The move followed a round of similar increases by other lenders, reflecting a continuing decline in that market.
Constituency Tests

- How do we know what nodes go in the tree?
- Classic constituency tests:
  - Substitution by proform
  - Question answers
  - Semantic grounds
    - Coherence
    - Reference
    - Idioms
  - Dislocation
  - Conjunction
- Cross-linguistic arguments, too

Conflicting Tests

- Constituency isn’t always clear
  - Units of transfer:
    - think about ~ penser à
    - talk about ~ hablar de
  - Phonological reduction:
    - I will go → I’ll go
    - I want to go → I wanna go
    - a le centre → au centre
  - Coordination
    - He went to and came from the store.

Questions from Last Time

- Q: Do we model deep vs surface structure?

[Example: Johnson 02]
### Parts-of-Speech (English)

- **Open class (lexical) words**
  - **Nouns**
    - Proper: IBM, Italy
    - Common: cat / cats, snow
  - **Verbs**
    - Main: see, registered
    - Auxiliary: can, had
  - **Adjectives**
    - yellow
  - **Adverbs**
    - slowly
  - **Prepositions**
    - to, with
  - **Particles**
    - off, up
  - **Determiners**
    - the, some
  - **Conjunctions**
    - and, or
  - **Pronouns**
    - he, its

- **Closed class (functional)**
  - **Numbers**
    - 122,312
  - **Conjunctions**
    - and, or

### Ambiguities

- Words can have multiple parts of speech
  
- Two basic sources of constraint:
  - Grammatical environment
  - Identity of the current word
- Many more possible features:
  - Suffixes, capitalization, name databases (gazetteers), etc...

---

Fed raises interest rates 0.5 percent

Mrs./NPP Shefer/NNP never/RB got/VBD around/RB to/TO joining/VBG
All/DT we/PRP gonna/VBN do/VB is/VBZ go/VB around/RB the/DT corner/NN
Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD
Why POS Tagging?

- Useful in and of itself (more than you’d think)
  - Text-to-speech: record, lead
  - Lemmatization: saw[v] → see, saw[n] → saw
  - Quick-and-dirty NP-chunk detection: grep (JJ | NN)* (NN | NNS)

- Useful as a pre-processing step for parsing
  - Less tag ambiguity means fewer parses
  - However, some tag choices are better decided by parsers

  DT NNP NN VBD VBN RP NN NNS
  The Georgia branch had taken on loan commitments …

  IN DT NN IN NN VBD NNS VBD
  The average of interbank offered rates plummeted …

Classical NLP: Parsing

- Write symbolic or logical rules:
  
  Grammar (CFG)
  Lexicon
  
  ROOT → S
  S → NP VP
  NP → DT NN
  VP → VBP NP
  VBP → raises
  NN → interest
  VBN → raises
  VBD
  VBN

- Use deduction systems to prove parses from words
  - Minimal grammar on “Fed raises” sentence: 36 parses
  - Simple 10-rule grammar: 592 parses
  - Real-size grammar: many millions of parses

- This scaled very badly, didn’t yield broad-coverage tools

Ambiguities: PP Attachment

I cleaned the dishes from dinner
I cleaned the dishes with detergent
I cleaned the dishes in my pajamas
I cleaned the dishes in the sink
Syntactic Ambiguities I

- Prepositional phrases:
  They cooked the beans in the pot on the stove with handles.

- Particle vs. preposition:
  The puppy tore up the staircase.

- Complement structures
  The tourists objected to the guide that they couldn’t hear.
  She knows you like the back of her hand.

- Gerund vs. participial adjective
  Visiting relatives can be boring.
  Changing schedules frequently confused passengers.

Syntactic Ambiguities II

- Modifier scope within NPs
  impractical design requirements
  plastic cup holder

- Multiple gap constructions
  The chicken is ready to eat.
  The contractors are rich enough to sue.

- Coordination scope:
  Small rats and mice can squeeze into holes or cracks in the wall.

Dark Ambiguities

- Dark ambiguities: most analyses are shockingly bad
  (meaning, they don’t have an interpretation you can get your mind around)

  This analysis corresponds to the correct parse of
  “This will panic buyers!”

