Bootstrapping Dependency Parsing from Web Scale Unstructured Data

Siva Reddy

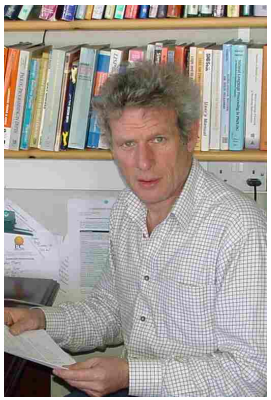
Lexical Computing Ltd, UK
http://sketchengine.co.uk

University of Malta
June 14 2012

in collaboration with Jan Joachimsen, Albert Gatt, Mike Rosner, and also with Anil Eragani, Varun Kuchibhotla

Acknowledgments

Adam Kilgarriff



Mike Rosner

Ray Fabri

Albert Gatt

Chris Staff

Outline

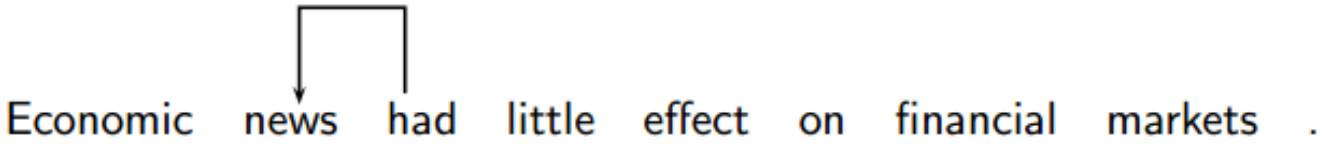
Dependency Structure

Economic news had little effect on financial markets .

1

[1] Image Courtesy: (McDonald and Nivre, 2007)

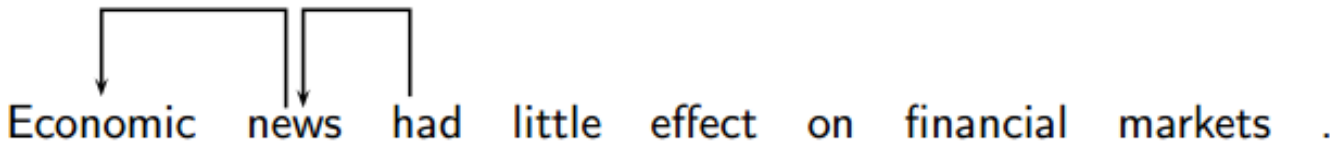
Dependency Structure



[2]

[2] Image Courtesy: (McDonald and Nivre, 2007)

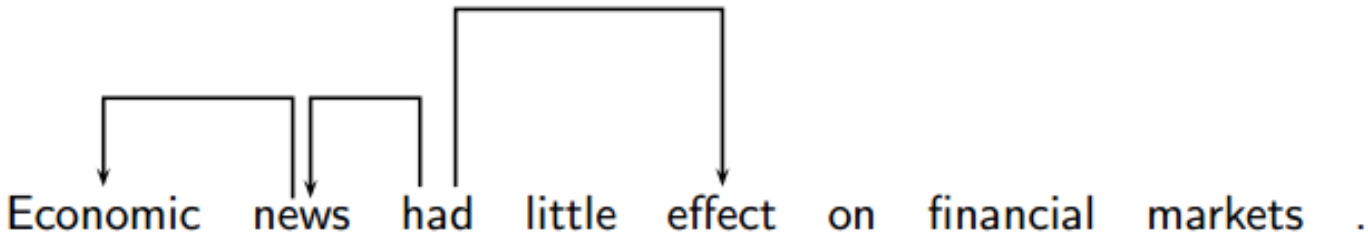
Dependency Structure



3

[3] Image Courtesy: (McDonald and Nivre, 2007)

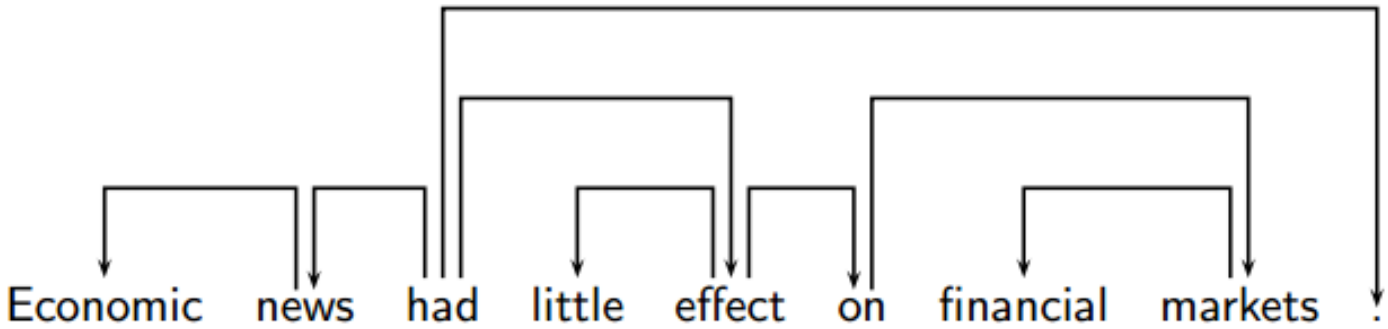
Dependency Structure



4

[4] Image Courtesy: (McDonald and Nivre, 2007)

