The Basics of Parsing

L715: Seminar on: Data manipulation for parser improvement

Dept. of Linguistics, Indiana University
Fall 2011

Where we're going

- Basic PCFG representation & parsing:
  - One approach: lexicalized parsing
  - Other approach (this semester): alter tree representations
  - Available trainable parsers
- Basic DG representation & parsing:
  - Basic approach #1: transition-based
  - Basic approach #2: graph-based
  - Available trainable parsers
- Available treebanks

PCFG representation & parsing

Manning and Schütze (2000, ch. 10–11)

A probabilistic context-free grammar (PCFG) captures syntactic regularities in languages probabilistically

Formally, consist of:

- Set of terminals (words): $\{w_1, ..., w_V\}$
- Set of nonterminals (categories): $\{N_1, ..., N_n\}$
- Designated start symbol: N_1 (often S)
- Set of rules/productions: $\{N_i \rightarrow \zeta_j\}$ (ζ_j is a sequence of terminals and nonterminals)
- A set of probabilities for each rule, s.t.:

(1) $\forall i \sum_j P(N_i \rightarrow \zeta_j) = 1$

  - These probabilities are the probability of a sequence of daughters given a particular mother, i.e., $P(N_i \rightarrow \zeta_j) = P(N_i \rightarrow \zeta_j | N_i)$

Probability of a tree

- To get $P(t)$, we simply multiply the probabilities of all the subtrees
- The probability of a sentence (w_{1m}), then, is the sum of all parses for that sentence

(2) $P(w_{1m}) = \sum_t P(t)$, where t yields w_{1m}

Assumptions of PCFGs

There are certain assumptions for a PCFG model

Main assumption: rules are independent of one another

- Place invariance: Probability of subtree doesn't depend on where it is in the string

(3) $\forall k$: (k = time point) $P(N_j \rightarrow \zeta)$ is the same

- Context-free: Probability of subtree doesn't depend on words outside of the subtree

(4) $P(N_j \rightarrow \zeta | \text{anything outside its span}) = P(N_j = \zeta)$

- Ancestor-free: Probability of subtree doesn't depend on nodes outside of the subtree

(5) $P(N_j \rightarrow \zeta | \text{any ancestor nodes}) = P(N_j = \zeta)$

Probabilistic grammar

S → NP VP : 0.8	S → NP VP PP : 0.2	
NP → DET N : 0.5	NP → NP PP : 0.5	
VP → V NP : 1.0		
PP → P NP : 1.0		
N → girl : 0.25	N → boy : 0.25	
N → park : 0.25	N → telescope : 0.25	
V → saw : 1.0		
P → with : 0.5	P → in : 0.5	DET → the : 1.0

This allows us, e.g., to sort out the different analyses for *The boy saw the girl in the park with the telescope*

Limitations of PCFGs

PCFGs cannot do everything in and of themselves:

- PCFGs do not take lexical information into account, making parse plausibility less than ideal and making PCFGs worse than *n*-grams as a language model.
- PCFGs have certain biases; i.e., the probability of a smaller tree is greater than a larger tree.
- When two different analyses use the same set of rules, they have the same probability, regardless of the order used.

Lexicalized parsing

Collins (1997)

To overcome the limitation of lacking lexical information, parsers started including it ...

- The models Collins (1997) are:
  - Lexicalized
  - Dependency-based

Dependencies

How to incorporate dependency information from a treebank which doesn't explicitly have dependencies ...

(6) John Smith, the president of IBM, announced his resignation yesterday.

