A Computational Model of Attachment Preferences

Frank Riccobono

LING 72500 - Sentence Processing
CUNY Graduate Center

April 10, 2014

Overview

The goal of this project is to create a computational model based on (M. F. Boston 2012) to account for cross-linguistic variations in attachment preferences using:

Overview

The goal of this project is to create a computational model based on (M. F. Boston 2012) to account for cross-linguistic variations in attachment preferences using:

- Dependency grammars

Overview

The goal of this project is to create a computational model based on (M. F. Boston 2012) to account for cross-linguistic variations in attachment preferences using:

- Dependency grammars
- Incremental, Transition-based dependency parser

Overview

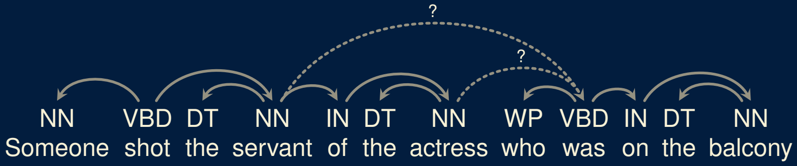
The goal of this project is to create a computational model based on (M. F. Boston 2012) to account for cross-linguistic variations in attachment preferences using:

- Dependency grammars
- Incremental, Transition-based dependency parser
- Features based on psycholinguistic theory

Overview

The goal of this project is to create a computational model based on (M. F. Boston 2012) to account for cross-linguistic variations in attachment preferences using:

- Dependency grammars
- Incremental, Transition-based dependency parser
- Features based on psycholinguistic theory

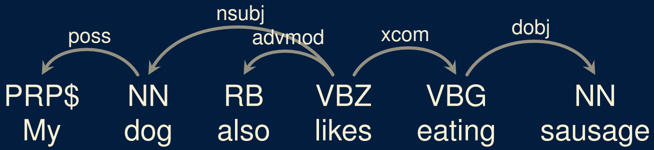

?
?
NN VBD DT NN IN DT NN WP VBD IN DT NN
Someone shot the servant of the actress who was on the balcony

Introduction to Dependency Grammar

Basic Principles

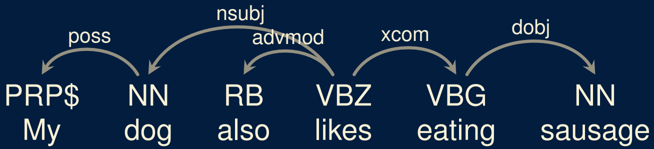
Introduction to Dependency Grammar

Basic Principles



Introduction to Dependency Grammar

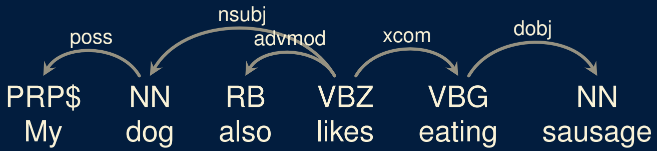
Basic Principles



Key notions (Tesnière 1959) and (Nivre 2006)

Introduction to Dependency Grammar

Basic Principles

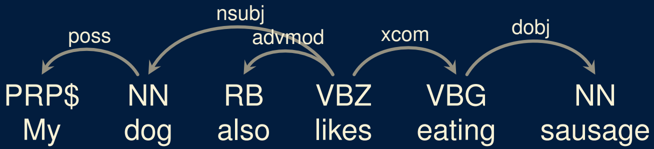


Key notions (Tesnière 1959) and (Nivre 2006)

- Constituent elements of a sentence are words [as opposed to phrases].

Introduction to Dependency Grammar

Basic Principles

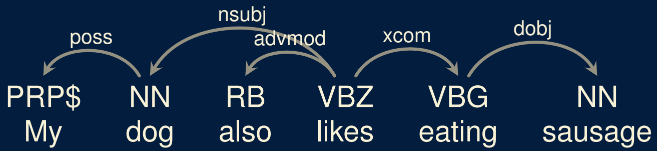


Key notions (Tesnière 1959) and (Nivre 2006)

- Constituent elements of a sentence are words [as opposed to phrases].
- The structure of the sentence is formed by the connections the mind perceives between the word and its neighbors.