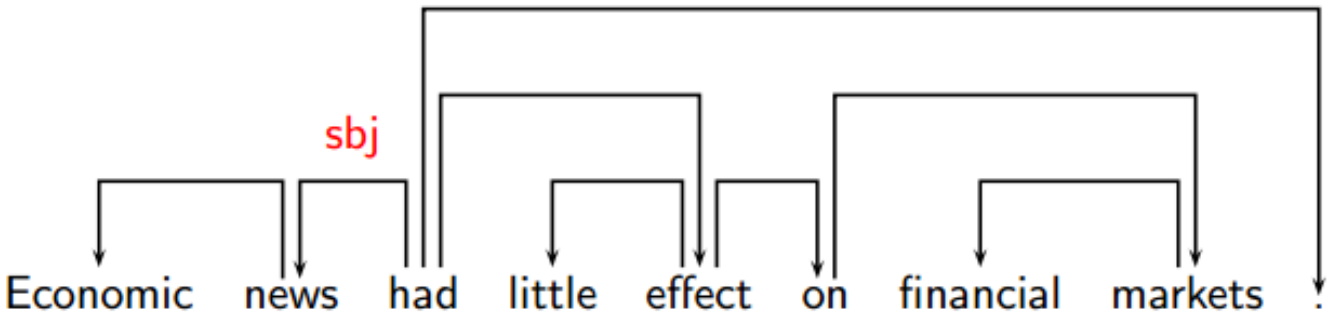
Dependency Structure



5

[5] Image Courtesy: (McDonald and Nivre, 2007)

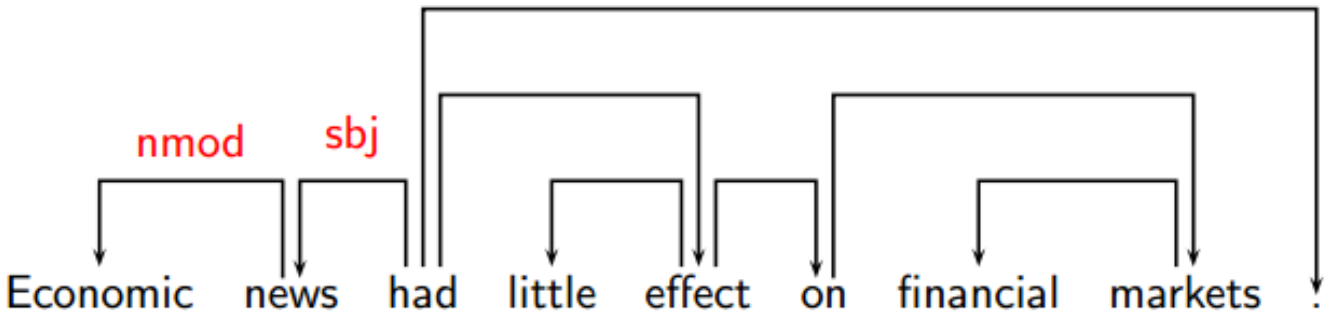
Dependency Structure



6

[6] Image Courtesy: (McDonald and Nivre, 2007)

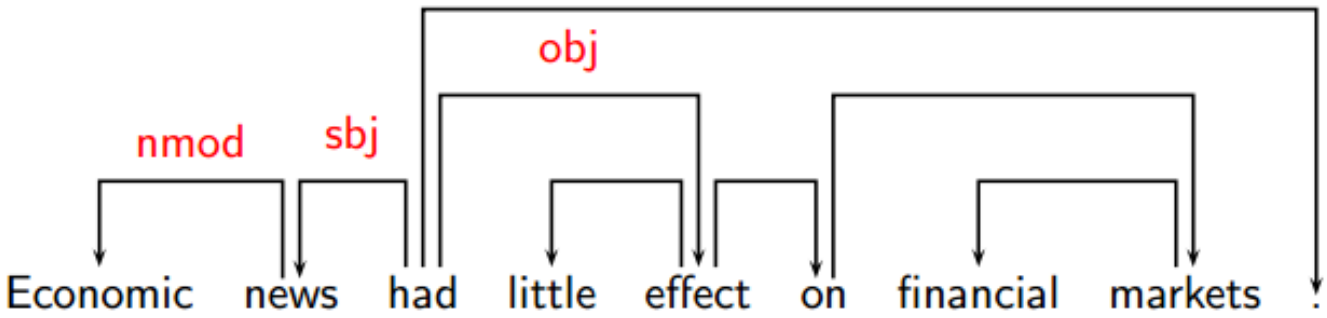
Dependency Structure



7

[7] Image Courtesy: (McDonald and Nivre, 2007)

