# Dependency Parsing

Ganesh Bhosale - 09305034
Neelamadhav G. - 09305045
Nilesh Bhosale - 09305070
Pranav Jawale - 09307606

under the guidance of

Prof. Pushpak Bhattacharyya

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay
Powai, Mumbai - 400 076

April 22, 2011

# Content

# Outline

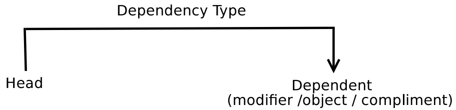
# Motivation

- Inspired by dependency grammar
- "Who-did-what-to-whom"
- Easy to encode long distance dependencies
- Better suited for free word order languages
- Proven to be useful in various NLP & ML applications

# Outline

# Dependency Grammar

- **Basic Assumption:** Syntactic structure essentially consists of lexical items linked by binary asymmetrical relations called dependencies.[1]



# Constituency parsing example

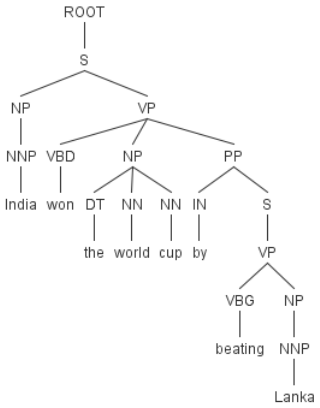



Figure: Output of Stanford parser [Penn treebank type]

**India won the world cup by beating Lanka**

# Example of dependency parser output

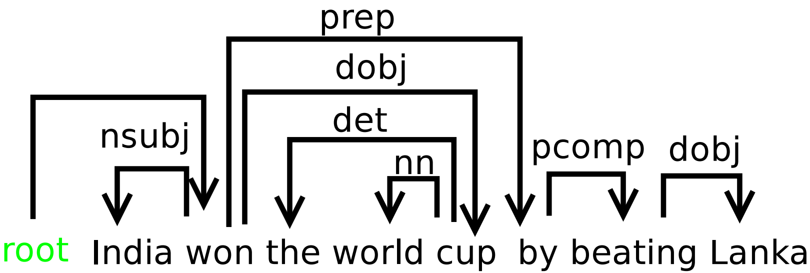



Figure: Output of Stanford dependency parser

# Example of dependency grammar


Figure: Parse tree

# Example of dependency grammar

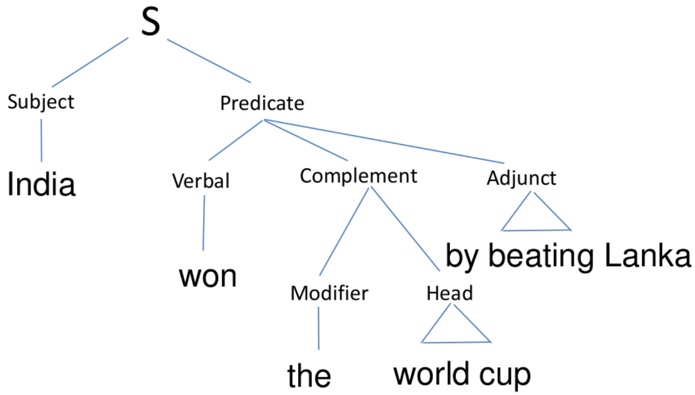
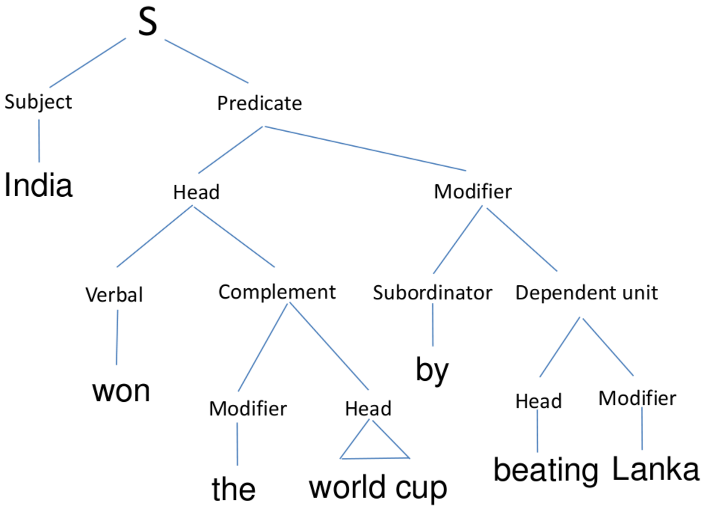