Introduction to Dependency Grammar

Basic Principles



Key notions (Tesnière 1959) and (Nivre 2006)

- Constituent elements of a sentence are words [as opposed to phrases].
- The structure of the sentence is formed by the connections the mind perceives between the word and its neighbors.
- The structural connections establish dependency relations between the words.

Introduction to Dependency Grammar

Comparison to Phrase Structure Graphs

Introduction to Dependency Grammar

Comparison to Phrase Structure Graphs

- A word has *exactly one* head.

Introduction to Dependency Grammar

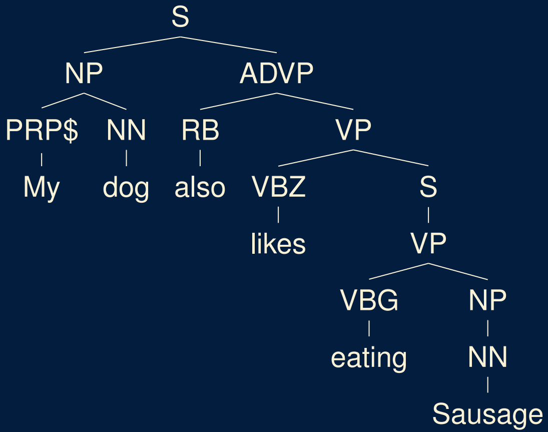
Comparison to Phrase Structure Graphs

- A word has *exactly one* head.
- Each head can have multiple dependents.

Introduction to Dependency Grammar

Comparison to Phrase Structure Graphs

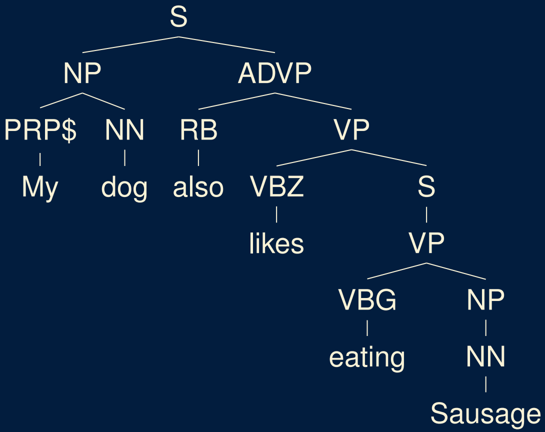
- A word has *exactly one* head.
- Each head can have multiple dependents.

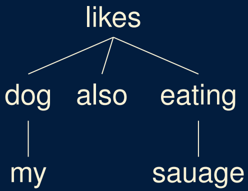


Introduction to Dependency Grammar

Comparison to Phrase Structure Graphs

- A word has *exactly one* head.
- Each head can have multiple dependents.





Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Components

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Components

- a queue, which represents the part of the sentence that has not yet been seen/heard

$$\textit{word}_7\ \textit{word}_8\ \textit{word}_9\ \ldots$$

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Components

- a queue, which represents the part of the sentence that has not yet been seen/heard

$$\mathit{word}_7\ \mathit{word}_8\ \mathit{word}_9 \ldots$$

- a stack, which represents working memory

$$\begin{array}{|c|} \mathit{word}_6 \\ \mathit{word}_5 \\ \mathit{word}_4 \\ \hline \end{array}$$

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Components

- a queue, which represents the part of the sentence that has not yet been seen/heard

$$word_7 \; word_8 \; word_9 \; \ldots$$

- a stack, which represents working memory

$$\begin{array}{|c|} word_6 \\ word_5 \\ word_4 \\ \hline \end{array}$$

- and a set of edges, which represents the dependency tree

$$\begin{gathered} \langle word_1, word_2 \rangle \\ \langle word_2, word_3 \rangle \\ \vdots \end{gathered}$$

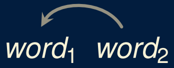
Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Transitions

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Transitions

- LEFT-ARC: Make the next word in the queue the head of the word at the top of the stack.

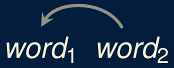


$word_1$ $word_2$

Introduction to Dependency Parsers

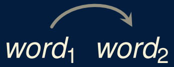
A Transition-Based System (Nivre 2008) - Transitions

- LEFT-ARC: Make the next word in the queue the head of the word at the top of the stack.