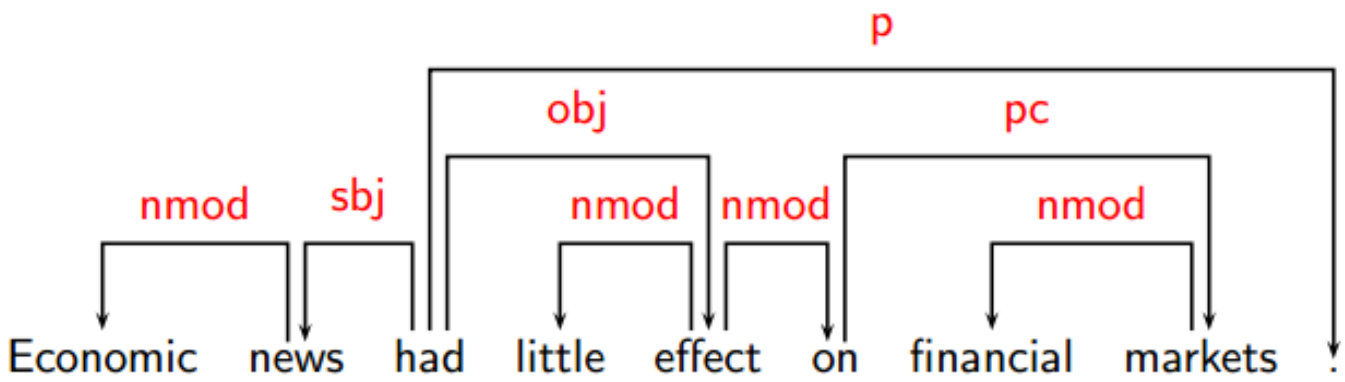
Dependency Structure



8

[8] Image Courtesy: (McDonald and Nivre, 2007)

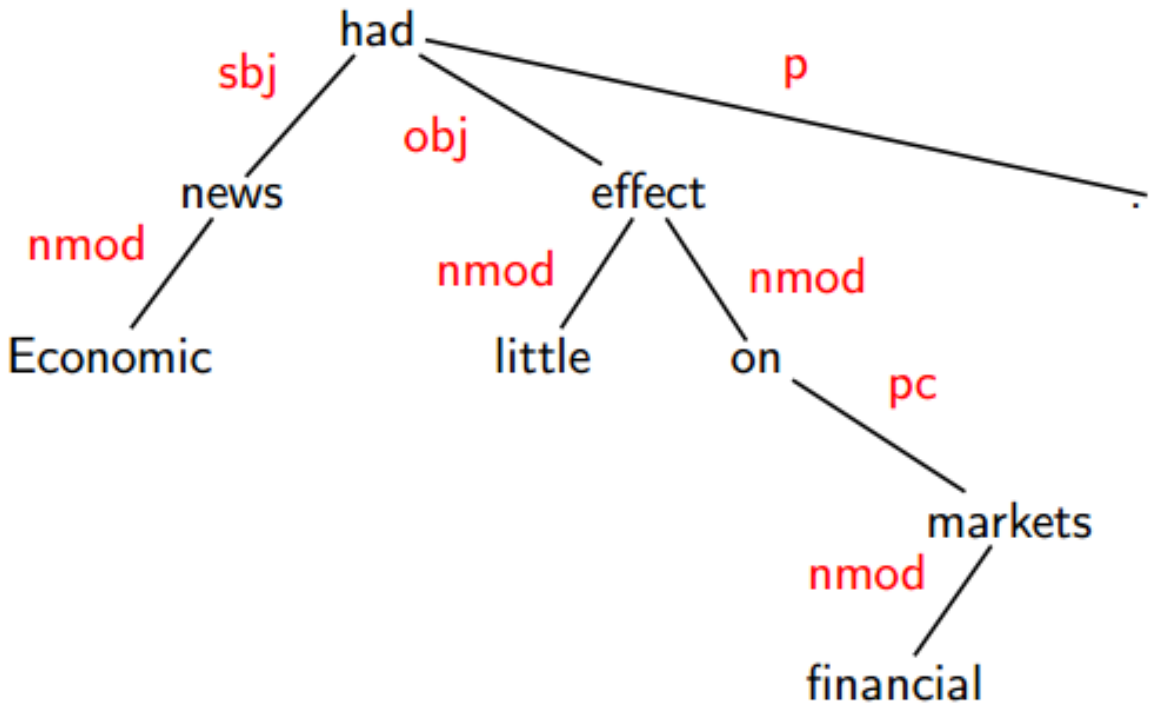
Dependency Structure



9

[9] Image Courtesy: (McDonald and Nivre, 2007)