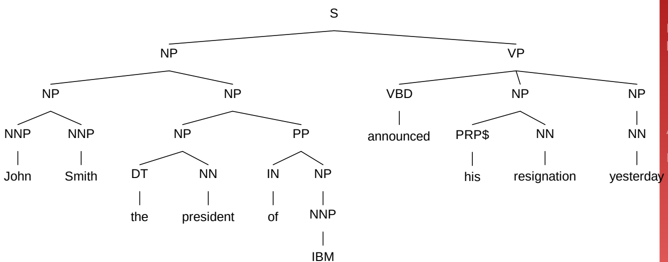
Here are some of the dependencies, along with their representation ... (base NPs are treated as a unit)

- *Dependent* → *Head* ... Hcat_Mother_DCat
- [John Smith] → announced ... VP_S_NP
- [the president] → [John Smith] ... NP_NP_NP
- [his resignation] → announced ... VBD_VP_NP
- yesterday → announced ... VBD_VP_NP

Dependency probabilities

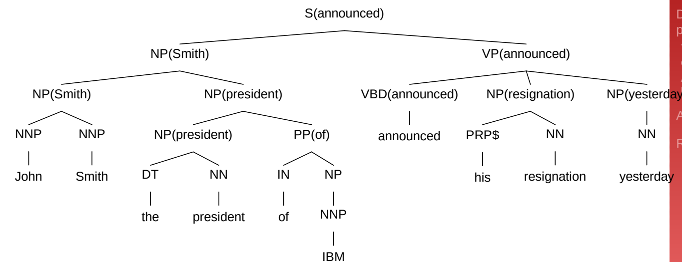
- Each dependency has a particular probability, e.g., $P(VBD\_VP\_NP)$
- The probability of a rule is made from the probabilities of the components
  - P(VP → VBD NP NP) composed of $P(VBD\_VP\_NP)$ for first NP and $P(VBD\_VP\_NP)$ for second NP

Motivating Lexicalization



yesterday attaches to *announced*

Propogating Head words



Lexicalized rules

- S(announced) → NP(Smith) VP(announced)
- NP(Smith) → NP(Smith) NP(president)
- NP(Smith) → NNP(John) NNP(Smith)
- NP(president) → NP(president) PP(of)
- PP(of) → IN(of) NP(IBM)
- ...

Methods were developed to incorporate head word information without succumbing to data sparsity

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

Generative models

Collins (1997) differs from Collins (1996) in that it is a *generative* model, not a conditional model

- Generative model: maximize the sentence-tree pairing $P(S, T)$
- Conditional (or parsing) model: maximize the tree given the sentence: $P(T|S)$

Maximizing one is equivalent to maximizing the other

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

Model 1

Need a way to get reasonable probability estimates for $P(RHS|LHS)$

Too many possible rules after adding lexical information

- Note: the parent *LHS* of a headword *h* is composed of the headword's label *H* plus left & right sisters

  (7) $LHS \rightarrow L_n ... L_1\ H\ R_1 ... R_m$

- Break down the probability of *LHS* into generating the left sisters, the head category, the right sisters

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

Model 2

For the second model, Collins:

- distinguishes adjuncts from complements
  - Mark complement categories with a C (e.g, NP-C) whether a subject or an object
  - Identify them in the training data (Penn Treebank) using rules similar to head-finding rules
- adds subcategorization information

The two tasks are related, because to know that something is a complement we have to know that it was selected for

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

Motivating subcategorization

Model 1 assumed that each child of a parent node was independently generated

This leads to some bad choices, even with complements marked:

- $[_{NP-C}$ Dreyfus] $[_{NP-C}$ the best fund] $[_{VP}$ was low]
- $[_{NP-C}$ $[_{NP}$ Dreyfus] $[_{NP}$ the best fund]] $[_{VP}$ was low]

$P(NP\text{-}C(Dreyfus)|S, VP, was) * P(NP\text{-}C(fund)|S, VP, was)$ is "unreasonably high", giving the wrong parse

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

Adding subcategorization information

1. Choose head constituent *H* with probability $P(H|LHS, h)$
2. Choose left and right subcat frames *LC* and *RC* with probabilities:
   - $P(LC|LHS, H, h)$
   - $P(RC|LHS, H, h)$
3. Generate left and right modifiers with probabilities:
   - $P(L_i|LHS, H, h, LC)$
   - $P(R_i|LHS, H, h, RC)$
   - Distance measure also used, but we ignore it here

The important thing to note is that subcategorization information is added to the surrounding, conditioning context

Other approach: Unlexicalized parsing

Another trend that arose is to not include lexical properties, but rather: better non-lexical properties

