

Syntactic Parsing: Dependency structures

COMP61332: Text Mining
Week 2
Riza Batista-Navarro

Syntactic ambiguity

John swerved and hit a tree riding his bike.

Killer sentenced to die for second time in 10 years.

Syntactic ambiguity

I saw the man on the hill with a telescope.

The violent police man injures the farmer with an ax.

Flying planes can be dangerous.

Visiting relatives can be boring.

Syntactic ambiguity

I saw the man on the hill with a telescope.

The violent police man injures the farmer with an ax.

Flying planes can be dangerous.

Visiting relatives can be boring.

In order to understand sentences, we cannot treat each word in isolation.

We need to determine what a word is **attached** or connected to.

Dependencies

Recall: a sentence consists of a number of elements, i.e., **tokens**

There are **connections** between tokens which form the **structure** of the sentence

Formally speaking: **dependency** = connection

an asymmetric binary relation linking two lexical items (tokens): one is the **head** and the other is the **dependent**

Dependencies

Head - Dependent also known as:

Superior - Inferior

Governor - Modifier

Regent - Subordinate

How do we define **head**?

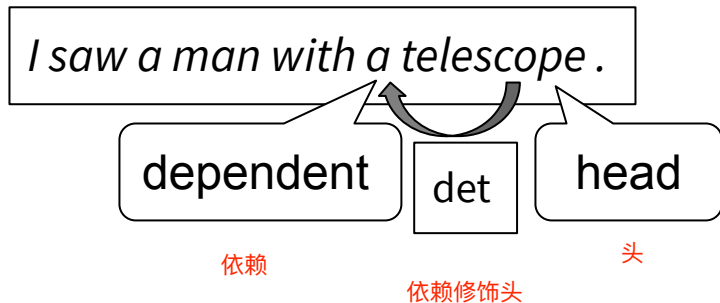
- The head of the sentence is the **main verb**; its subject and object are dependents
- For the rest of the tokens: check **if a token is *modifying* something** else
 - **modified token: *head***
 - **token that modifies (modifier): *dependent*** 依赖修饰头

Dependency structure

Sentence structure based on **dependencies**

Analysis of dependency structure:

- looking for dependencies between token pairs
- drawing a link between two tokens and specifying a **label:** grammatical function



Grammatical functions: Examples

Universal dependency relations

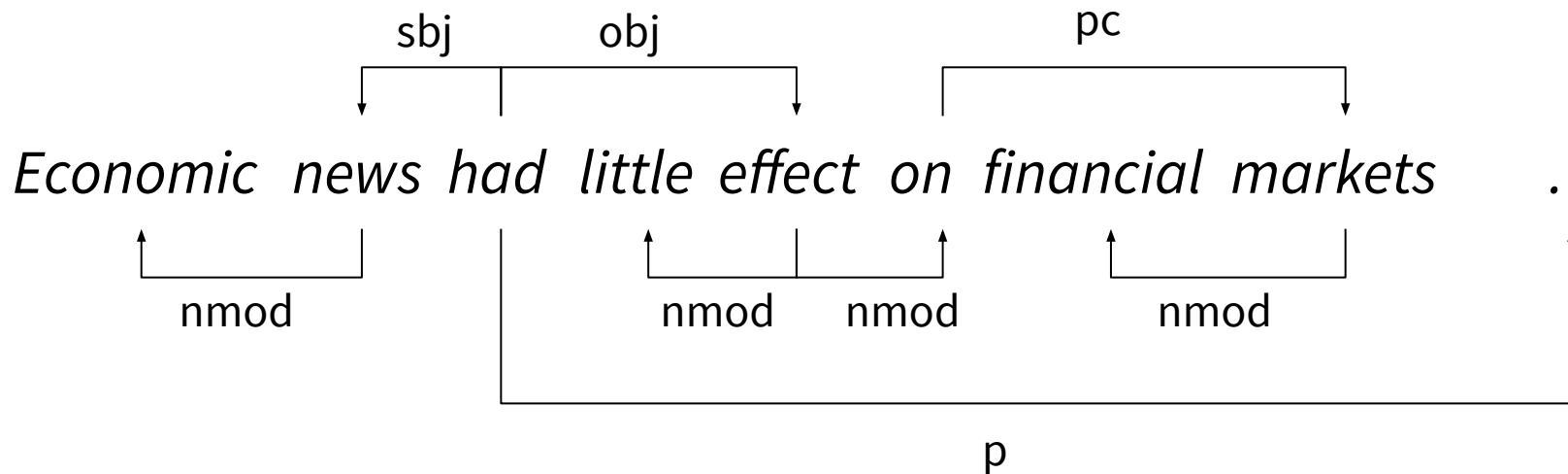
Label	Description	Example sentence	Example dependency
nmod/amod (nominal modifier)	phrase that modifies a noun phrase (NP)	Sam eats red meat.	nmod(meat, red)
aux (auxiliary)	non-main verb of a clause	He should leave.	aux(leave, should)
cc (coordination)	connects element of a conjunct and coordinating word	They either ski or snowboard.	cc(ski, or)
det (determiner)	between the head of an NP and its determiner	The man is here.	det(man, the)
obj/dobj (object)	between a verb and an NP denoting the entity acted upon	She gave me a raise.	obj(gave, raise)