Dependency Tree



10

[10] Image Courtesy: (McDonald and Nivre, 2007)

Data Driven Dependency Parsing[11]

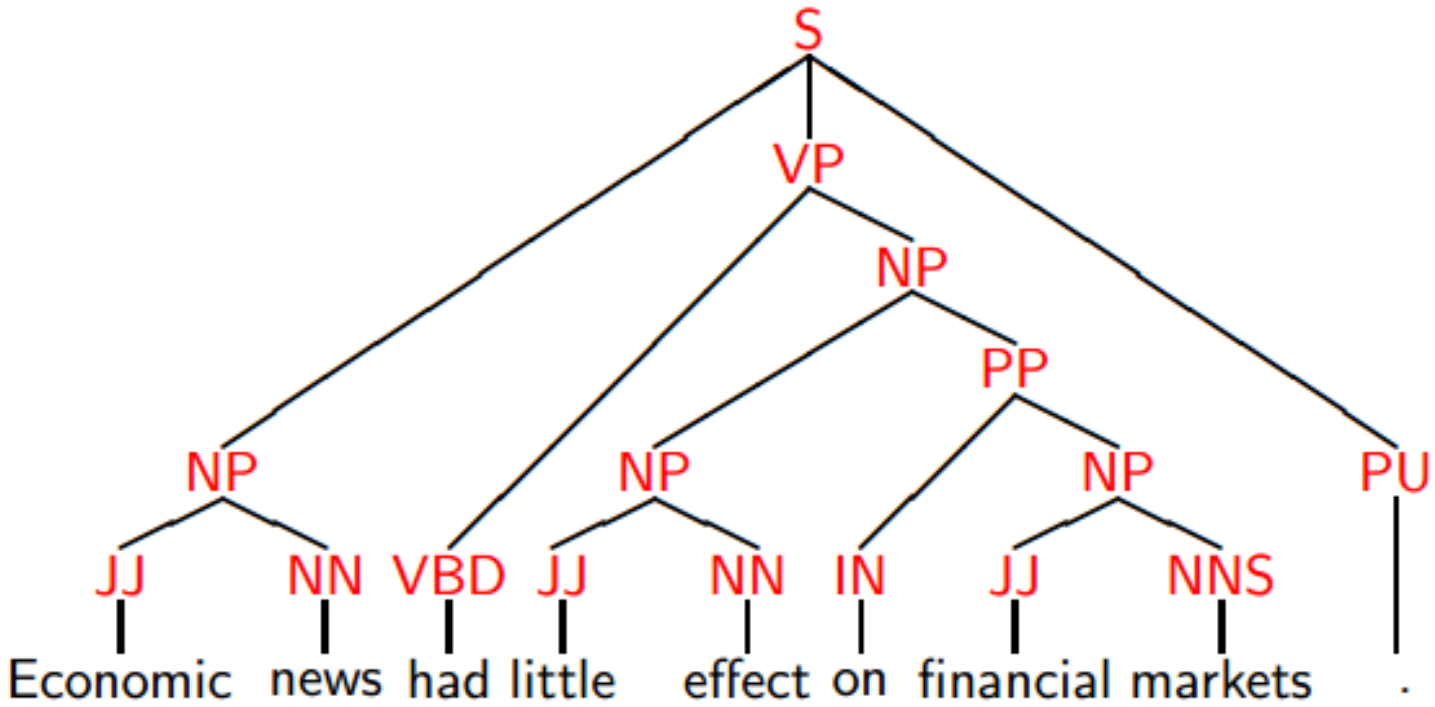
- Data Driven involves Machine Learning
- Parametrize a model
- **Supervised:** Learn parameters from annotated data

[11] Courtesy: (McDonald and Nivre, 2007)

Supervised Data Driven Dependency Parsing

- Classification problem
- Weighted Graph Minimization
- Constraint Programming
- Probabilistic Parsing