- e.g., NP is less informative than NP-SBJ vs. NP-OBJ

In other words: the treebank representation *matters greatly*

- We will spend the rest of the semester exploring this in more detail

Available Constituency Parsers

- LoPar: http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/LoPar.html
  - Trainable; models for English & German
- BitPar: http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html
  - Trainable; models for English & German
- Charniak & Johnson parser: http://github.com/BLLIP/bllip-parser
  - Trainable; mainly used for English

Available Constituency Parsers (2)

- Collins/Bikel parser: http://people.csail.mit.edu/mcollins/code.html http://www.cis.upenn.edu/~dbikel/software.html
  - Trainable on English, Chinese, and Arabic; designed for Penn Treebank-style annotation
- Stanford parser: http://nlp.stanford.edu/downloads/lex-parser.shtml
  - Trainable; models for English, German, Chinese, & Arabic; dependencies also available
- Berkeley parser: http://code.google.com/p/berkeleyparser/
  - Trainable; models for English, German, and Chinese

Dependency parsing

Two main kinds of dependency parsing in use now:

- Transition-based dependency parsing
- Graph-based dependency parsing

Transition-based dependency parsing

MaltParser is a form of transition-based dependency parsing

Let's look at: http://jones.ling.indiana.edu/~mdickinson/nasslli10/02/02-transition.pdf

Graph-based dependency parsing

MSTParser is a form of graph-based dependency parsing

Let's look at: http://jones.ling.indiana.edu/~mdickinson/nasslli10/04/04-graph.pdf

Trainable Dependency Parsers

- Jason Eisner's **probabilistic dependency parser**
  - Based on bilexical grammar
  - Contact Jason Eisner: jason@cs.jhu.edu
  - Written in LISP
- Ryan McDonald's **MSTParser**
  - Based on the algorithms of McDonald et al. (2005a,b)
  - URL: http://www.seas.upenn.edu/~ryantm/software/MSTParser/
  - Written in JAVA

Trainable Dependency Parsers (2)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Joakim Nivre's **MaltParser**
  - Inductive dependency parser with memory-based learning and SVMs
  - URL: http://w3.msi.vxu.se/~nivre/research/MaltParser.html
  - Executable versions are available for Solaris, Linux, Windows, and MacOS (open source version planned for fall 2006)

Dependency Parsers for Specific Languages

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Dekang Lin's **Minipar**
  - Principle-based parser
  - Grammar for English
  - URL: http://www.cs.ualberta.ca/~lindek/minipar.htm
  - Executable versions for Linux, Solaris, and Windows
- Wolfgang Menzel's **CDG Parser**:
  - Weighted constraint dependency parser
  - Grammar for German, (English under construction)
  - Online demo: http://nats-www.informatik.uni-hamburg.de/Papa/ParserDemo
  - Download: http://nats-www.informatik.uni-hamburg.de/download

Dependency Parsers for Specific Languages (2)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Taku Kudo's **CaboCha**
  - Based on algorithms of Kudo and Matsumoto (2002), uses SVMs
  - URL: http://www.chasen.org/~taku/software/cabocha/
  - Web page in Japanese
- Gerold Schneider's **Pro3Gres**
  - Probability-based dependency parser
  - Grammar for English
  - URL: http://www.ifi.unizh.ch/CL/gschneid/parser/
  - Written in PROLOG
- Daniel Sleator's & Davy Temperley's **Link Grammar Parser**
  - Undirected links between words
  - Grammar for English
  - URL: http://www.link.cs.cmu.edu/link/

Constituent Treebanks

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Penn Treebank
  - ca. 1 million words
  - Available from LDC, license fee
  - URL: http://www.cis.upenn.edu/~treebank/home.html
  - Dependency conversion rules, available from e.g. Collins (1999)
  - For conversion with arc labels: Penn2Malt: http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html
- BulTreebank
  - ca. 14 000 sentences
  - URL: http://www.bultreebank.org/
  - Dependency version available from Kiril Simov (kivs@bultreebank.org)