Grammatical functions: Examples

Universal dependency relations

Label	Description	Example sentence	Example dependency
iobj (indirect object)	between a verb and an NP acting as indirect object	She gave me a raise.	iobj (gave, me)
vmod (verb modifier)	phrase that modifies a verb phrase (VP)	Genetically modified food.	vmod(modified, genetically)
sbj/nsubj (subject)	between a verb and an NP denoting the entity that serves as subject or agent	Clinton defeated Cole.	sbj(defeated, Clinton)
pobj/pc (object of preposition)	head of NP following a preposition	I sat on the chair.	pobj(on, chair)
p/punct (punctuation)	between the head of the sentence (the root) and punctuation	Go home!	punct(Go, !)

Example sentence



Dependency graph

A directed graph G with:

- a set of nodes V

- a set of edges E

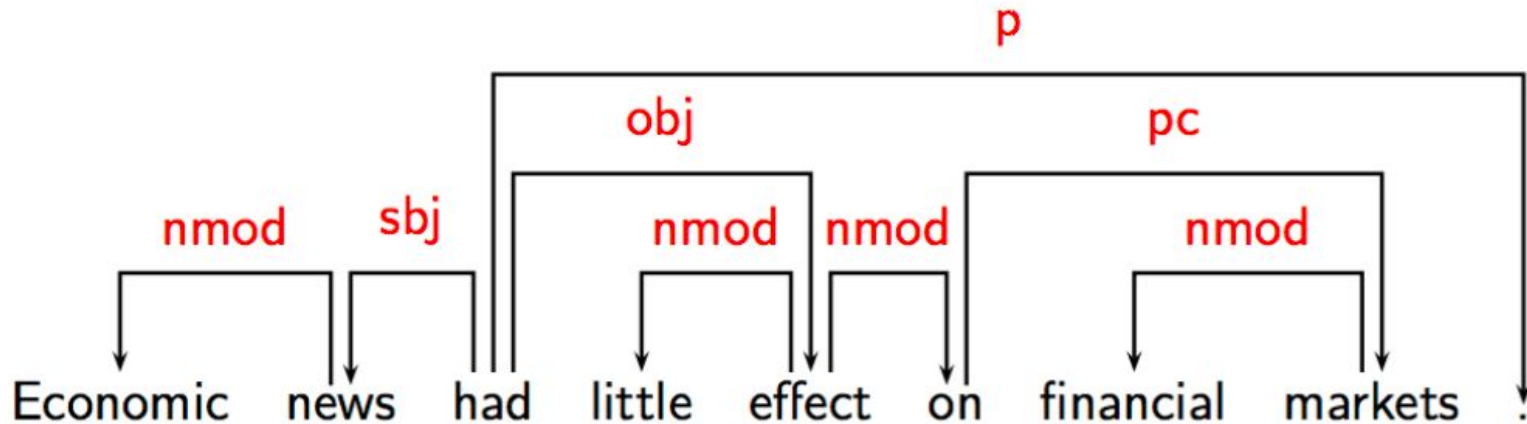
- with a linear precedence order on V

Nodes represent tokens (surface form, sometimes together with the POS tag)

Edges are labelled with relation type (grammatical functions)