Grammar Driven Dependency Parsing

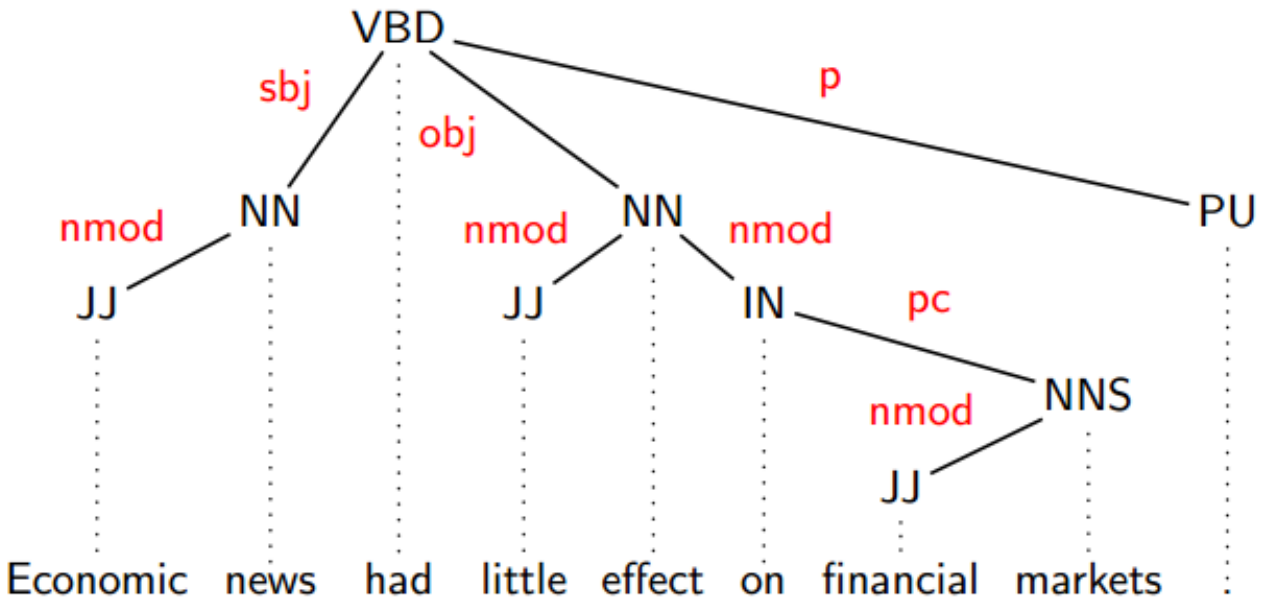


12

- S $\rightarrow$ NP VP
- NP $\rightarrow$ DT? JJ? NN
- Dependency tree is extracted from the above tree using additional rules

12 Image Courtesy: (McDonald and Nivre, 2007)

Grammar Driven Dependency Parsing



13

- NMOD → DEP:NN **HEAD:NN**
- SBJ → DEP:NN **HEAD:V.***
- OBJ → HEAD:V.* DT? JJ? **DEP:N.***

13 Image Courtesy: (McDonald and Nivre, 2007)

Pros and Cons

- Pros: Data Driven Models have higher accuracy
- Pros: Grammar Models are applicable to unseen data i.e. higher coverage
- Cons: Supervised Data Driven Models rely on annotated tree banks
- Cons: Hard to write lexical level grammar (i.e. not just POS tag based)

Better Solution

- *A method that uses unannotated data - exploit Web Scale Unstructured data*
- *A method that uses lexical level grammar e.g. the objects of eat-v are preferably sandwich, apple etc.*

Pros and Cons

- Pros: Data Driven Models have higher accuracy
- Pros: Grammar Models are applicable to unseen data i.e. higher coverage
- Cons: Supervised Data Driven Models rely on annotated tree banks
- Cons: Hard to write lexical level grammar (i.e. not just POS tag based)

Better Solution

- *A method that uses unannotated data - exploit Web Scale Unstructured data*
- *A method that uses lexical level grammar e.g. the objects of eat-v are preferably sandwich, apple etc.*