Figure: Parse tree

# Example of dependency grammar



# List of dependency relations

- aux - auxiliary
- arg - argument
- cc - coordination
- conj - conjunct
- expl - expletive (expletive /there)
- mod - modifier
- punct - punctuation
- ref - referent

Marie-Catherine de Marneffe and Christopher D. Manning. September 2008. "Stanford typed dependencies manual"[2]

# Dependency Parsing

- Input: Sentence
  $x = w_0, w_1, ..., w_n$
- Output: Dependency graph $G = (V, A)$ for $x$ where:
  - $V = 0, 1, ..., n$ is the vertex set,
  - $A$ is the arc set, i.e., $(i, j, k) \in A$ represents a dependency from $w_i$ to $w_j$ with label $l_k \in L$

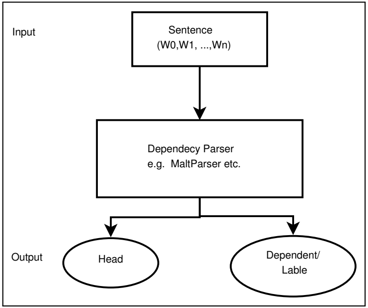



Figure: Dependency Parsing

# Outline

# Basic Requirements

- Unity: Single tree with a unique root consisting of all the words in the input sentence.
- Uniqueness: Each word has only one Head (Some parsing techniques (ex. Hudson) works with multiple heads)

# Types of Dependency Parsing

- Rule Based (Grammar Driven)
  - Fundamental algorithm of dependency parsing (by Michael A. Covington)[3]
- Machine Learning Based (Data Driven)
  - Statistical Dependency Analysis with SVM (by Hiroyasu et.al)[4]

Joakim Nivre, (2005) "Dependency Grammar and Dependency Parsing,". Vaxjo University

# Strategy-1 (Brute-force search)

- Examine each pair of words in the entire sentence whether they can be *head-to-dependent* or *dependent-to-head* based on the grammar
  - For $n$ words $n(n-1)$ pairs, so the complexity is $O(n^2)$

Michael A. Covington. 2001. "A Fundamental Algorithm for Dependency Parsing", 39th Annual ACM Southeast Conference

# Strategy-2 (Exhaustive left-to-right search)

- Take words one by one starting at the beginning of the sentence, and try linking each word as head or dependent of every previous word.
  - Whether to check from 1 to $n-1$ or from $n-1$ to 1: Most of the times head and dependents are more like to be near the target word.
  - Whether it is better to look for heads and then dependents or dependents then heads: Cannot yet be determined
  - Whether the algorithm enforce unity and uniqueness

Michael A. Covington. 2001. “A Fundamental Algorithm for Dependency Parsing”, 39th Annual ACM Southeast Conference

# Strategy-3 (Enforcing Uniqueness)

- When a word has a head it cannot have another one.
  - When looking for dependent of the current word: do not consider words that are already dependents of something else.
  - When looking for the head of the current word: stop after finding one head.

Michael A. Covington. 2001. “A Fundamental Algorithm for Dependency Parsing”, 39th Annual ACM Southeast Conference

# Strategy-3 (Enforcing Uniqueness)

```
Given an n-word sentence:
[1] for i := 1 to n do
[2] begin
[3]   for j := i - 1 down to 1 do
[4]   begin
[5]       If no word has been
                linked as head of word i, then
[6]          if the grammar permits,
                link word j as head of word i;
[7]       If word j is not a dependent
                of some other word, then
[8]          if the grammar permits,
               link word j as dependent of word i
[9]   end
[10] end
```

# Dependency parsing with SVM

**Support Vector Machines**

- Binary classifier based on maximum margin strategy.
- In svm's we find a hyperplane $w.x + b = 0$ which correctly separates training examples and has maximum margin which is the distance between hyper planes: $w.x + b >= +1$ and $w.x + b \leq -1$