  Unknown words and new usages

  Solution: We need mechanisms to focus attention on the best ones, probabilistic techniques do this

PCFGs
Probabilistic Context-Free Grammars

- A context-free grammar is a tuple $< N, T, S, R >$
  - $N$: the set of non-terminals
    - Phrasal categories: S, NP, VP, ADJP, etc.
    - Parts-of-speech (pre-terminals): NN, JJ, DT, VB
  - $T$: the set of terminals (the words)
  - $S$: the start symbol
    - Often written as ROOT or TOP
    - Not usually the sentence non-terminal $S$
  - $R$: the set of rules
    - Of the form $X \rightarrow Y_1 Y_2 \ldots Y_k$, with $X, Y_i \in N$
    - Examples: $S \rightarrow NP VP$, $VP \rightarrow VP CC VP$
    - Also called rewrites, productions, or local trees

- A PCFG adds:
  - A top-down production probability per rule $P(Y_1 Y_2 \ldots Y_k | X)$

Treebank Sentences

- $(S \ (NP\text{-SBJ} \ The \ move) \ (VP \ followed \ (NP \ (NP \ other \ lenders))) \ (PP \ against \ (NP \ Arizona \ real \ estate \ loans))))$

Treebank Grammars

- Need a PCFG for broad coverage parsing.
- Can take a grammar right off the trees (doesn’t work well):

- Better results by enriching the grammar (e.g., lexicalization).
- Can also get state-of-the-art parsers without lexicalization.

Treebank Grammar Scale

- Treebank grammars can be enormous
  - As FSAs, the raw grammar has ~10K states, excluding the lexicon
  - Better parsers usually make the grammars larger, not smaller
Chomsky Normal Form

- **Chomsky normal form:**
  - All rules of the form $X \rightarrow YZ$ or $X \rightarrow w$
  - In principle, this is no limitation on the space of (P)CFGs
    - N-ary rules introduce new non-terminals
  - Unaries / empties are "promoted"
  - In practice it’s kind of a pain:
    - Reconstructing n-aries is easy
    - Reconstructing unaries is trickier
    - The straightforward transformations don’t preserve tree scores
  - Makes parsing algorithms simpler!

CKY Parsing

A Recursive Parser

```plaintext
bestScore(X,i,j)
if (j = i+1)
  return tagScore(X,s[i])
else
  return max score(X→YZ) *
    bestScore(Y,i,k) *
    bestScore(Z,k,j)
```

- Will this parser work?
- Why or why not?
- Memory requirements?

A Memoized Parser

```plaintext
bestScore(X,i,j)
if (scores[X][i][j] == null)
  if (j = i+1)
    score = tagScore(X,s[i])
  else
    score = max score(X→YZ) *
      bestScore(Y,i,k) *
      bestScore(Z,k,j)
  scores[X][i][j] = score
return scores[X][i][j]
```

- One small change:
A Bottom-Up Parser (CKY)

- Can also organize things bottom-up

```plaintext
bestScore(s)
  for (i : [0, n-1])
    for (X : tags[s[i]])
      score[X][i][i+1] =
        tagScore(X, s[i])
  for (diff : [2, n])
    for (i : [0, n-diff])
      j = i + diff
      for (X : rule)
        for (k : [i+1, j-1])
          score[X][i][j] =
            max(score[X][i][j],
                score(X->YZ) *
                score[Y][i][k] *
                score[Z][k][j])
```

 Unary Rules

- Unary rules?

```plaintext
bestScore(X, i, j, s)
  if (j = i+1)
    return tagScore(X, s[i])
  else
    return max(score(X->YZ) *
               bestScore(Y, i, k) *
               bestScore(Z, k, j)
               score(X->Y) *
               bestScore(Y, i, j))
```

 CNF + Unary Closure

- We need unaries to be non-cyclic
  - Can address by pre-calculating the unary closure
  - Rather than having zero or more unaries, always have exactly one

```plaintext
bestScoreU(X, i, j, s)
  if (j = i+1)
    return tagScore(X, s[i])
  else
    return max(score(X->YZ) *
               bestScoreU(Y, i, k) *
               bestScoreU(Z, k, j)
               score(X->Y) *
               bestScoreU(Y, i, j))
```

 Alternating Layers

```plaintext
bestScoreB(X, i, j, s)
  return max(score(X->YZ) *
            bestScoreB(Y, i, k) *
            bestScoreB(Z, k, j))
```

- Alternate unary and binary layers
- Reconstruct unary chains afterwards
Learning PCFGs

Treebank PCFGs

- Use PCFGs for broad coverage parsing
- Can take a grammar right off the trees (doesn’t work well):

```
ROOT
  |--- S
    |--- NP VP.
      |--- PRP
      |--- VBD ADJP
```

Model | F1
------|-----
Baseline | 72.0

Conditional Independence?