Outline

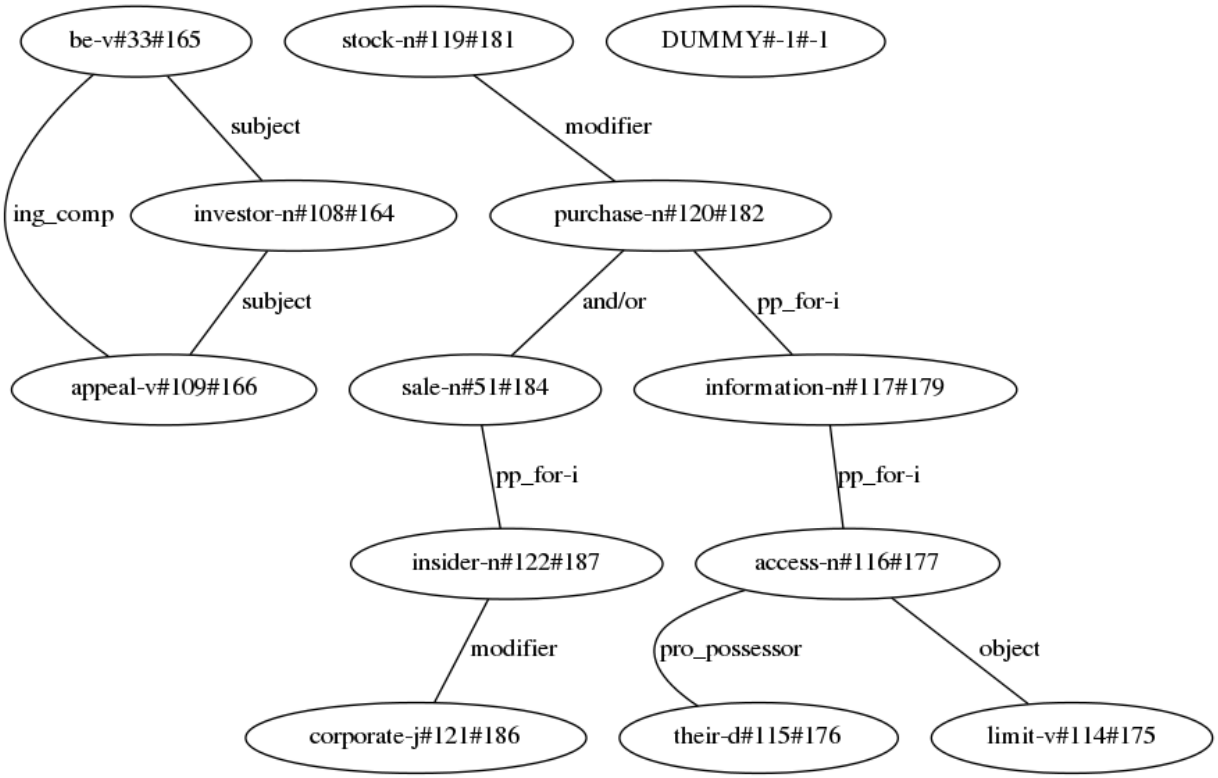
Sketch Grammar

- A grammar to extract word profiles, the Word Sketches
- Developed by Sketch Engine `http://sketchengine.co.uk`
- Grammar describing the relation between a target word and its dependent, constrained on the surrounding context.
- Word Sketch of play (verb) `http://bit.ly/MQjqBA`

Sketch Grammar Rules

- built using Corpus Query Language, a regular expression based language to search in a corpus
- Demo: All the words that start with “t” and are *verbs*: [word="t.*" & tag="V.*"]
- usually tag level patterns are represented
- Sketch Grammar for object relations: `http://bit.ly/MQlCsM`
- May violate the dependency constraint - “a word can have only one head”

Dependency Graph (not tree)



Outline

Our Parser: Grammar + Unstructured Data Driven

- Combination of strengths of Grammar and Data Driven Parsing
- Word Sketches describe syntactic preferences in each relation
  - Built from very large unstructured corpus
  - The voice of the majority at lexical level
  - Word Sketch of a word can be treated as its lexical level grammar
  - Tag level grammar -> lexical level grammar

Our Parser: Weighted Dependency Graph

- Build a graph from Sketch Grammar
- Weigh the edges using Word Sketches (knowledge from web-scale data)
- Extract the dependency tree out of the graph optimally
- Can be iterated further by building word sketches from extracted dependency trees

Our Parser: Weighted Dependency Graph

- Build a graph from Sketch Grammar
- Weigh the edges using Word Sketches (knowledge from web-scale data)
- Extract the dependency tree out of the graph optimally
- Can be iterated further by building word sketches from extracted dependency trees

