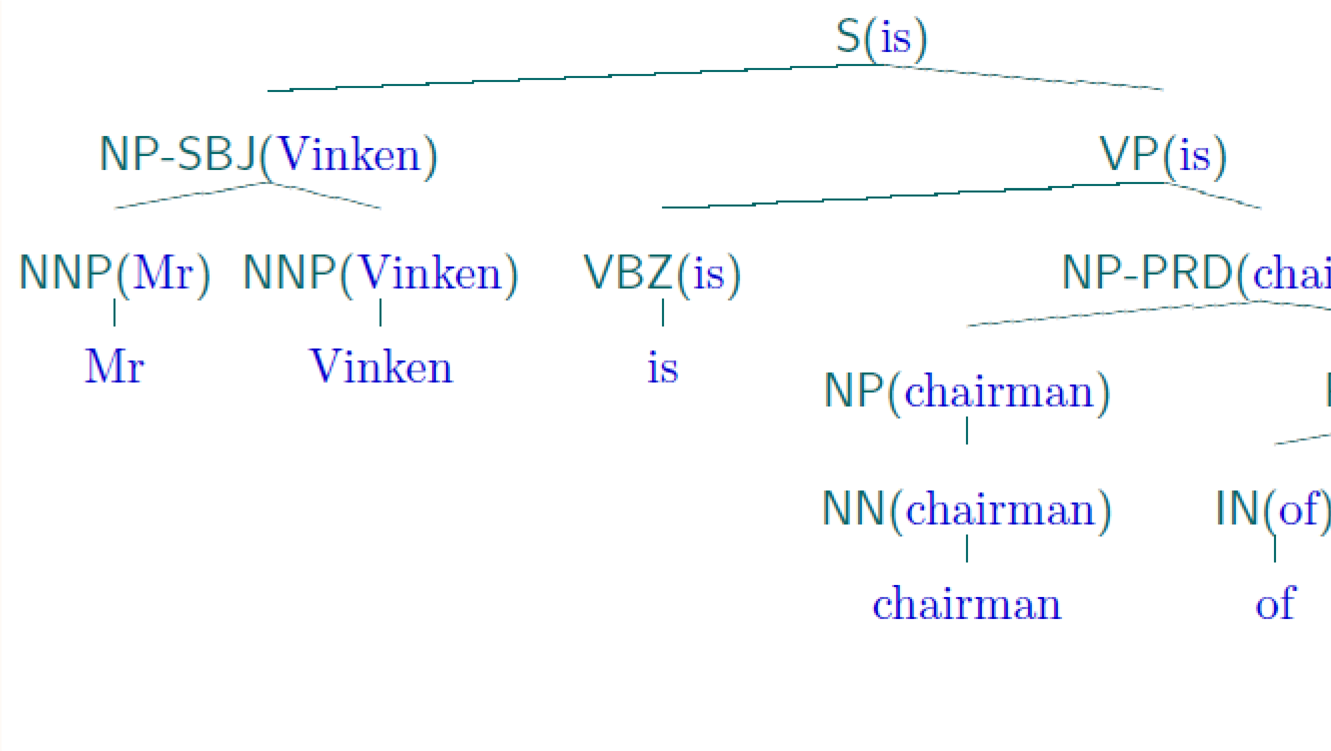
Dependency graphs are *not* required to avoid crossings

- So they can directly represent long-distance dependencies
- And work well for case-marked, free-constituent-order languages

11. Lexicalised PCFG

It's possible to add some of the benefits of dependency grammar to PCFGs

- By decorating categories in the tree with their head



We can't do statistics directly on these augmented categories

- The space is too large to give anything other than very small numbers

But a range of techniques have been developed in the last few years to work around this

- Sharon will come back to this in a few weeks

12. Putting words first: Categorial grammar

Categorial grammar represents a different approach to putting words at the centre of things

- Albeit one which still gives linear order a key role

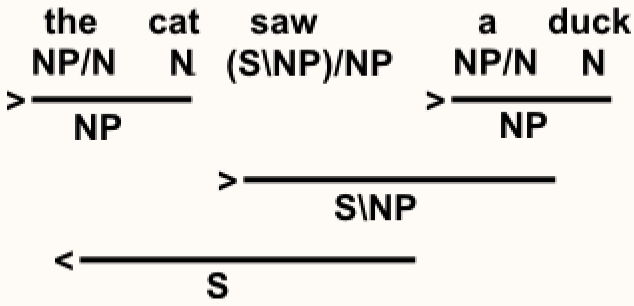
In the simplest form of CG, all we need is a lexicon like this

```
N: duck, cat, ...
NP/N: the, a, ...
S\NP: ran, slept, ...
(S\NP)/NP: saw, liked
```

Where we read e.g. `NP/N` as the category for things which combine with an `N` to their *right* to produce an `NP`

- And `S\NP` as the category for things which combine with an `NP` to their *left* to produce an `S`

In the obvious way this gives us the following derivation for *the cat saw a duck*:



The arrows next to the derivation steps identify which of the (meta-)rules was used for that step:

forward combination (>)

`X/Y Y → X`

backward combination (<)

`Y X\Y → X`

These are the only two rules, or rule schemata, needed for the simplest categorial grammars