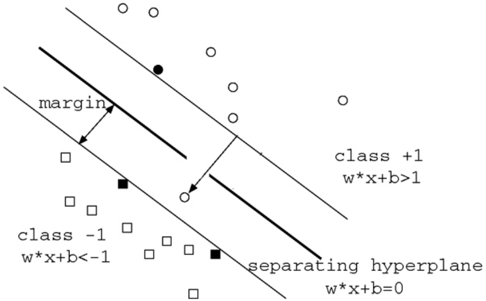



Figure: Support Vector Machine

# Dependency parsing with SVM(cont..)

- This is a multiclass classification problem.
- There are three classes (parsing actions)
  1. Shift
  2. Right
  3. Left
- We construct three binary classifiers corresponding to each action
  1. Left Vs. Shift
  2. Left Vs. Right
  3. Right Vs. Shift

Hiroyasu Yamada, Yuji Matsumoto, 2003. "STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,", Graduate School of Information Science,.

# Dependency parsing with SVM(cont..)

- Let us take two neighboring words (Target words)
- **Shift**: No dependencies between the target nodes and the point of focus simply moves to right

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| I | saw | a | girl | with | → | I | saw | a | girl | with |
| PRP | VBD | DT | NN | IN | | PRP | VBD | DT | NN | IN |

Figure: An example of the action shift

Hiroyasu Yamada, Yuji Matsumoto, 2003. "STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,", Graduate School of Information Science,.

# Dependency parsing with SVM(cont..)

- **Right**: Left node of target nodes becomes a child of right one

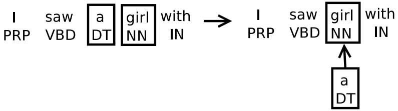



Figure: An example of the action right

Hiroyasu Yamada, Yuji Matsumoto, 2003. "STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,", Graduate School of Information Science,.

# Dependency parsing with SVM(cont..)

- **Left**: Right node of target nodes becomes a child of left one

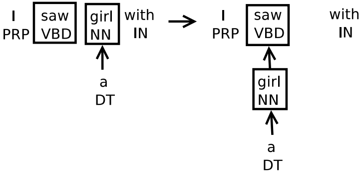



Figure: An example of the action left

Hiroyasu Yamada, Yuji Matsumoto, 2003. "STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,", Graduate School of Information Science,.

# Dependency parsing with SVM(cont..)

**Learning dependency structure**

- POS tag and words are used as the candidate features for the nodes within left and right context.

Table: Summary of the feature types and their values

| Type | Value |
|---|---|
| pos | part of speech(POS) tag string |
| lex | word string |
| ch-L-pos | POS tag string of the child node modifying to the parent node from left side |
| ch-L-lex | word string of the child node node modifying to the parent node from left side |
| ch-R-pos | POS tag string of the child node modifying to the parent node from right side |
| ch-R-lex | word string of the child node modifying to the parent node from right side |

Hiroyasu Yamada, Yuji Matsumoto, 2003. "STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,", Graduate School of Information Science,.

# Dependency parsing with SVM(cont..)

**Parsing Algorithm**

- Estimate appropriate parsing actions using contextual information surrounding the target node.
- The parser constructs a dependency tree by executing the estimated action.

Hiroyasu Yamada, Yuji Matsumoto, 2003. “STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,”, Graduate School of Information Science,.

# Outline

# Dependency Parser Evaluation

1. Evaluation Methodology
   1. Sentence based : Quantitative
   2. Token based : Qualitative
2. Input data tracks
   1. Multilingual
   2. Domain Adaptation

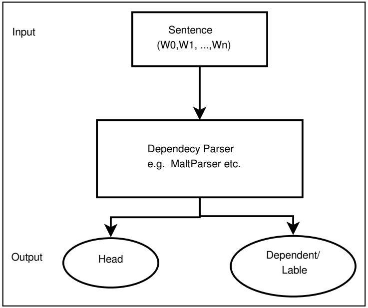



Figure: Dependency Parsing

# Evaluation Methodology

- Quantitative Evaluation
  - It takes into account the number of sentence which were parsed correctly[6].
- Metrics
  1. Correct Sentences
     i.e. Sentences with correct labeled graph
  2. Incorrect Sentences
     i.e. Sentences with not correct labeled graph
  3. Sentence Length
- Critique
  - Does not analyze linguistic errors