Constituent Treebanks (2)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Penn Chinese Treebank
  - ca. 4 000 sentences
  - Available from LDC, license fee
  - URL: http://www.cis.upenn.edu/~chinese/ctb.html
  - For conversion with arc labels: Penn2Malt: http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html
- Sinica Treebank
  - ca. 61 000 sentences
  - Available Academia Sinica, license fee
  - URL: http://godel.iis.sinica.edu.tw/CKIP/engversion/treebank.htm
  - Dependency version available from Academia Sinica

Constituent Treebanks (3)

- Alpino Treebank for Dutch
  - ca. 150 000 words
  - Freely downloadable
  - URL: http://www.let.rug.nl/vannoord/trees/
  - Dependency version downloadable at http://nextens.uvt.nl/~conll/free_data.html
- TIGER/NEGRA
  - ca. 50 000/20 000 sentences
  - Freely available, license agreement
  - TIGER URL: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/ NEGRA URL: http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/
  - Dependency version of TIGER is included in release

Constituent Treebanks (4)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- TüBa-D/Z
  - ca. 22 000 sentences
  - Freely available, license agreement
  - URL: http://www.sfs.uni-tuebingen.de/en_tuebadz.shtml
  - Dependency version available from SfS Tübingen
- TüBa-J/S
  - Dialog data
  - ca. 18 000 sentences
  - Freely available, license agreement
  - Dependency version available from SfS Tübingen
  - URL: http://www.sfs.uni-tuebingen.de/en_tuebajs.shtml (under construction)

Constituent Treebanks (5)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Cast3LB
  - ca. 18 000 sentences
  - URL: http://www.dlsi.ua.es/projectes/3lb/index_en.html
  - Dependency version available from Toni Martí (amarti@ub.edu)
- Talbanken05
  - ca. 300 000 words
  - Freely downloadable
  - URL: http://w3.msi.vxu.se/~nivre/research/Talbanken05.html
  - Dependency version also available

Dependency Treebanks

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Prague Arabic Dependency Treebank
  - ca. 100 000 words
  - Available from LDC, license fee (CoNLL-X shared task data, catalogue number LDC2006E01)
  - URL: http://ufal.mff.cuni.cz/padt/
- Prague Dependency Treebank
  - 1.5 million words
  - 3 layers of annotation: morphological, syntactical, tectogrammatical
  - Available from LDC, license fee (CoNLL-X shared task data, catalogue number LDC2006E02)
  - URL: http://ufal.mff.cuni.cz/pdt2.0/

Dependency Treebanks (2)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Danish Dependency Treebank
  - ca. 5 500 trees
  - Annotation based on Discontinuous Grammar Kromann (2005)
  - Freely downloadable
  - URL: http://www.id.cbs.dk/~mtk/treebank/
- Bosque, Floresta sintá(c)tica
  - ca. 10 000 trees
  - Freely downloadable
  - URL: http://acdc.linguateca.pt/treebank/info_floresta_English.html

Dependency Treebanks (3)

The Basics of Parsing
PCFG parsing
Lexicalized parsing
Unlexicalized parsing
Available Constituency Parsers
Dependency parsing
Transition-based parsing
Graph-based parsing
Available Dependency Parsers
Available treebanks
References

- Slovene Dependency Treebank
  - ca. 30 000 words
  - Freely downloadable
  - URL: http://nl.ijs.si/sdt/
- METU-Sabanci Turkish Treebank
  - ca. 7 000 trees
  - Freely available, license agreement
  - URL: http://www.ii.metu.edu.tr/~corpus/treebank.html

References

Collins, Michael (1997). Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 16–23.

Collins, Michael (1999). Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.

Kromann, Matthias Trautner (2005). *Discontinuous Grammar: A Dependency-Based Model of Human Parsing and Language Learning*. Doctoral Dissertation, Copenhagen Business School.

Kudo, Taku and Yuji Matsumoto (2002). Japanese Dependency Analysis Using Cascaded Chunking. In *Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL)*. pp. 63–69.

Manning, C. D. and H. Schütze (2000). *Foundations of Statistical Natural Language Processing*. MIT Press.

McDonald, Ryan, Koby Crammer and Fernando Pereira (2005a). Online Large-Margin Training of Dependency Parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 91–98.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov and Jan Hajič (2005b). Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. pp. 523–530.