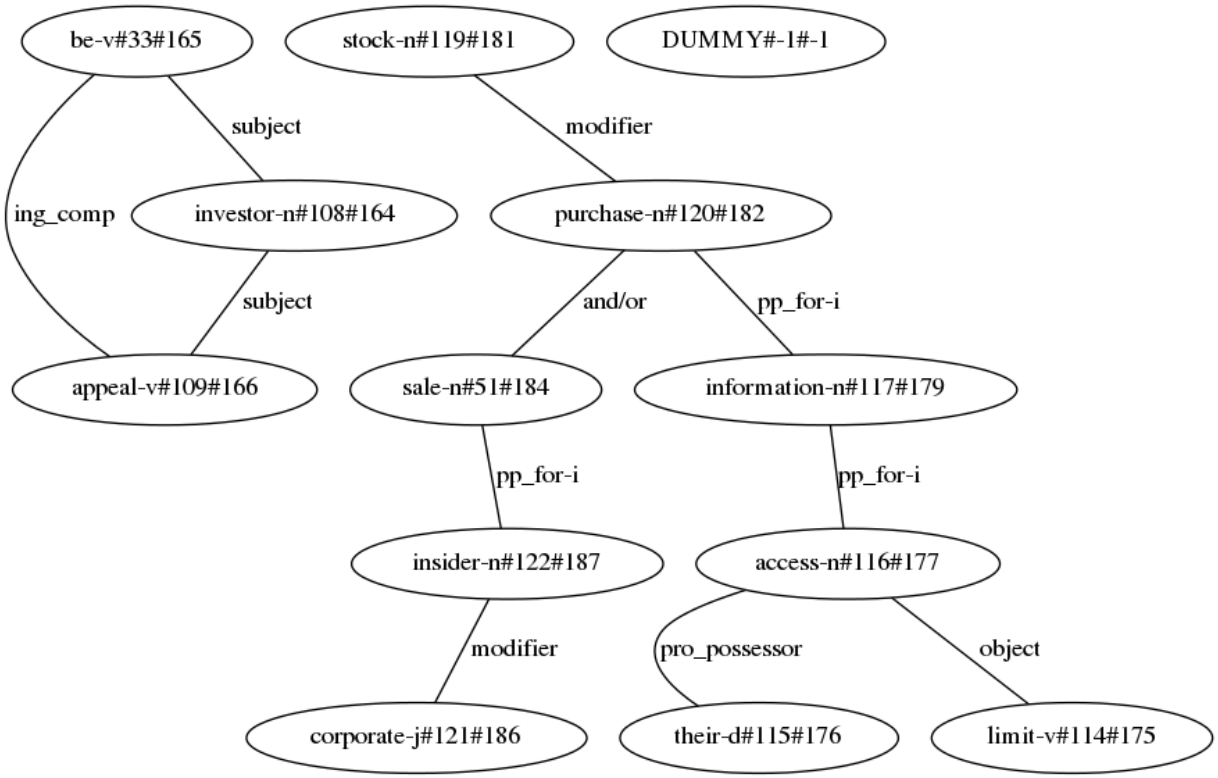
Our Parser: Weighted Dependency Graph

- Build a graph from Sketch Grammar
- Weigh the edges using Word Sketches (knowledge from web-scale data)
- Extract the dependency tree out of the graph optimally
- Can be iterated further by building word sketches from extracted dependency trees