Various notations

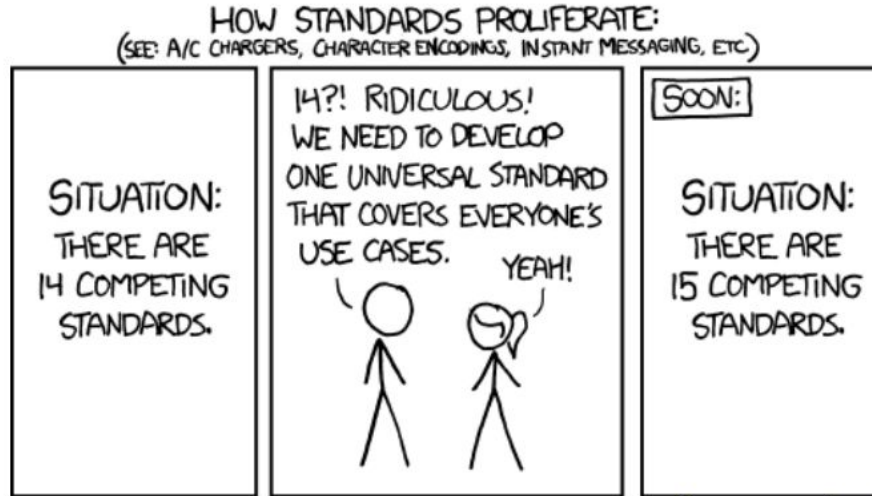
Notation 1: **Horizontally arranged tokens with directed edges**



Tokens + grammatical function

Various notations

There are far too many!

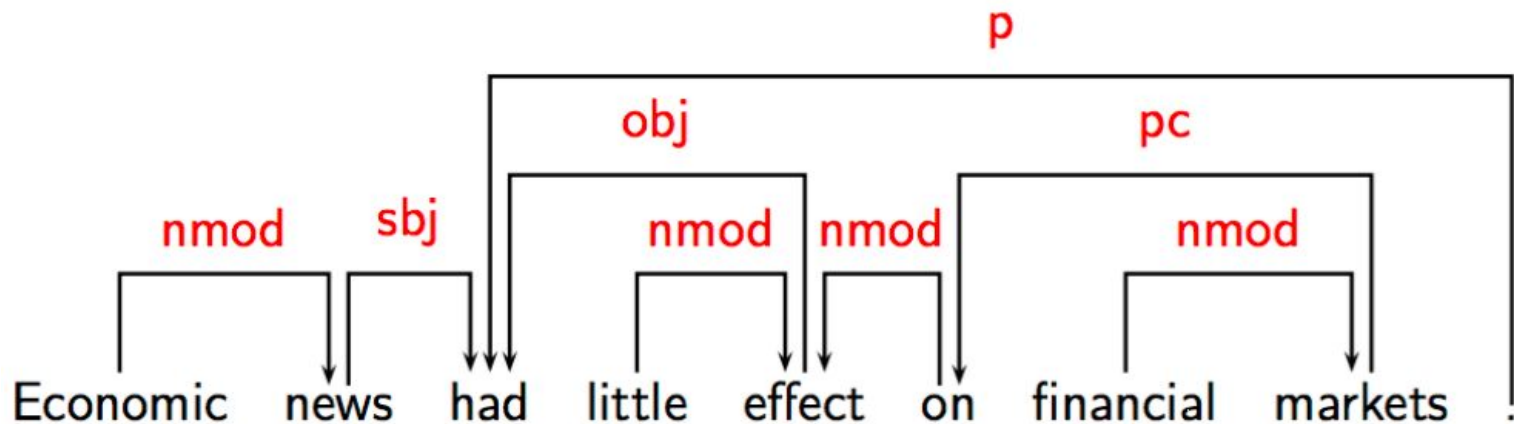


<https://xkcd.com/927/>

... But don't worry; these are just formalisms. You will still see conceptual similarities.