$word_1$ $word_2$

- RIGHT-ARC: Make the next word in the queue the dependent of the word at the top of the stack.

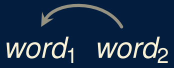


$word_1$ $word_2$

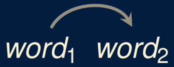
Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - Transitions

- Left-Arc: Make the next word in the queue the head of the word at the top of the stack.



- Right-Arc: Make the next word in the queue the dependent of the word at the top of the stack.

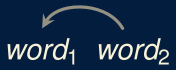


- Reduce: Remove the word at the top of the stack.

Introduction to Dependency Parsers

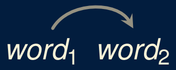
A Transition-Based System (Nivre 2008) - Transitions

- LEFT-ARC: Make the next word in the queue the head of the word at the top of the stack.



word$_1$ *word*$_2$

- RIGHT-ARC: Make the next word in the queue the dependent of the word at the top of the stack.



word$_1$ *word*$_2$

- REDUCE: Remove the word at the top of the stack.
- SHIFT: Move the next word from the queue onto the stack.

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - The Oracle

How does the Parser know which action to take?

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - The Oracle

How does the Parser know which action to take?

[illegible] [illegible] [illegible] [illegible] [illegible] [illegible]

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - The Oracle

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - The Oracle

The Oracle predicts the next transition based on all the information available at the time

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - The Oracle

The Oracle predicts the next transition based on all the information available at the time

`stack`	: Does/VBZ
`queue`	: serve/VB
`distance`	: 3
`DLT`	: 1
`intervenors`	: NN
`baseline act.`	: 0.0681
`retrieval`	: 0.0292 ms.
`SBI`	: 1.5

Introduction to Dependency Parsers

A Transition-Based System (Nivre 2008) - The Oracle

The Oracle predicts the next transition based on all the information available at the time

`stack`	: Does/VBZ
`queue`	: serve/VB
`distance`	: 3
`DLT`	: 1
`intervenors`	: NN
`baseline act.`	: 0.0681
`retrieval`	: 0.0292 ms.
`SBI`	: 1.5

Next Transition should be Right-Arc.

The Memory Model

Some math… (Lewis and Vasishth 2005)

The Memory Model

Some math… (Lewis and Vasishth 2005)

- A retrieval occurs when we use the word on the stack (LEFT-ARC, RIGHT-ARC)

The Memory Model

Some math… (Lewis and Vasishth 2005)

- A retrieval occurs when we use the word on the stack (Left-Arc, Right-Arc)
- Retrieval time is defined as: $T_i = Fe^{-A_i}$. ($F = 0.14$)

The Memory Model

Some math… (Lewis and Vasishth 2005)

- A retrieval occurs when we use the word on the stack (LEFT-ARC, RIGHT-ARC)
- Retrieval time is defined as: $T_i = Fe^{-A_i}$. ($F = 0.14$)
- Activation is defined as: $A_i = B_i + \sum_j W_j S_{ji}$. ($W_j = 1$)

The Memory Model

Some math… (Lewis and Vasishth 2005)

- A retrieval occurs when we use the word on the stack (LEFT-ARC, RIGHT-ARC)
- Retrieval time is defined as: $T_i = Fe^{-A_i}$. ($F = 0.14$)
- Activation is defined as: $A_i = B_i + \sum_j W_j S_{ji}$. ($W_j = 1$)
- Baseline Activation is defined as: $B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$.

The Memory Model

Some math… (Lewis and Vasishth 2005)

- A retrieval occurs when we use the word on the stack (LEFT-ARC, RIGHT-ARC)
- Retrieval time is defined as: $T_i = Fe^{-A_i}$. ($F = 0.14$)
- Activation is defined as: $A_i = B_i + \sum_j W_j S_{ji}$. ($W_j = 1$)
- Baseline Activation is defined as: $B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$.
- Similarity is defined as: $S_{ji} = S_{max} - \ln(\text{fan}_j)$. ($S_{max} = 1.5$)

The Memory Model

Some math… (Lewis and Vasishth 2005)

- A retrieval occurs when we use the word on the stack (LEFT-ARC, RIGHT-ARC)
- Retrieval time is defined as: $T_i = Fe^{-A_i}$. ($F = 0.14$)
- Activation is defined as: $A_i = B_i + \sum_j W_j S_{ji}$. ($W_j = 1$)
- Baseline Activation is defined as: $B_i = \ln \left(\sum_{j=1}^{n} t_j^{-d} \right)$.
- Similarity is defined as: $S_{ji} = S_{max} - \ln(\text{fan}_j)$. ($S_{max} = 1.5$)
- fan_j identifies the number of words already seen that have the same grammatical category as the cue j.