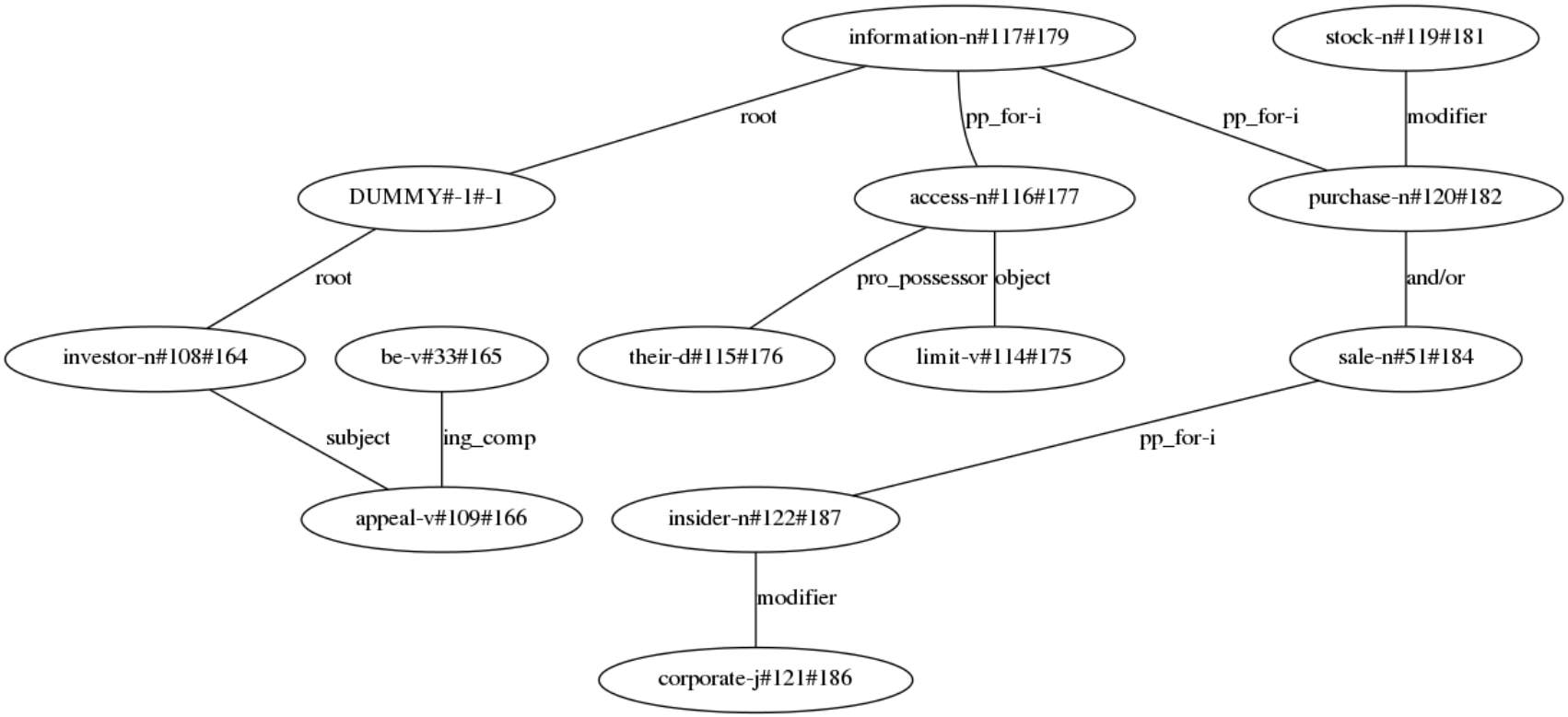
Our Parser: Weighted Dependency Graph

- Build a graph from Sketch Grammar
- Weigh the edges using Word Sketches (knowledge from web-scale data)
- Extract the dependency tree out of the graph optimally
- Can be iterated further by building word sketches from extracted dependency trees

Dependency Graph



Dependency Tree



Outline

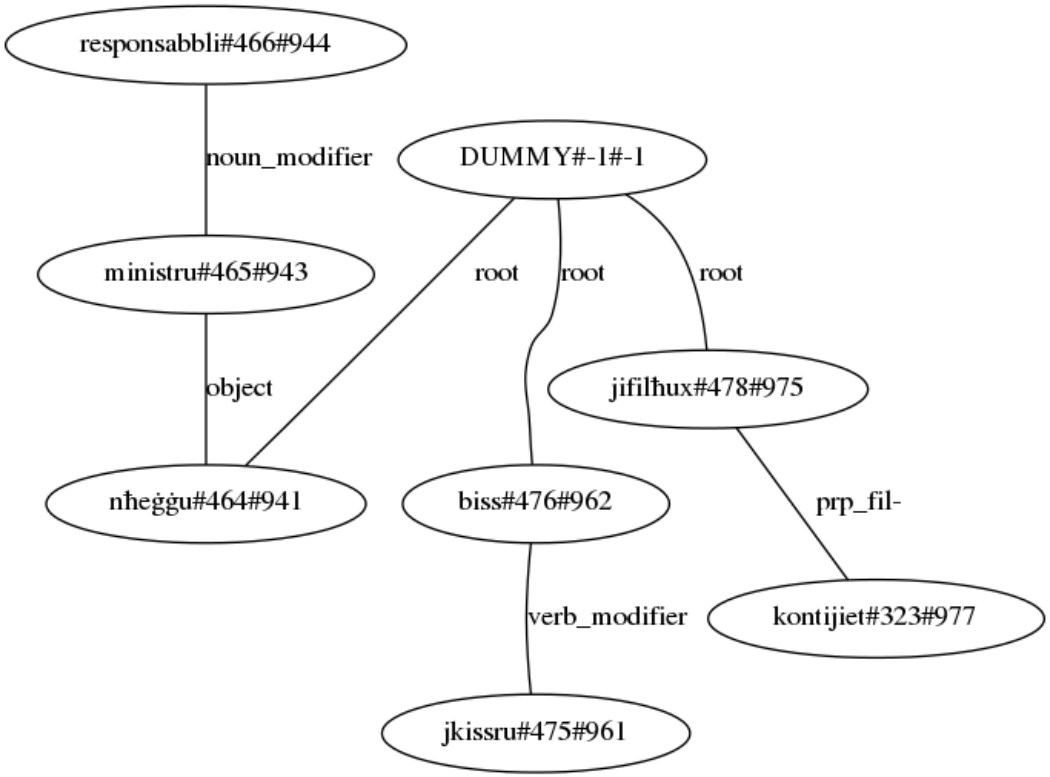
Ongoing work on Maltese: My involvement

- Good news: Maltese now have a POS tagger *[my minor role]*
- Recently written Sketch Grammar *[My major role: helping Jan in writing grammar]*
- Word Sketches: `http://bit.ly/MCyD52`
- Further development of Sketch Grammar to account for all dependency labels
- Build dependency tree bank using Grammar + Web-scale Maltese Data (MLRS data + METANET tools) *[My future work, includes many other languages]*
- Manually correct the errors in tree bank

Sample Maltese Dependency Trees: Manual

Show Tred Demo

Sample Maltese Dependency Trees: Automatic



Sketch Grammar Generation

- Observe patterns in tree bank (NMOD: approx 400 patterns)
- Generalize each pattern to create rules (NMOD: 196 rules)
- Merge the rules intelligently to create minimum number of rules (NMOD: 66 rules)
- Aiming for 10-20 rules
- The fruits will be seen soon

Summary

- exploiting grammar and web-scale unstructured data for dependency parsing
- Strengths from both the methods
- TODOs: special handling of certain grammatical relations: conjunction

A Parser Factory for many languages

Summary

- exploiting grammar and web-scale unstructured data for dependency parsing
- Strengths from both the methods
- TODOs: special handling of certain grammatical relations: conjunction

A Parser Factory for many languages