# Evaluation Methodology(cont...)

- Qualitative Evaluation
  - It takes into account type of mistakes performed by the parser.
- Metrics [5]
  1. Labeled attachment score (LAS):
     i.e. Tokens with correct head and label
  2. Unlabeled attachment score (UAS):
     i.e. Tokens with correct head
  3. Label accuracy (LA):
     i.e. Tokens with correct label

Sabine Buchholz and Erwin Marsi. 2006. "CoNLLX shared task on Multilingual Dependency Parsing.",, Proceedings of the 10th Conference on Computational Natural Language Learning New York City. Association for Computational Linguistic.

# Quantitative Analysis

- Experiment conducted with 46 sentences on dependency parser (e.g. Standford, Malt, Rasp etc) and manually checked correctness of head and label output of parser.[6]

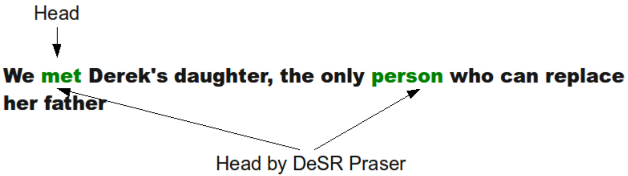
Table: Parser Evaluation

| Parser | Correct Sentences | Incorrect Sentences |
|---|---|---|
| Stanford | 67% | 33% |
| DeSR | 54% | 46% |
| RASP | 45% | 55% |
| MINIPAR | 56% | 44% |
| MALT | 63% | 37% |

Elisabet Comelles, Victoria Arranz, Irene Castellon 2010 "Constituency and Dependency Parsers Evaluation", Sociedad Espanola para el Procesamiento del Lenguaje Natural.

# Qualitative Analysis

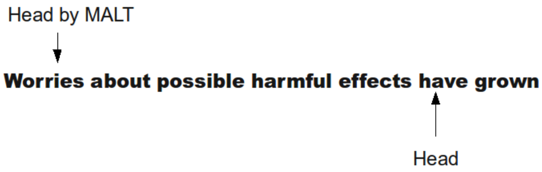
- Identification of more than one Head[6]




Elisabet Comelles, Victoria Arranz, Irene Castellon 2010 "Constituency and Dependency Parsers Evaluation", Sociedad Espanola para el Procesamiento del Lenguaje Natural.

# Qualitative Analysis (cont..)

- Wrong Identification of Head[6]




Elisabet Comelles, Victoria Arranz, Irene Castellon 2010 "Constituency and Dependency Parsers Evaluation", Sociedad Espanola para el Procesamiento del Lenguaje Natural.

# Qualitative Analysis (cont..)

- Wrong Dependencies[6]


Elisabet Comelles, Victoria Arranz, Irene Castellon 2010 “Constituency and Dependency Parsers Evaluation”, Sociedad Espanola para el Procesamiento del Lenguaje Natural.

# Input data tracks

- Multilingual Tracks[5]
  - Consist of different languages
  - Annotated training data
  - Test data
- Domain Adaptation data
  - The goal is to adopt annotated resources from sources domain to a target domain of interest.
  - Source training data: Wall Street Journal
  - Target data:
    - Biomedical abstract
    - Chemical abstract

Sabine Buchholz and Erwin Marsi. 2006. "CoNLLX shared task on Multilingual Dependency Parsing.",, Proceedings of the 10th Conference on Computational Natural Language Learning New York City. Association for Computational Linguistic.

# Observations

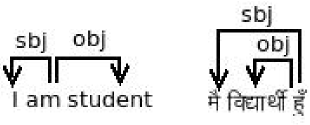
- Data Driven parsing System, it performs worse on data that does not come from the training domain[5]
- Parsing accuracy differed greatly between languages[5]
  LAS
  - Low (76.31 - 76.94%)
    Arabic,Greek
  - Medium (79.19 - 80.21%)
    Hungarian, Turkish
  - High (84.40 - 89.61%)
    Chines, English

Sabine Buchholz and Erwin Marsi. 2006. “CoNLLX shared task on Multilingual Dependency Parsing.”,, Proceedings of the 10th Conference on Computational Natural Language Learning New York City. Association for Computational Linguistic.