Various notations

Notation 2: **Horizontally arranged tokens with directed edges**

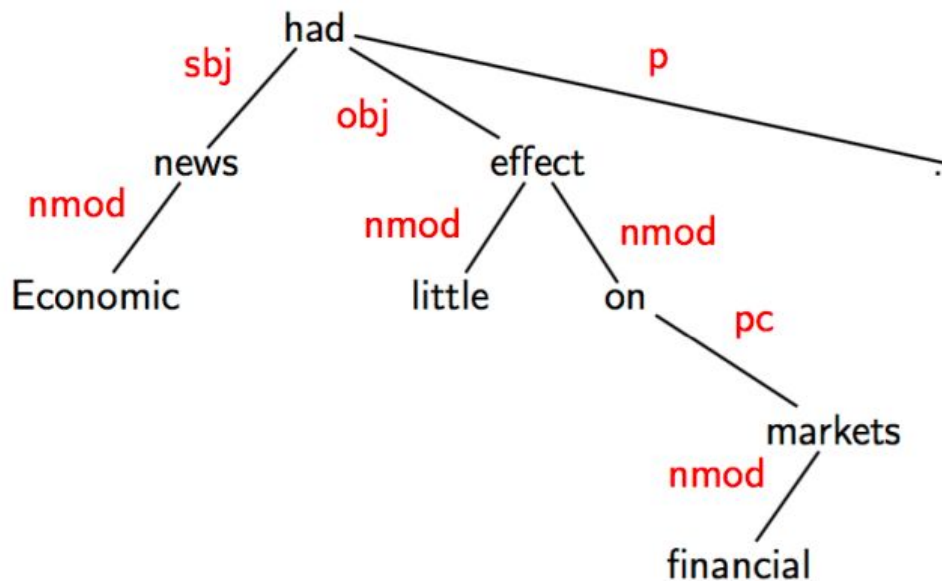


Tokens + grammatical function

... but arrows are in the opposite direction!