Past Experiments

Summary

The model, as developed by Boston et al has been shown to:

- Predict reading difficulty as measured in eye-tracking (M. Boston et al. 2008).
- Predict strong and week island constraint violations (M. F. Boston 2011)
- Predict garden path phenomena (M. F. Boston and Hale 2007)

The Experiment

Current Status

The Experiment

Current Status

- The key feature extractors have been written (with some bugs).

The Experiment

Current Status

- The key feature extractors have been written (with some bugs).
- When trained and tested on a very small corpus, the oracle is currently 85.85% accurate at selecting the next transition.

The Experiment

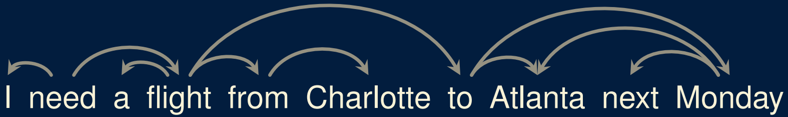
Current Status

- The key feature extractors have been written (with some bugs).
- When trained and tested on a very small corpus, the oracle is currently 85.85% accurate at selecting the next transition.
- Sentence-level performance is not great:

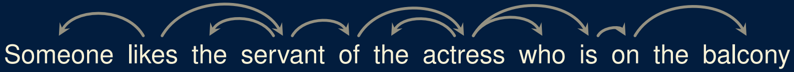
The Experiment

Current Status

- The key feature extractors have been written (with some bugs).
- When trained and tested on a very small corpus, the oracle is currently 85.85% accurate at selecting the next transition.
- Sentence-level performance is not great:



I need a flight from Charlotte to Atlanta next Monday



Someone likes the servant of the actress who is on the balcony

The Experiment

Next Steps

The Experiment

Next Steps

- Complete development of the model.

The Experiment

Next Steps

- Complete development of the model.
- Obtain sufficient training and testing data in two languages (most likely English and Italian).

The Experiment

Next Steps

- Complete development of the model.
- Obtain sufficient training and testing data in two languages (most likely English and Italian).
- Run a baseline experiment using just the Boston et al. features.

	English	Italian
Percent Attachment to first noun	?	?
Percent Attachment to second noun	?	?

The Experiment

Next Steps

- Complete development of the model.
- Obtain sufficient training and testing data in two languages (most likely English and Italian).
- Run a baseline experiment using just the Boston et al. features.

	English	Italian
Percent Attachment to first noun	?	?
Percent Attachment to second noun	?	?

- Explore impact of other features (e.g. phrase length).

Select References I

Boston, Marisa Ferrara (2011). *A processing model of the strong and weak island distinction*. Presentation at the CUNY Sentence Processing Conference.

– (2012). “A Computational Model Of Cognitive Constraints In Syntactic Locality”. PhD thesis. Cornell University.

Boston, Marisa Ferrara and John T Hale (2007). “Garden-pathing in a statistical dependency parser”. In: *Proceedings of the Midwest Computational Linguistics Colloquium. West Lafayette, IN: Midwest Computational Linguistics Colloquium*.

Boston, Marisa et al. (2008). “Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus”. In: *Mind Research Repository* 1.

Select References II

Lewis, Richard L and Shravan Vasishth (2005). "An activation-based model of sentence processing as skilled memory retrieval". In: *Cognitive science* 29.3, pp. 375–419.

Nivre, Joakim (2006). "Dependency parsing". In: *Inductive Dependency Parsing*, pp. 45–86.

– (2008). "Sorting out dependency parsing". In: *Advances in Natural Language Processing*. Springer, pp. 16–27.

Tesnière, L. (1959). *Éléments de syntaxe structurale*. Éditions Klinksieck.