# Outline

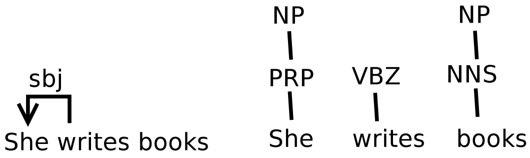
# Word Order

- Dependency structure independent of word order
- Suitable for free word order languages



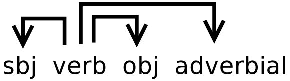
# Transparency

- Direct encoding of predicate-argument structure
- Fragments directly interpretable
- But only with labeled dependency graphs



# Expressivity

- Limited expressivity:
  - The Dependency tree contains one node per word and word at a time operation.
  - Impossible to distinguish between phrase modification and head modification in unlabeled dependency structure




V, VP or S modification?

# Outline

# Semantic Role Labeling:

- Identifying semantic arguments of the verb or predicate of a sentence.
- Classifying those semantic arguments to their specific semantic roles.
- Used in Question Answering System
- Example: MiniPar, Prague Treebank, etc.

# MINIPAR

- MINIPAR is a principle-based parser.
- It represents its grammar as a network where the nodes represent grammatical categories and the links represent types of (dependency) relationships.
- Output of MINIPAR is a dependency tree: It uses the heads of the phrases to decide the governor and dependent of a dependency relation.
- MINIPAR constructs all possible parses of an input sentence, then gives as output the one with the highest probability value.

# MINIPAR(cont..)

- Evaluation with the SUSANNE corpus[7]
- Achieves about 88% precision and 80% recall with respect to dependency relationships.
- It parses about 300 words per second.

Lin, D. 1998. A Dependency-based Method for Evaluating Broad-Coverage Parsers. Journal of Natural Language Engineering, p. 97â114.

# Text Mining in Biomedical domain

**Goal** Extracting information from statements concerning relations of biomedical entities, such as protein-protein interactions.

**Evaluation** 94% dependency coverage on GENIA corpus[8]

# Outline

# Conclusion

- A Dependency Parsing Approach is suitable for free word order languages.
- A Dependency Parsing Approach is applicable to many NLP and Machine Learning applications ex. Biomedical Text Mining, Semantic Role Labeling, etc.
- Data Driven parsing System, it performs worse on data that does not come from the training domain[5]

# Outline

# References


Joakim Nivre, (2005) *"Dependency Grammar and Dependency Parsing,"*. Vaxjo University

Marie-Catherine de Marneffe and Christopher D. Manning. September 2008. *"Stanford typed dependencies manual"*

Michael A. Covington. 2001. *"A Fundamental Algorithm for Dependency Parsing", 39th Annual ACM Southeast Conference*

Hiroyasu Yamada, Yuji Matsumoto, 2003. *"STATISTICAL DEPENDENCY ANALYSIS WITH SUPPORT VECTOR MACHINES,", Graduate School of Information Science,.*

Elisabet Comelles, Victoria Arranz, Irene Castellon 2010 *"Constituency and Dependency Parsers Evaluation"*, Sociedad Espanola para el Procesamiento del Lenguaje Natural.

Sabine Buchholz and Erwin Marsi. 2006. *"CoNLLX shared task on Multilingual Dependency Parsing.",*, Proceedings of the 10th Conference on Computational Natural Language Learning New York City. Association for Computational Linguistic.

Lin, D. 1998, *"A Dependency based Method for Evaluating Broad Coverage Parsers".* Journal of Natural Language Engineering,p. 97 to 114.

Pyysalo, S., Ginter, F., Pahikkala, T., Boberg, J., J Ìrvinen, J., and a Salakoski, T. 2006. *"Evaluation of two dependency parsers on biomedical corpus targeted at protein -protein interactions",*, International Journal of Medical Informatics, 75(6):430 to 442.

Pyysalo, S., Ginter, F., Laippala, V., Haverinen, K., Heimonen, J.,and Salakoski, T. 2007. *"On the unification of syntactic annotations under the stanford dependency scheme: A case study on BioInfer and GENIA",*, In ACL'07 workshop on Biological, translational, and clinical language processing (BioNLP'07), pages 25 to 32, Prague, Czech Republic. Association for Computational Linguistics