- Not every NP expansion can fill every NP slot
- A grammar with symbols like “NP” won’t be context-free
- Statistically, conditional independence too strong

Non-Independence

- Independence assumptions are often too strong.
- Example: the expansion of an NP is highly dependent on the parent of the NP (i.e., subjects vs. objects).
- Also: the subject and object expansions are correlated!
Grammar Refinement

- Example: PP attachment

```
They raised a point of order
```

Grammar Refinement

```
S
  NP-She
  VP
    PRP VBD NP-noise
      She heard DT NN
      the noise
```

- Structure Annotation [Johnson ’98, Klein&Manning ’03]
- Lexicalization [Collins ’99, Charniak ’00]
- Latent Variables [Matsuzaki et al. 05, Petrov et al. ’06]

Structural Annotation

The Game of Designing a Grammar

```
S
  NP
    PRP VBD NP-VP
      She heard DT NN
      the noise
```

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Structural annotation
Lexicalization

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Structural annotation [Johnson '98, Klein and Manning 03]
  - Head lexicalization [Collins '99, Charniak '00]

Problems with PCFGs

- If we do no annotation, these trees differ only in one rule:
  - VP → VP PP
  - NP → NP PP
- Parse will go one way or the other, regardless of words
- We addressed this in one way with unlexicalized grammars (how?)
- Lexicalization allows us to be sensitive to specific words

What’s different between basic PCFG scores here?
What (lexical) correlations need to be scored?
Lexicalized Trees

- Add “head words” to each phrasal node
  - Syntactic vs. semantic heads
  - Headship not in (most) treebanks
  - Usually use head rules, e.g.:
    - NP:
      - Take leftmost NP
      - Take rightmost N
      - Take rightmost JJ
      - Take right child
    - VP:
      - Take leftmost VB
      - Take leftmost VP
      - Take left child

Lexical Derivation Steps

- A derivation of a local tree [Collins 99]
  - Choose a head tag and word
  - Choose a complement bag
  - Generate children (incl. adjuncts)
  - Recursively derive children

Lexicalized PCFGs?

- Problem: we now have to estimate probabilities like
  \[ VP(saw) \rightarrow \text{VB}(\text{saw}) \ \text{NP}(\text{her}) \ \text{NP}(\text{today}) \]
- Never going to get these atomically off of a treebank
- Solution: break up derivation into smaller steps

Lexicalized CKY

- bestScore(X, i, j, h)
  - if \((j = i + 1)\)
    - return tagScore(X, s[i])
  - else
    - return max(max_score(X[h] -> Y[h] Z[h']) * bestScore(Y, i, k, h) * bestScore(Z, k, j, h'),
      max_score(X[h] -> Y[h'] Z[h]) * bestScore(Y, i, k, h') * bestScore(Z, k, j, h)
Results

- **Some results**
  - Collins 99 – 88.6 F1 (generative lexical)
  - Charniak and Johnson 05 – 89.7 / 91.3 F1 (generative lexical / reranked)
  - Petrov et al 06 – 90.7 F1 (generative unlexical)
  - McClosky et al 06 – 92.1 F1 (gen + rerank + self-train)

- **However**
  - Bilexical counts rarely make a difference (why?)
  - Gildea 01 – Removing bilexical counts costs < 0.5 F1

Latent Variable PCFGs

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Parent annotation [Johnson ’98]
  - Head lexicalization [Collins ’99, Charniak ’00]
  - Automatic clustering?
Learned Splits

- **Proper Nouns (NNP):**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NNP-12</td>
<td>John</td>
<td>Robert</td>
<td>James</td>
</tr>
<tr>
<td>NNP-2</td>
<td>J.</td>
<td>E.</td>
<td>L.</td>
</tr>
<tr>
<td>NNP-1</td>
<td>Bush</td>
<td>Noriega</td>
<td>Peters</td>
</tr>
<tr>
<td>NNP-15</td>
<td>New</td>
<td>San</td>
<td>Wall</td>
</tr>
<tr>
<td>NNP-3</td>
<td>York</td>
<td>Francisco</td>
<td>Street</td>
</tr>
</tbody>
</table>

- **Personal pronouns (PRP):**

<table>
<thead>
<tr>
<th>PRP-0</th>
<th>It</th>
<th>He</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRP-1</td>
<td>it</td>
<td>he</td>
<td>they</td>
</tr>
<tr>
<td>PRP-2</td>
<td>it</td>
<td>them</td>
<td>him</td>
</tr>
</tbody>
</table>

- **Relative adverbs (RBR):**

<table>
<thead>
<tr>
<th>RBR-0</th>
<th>further</th>
<th>lower</th>
<th>higher</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBR-1</td>
<td>more</td>
<td>less</td>
<td>More</td>
</tr>
<tr>
<td>RBR-2</td>
<td>earlier</td>
<td>Earlier</td>
<td>later</td>
</tr>
</tbody>
</table>

- **Cardinal Numbers (CD):**

<table>
<thead>
<tr>
<th>CD-7</th>
<th>one</th>
<th>two</th>
<th>Three</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD-4</td>
<td>1989</td>
<td>1990</td>
<td>1988</td>
</tr>
<tr>
<td>CD-11</td>
<td>million</td>
<td>billion</td>
<td>trillion</td>
</tr>
<tr>
<td>CD-0</td>
<td>1</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>CD-3</td>
<td>1</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>CD-9</td>
<td>78</td>
<td>58</td>
<td>34</td>
</tr>
</tbody>
</table>