Various notations

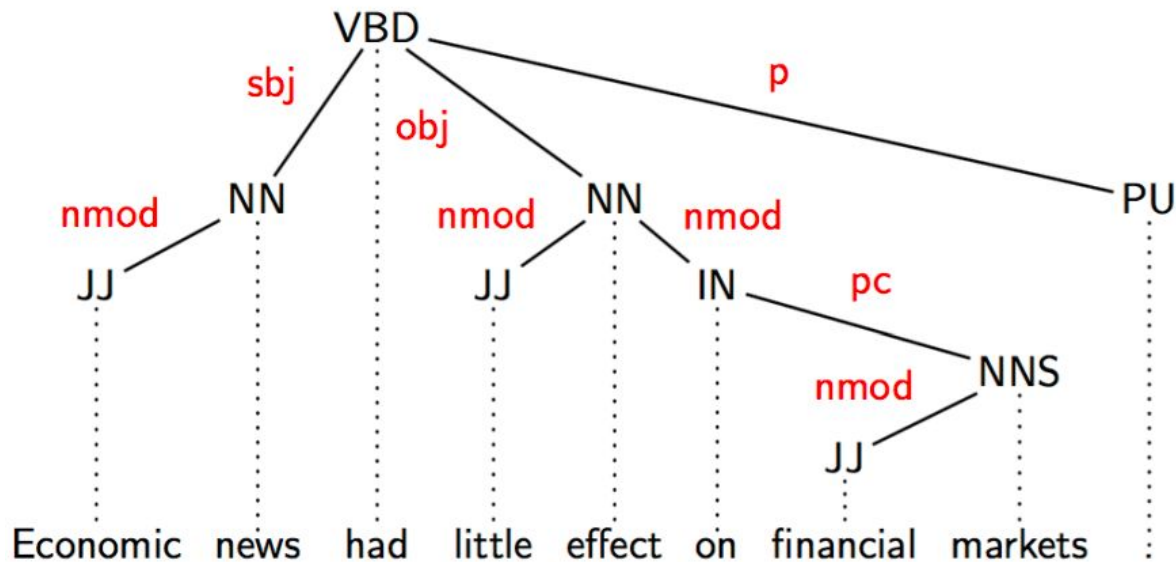
Notation 3: **Tree of tokens**



Tokens + grammatical function... Direction is naturally encoded by the tree

Various notations

Notation 4: **Tree of POS tags mapped to tokens**

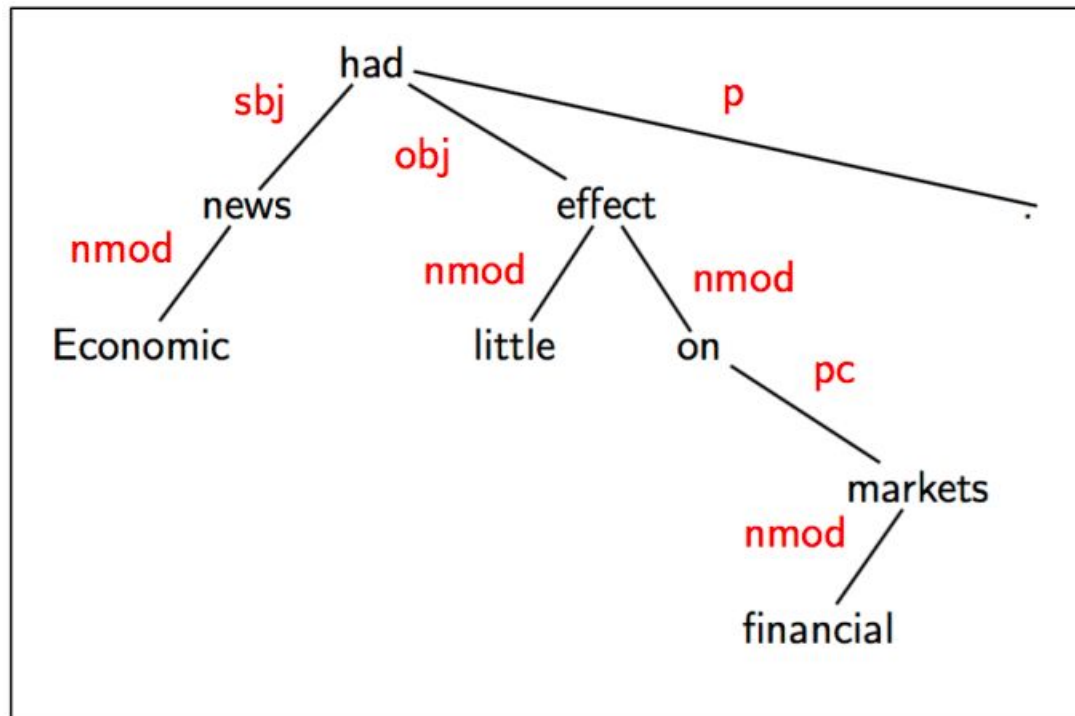


Tokens + grammatical function + **POS tags**

Machine-readable notation

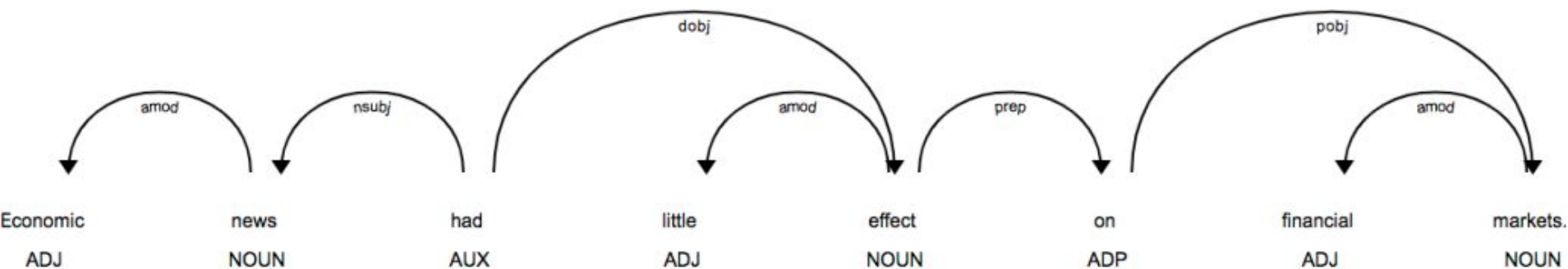
Using indentation

```
had
  <subj> news
    <nmod> Economic
  <obj> effect
    <nmod> little
    <nmod> on
      <pc> markets
        <nmod> financial
  <p> .
```



Machine-readable Notation in spaCy

Token attributes



Machine-readable notation in spaCy

Token attributes

Token	Lemma	POS	Dep label	Dependents (if any)
Economic	economic	ADJ	amod	[]
news	news	NOUN	nsubj	['Economic:amod']
had	have	AUX	ROOT	['news:nsubj', 'effect:dobj', '.:punct']
little	little	ADJ	amod	[]
effect	effect	NOUN	dobj	['little:amod', 'on:prep']
on	on	ADP	prep	['markets:pobj']
Financial	financial	ADJ	amod	[]
markets	market	NOUN	pobj	['financial:amod']
.	.	PUNCT	punct	[]