

# **Semantic Analysis**

Mariana Romanyshyn  
Grammarly, Inc.

# Contents

1. Word sense disambiguation
2. Semantic role labeling
3. Semantic parsing
4. Textual entailment

1.

# Word sense disambiguation

# Words have meanings

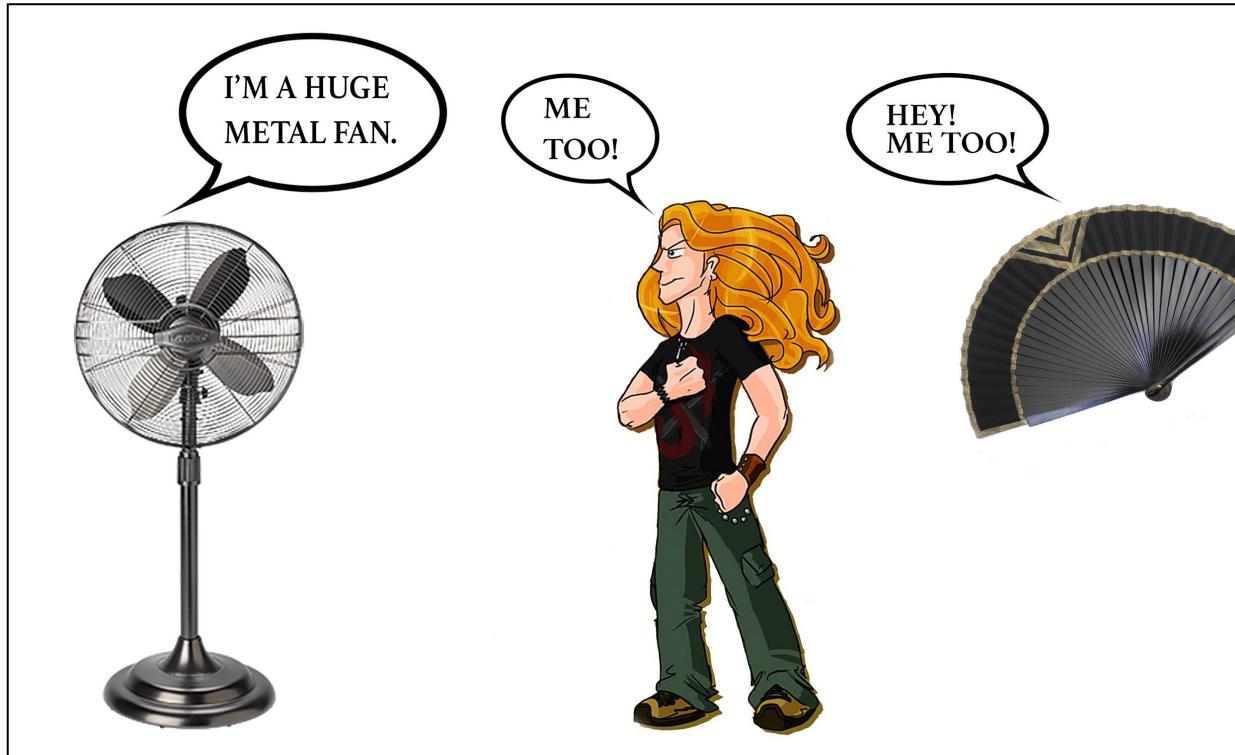


Image by Tetiana Turchyn

# Is it serious?

- ~40% of English words are polysemous
- most polysemous - verbs (~55% in WordNet)
- resources disagree
  - “head”, noun:
    - 11 meanings - Macmillan Dictionary
    - 16 meanings - Longman Dictionary
    - 33 meanings - WordNet
    - 34 meanings - Oxford Dictionary
- meanings overlap
  - *John works for the **newspaper** that you are reading.*

# Is it just English?

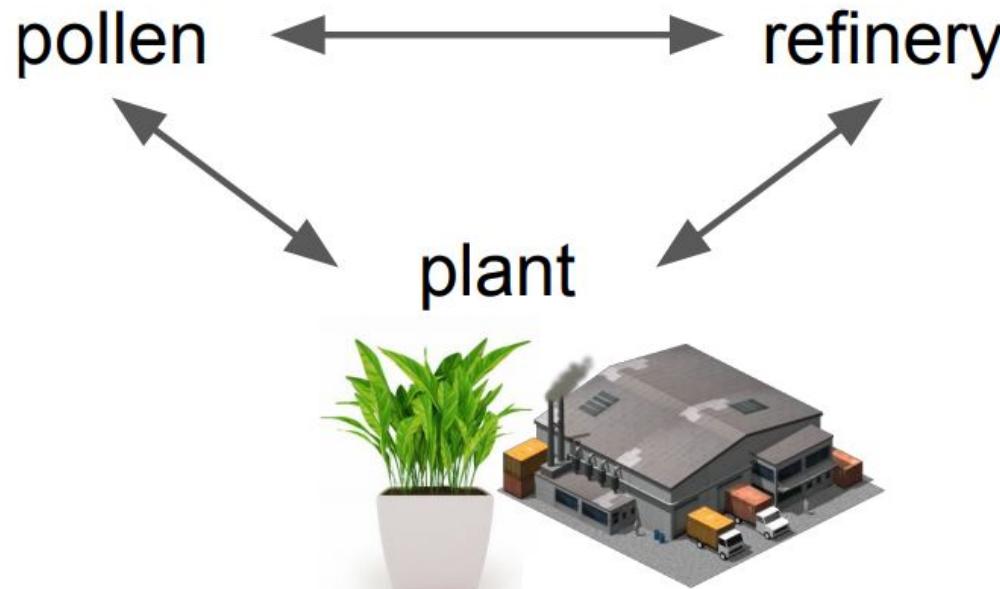
*... зробити так, щоби впала стіна?*

- стіна будинку
- стіни айсбергів
- мур
- те, що відокремлює, роз'єднує



# What does it mean for NLP?

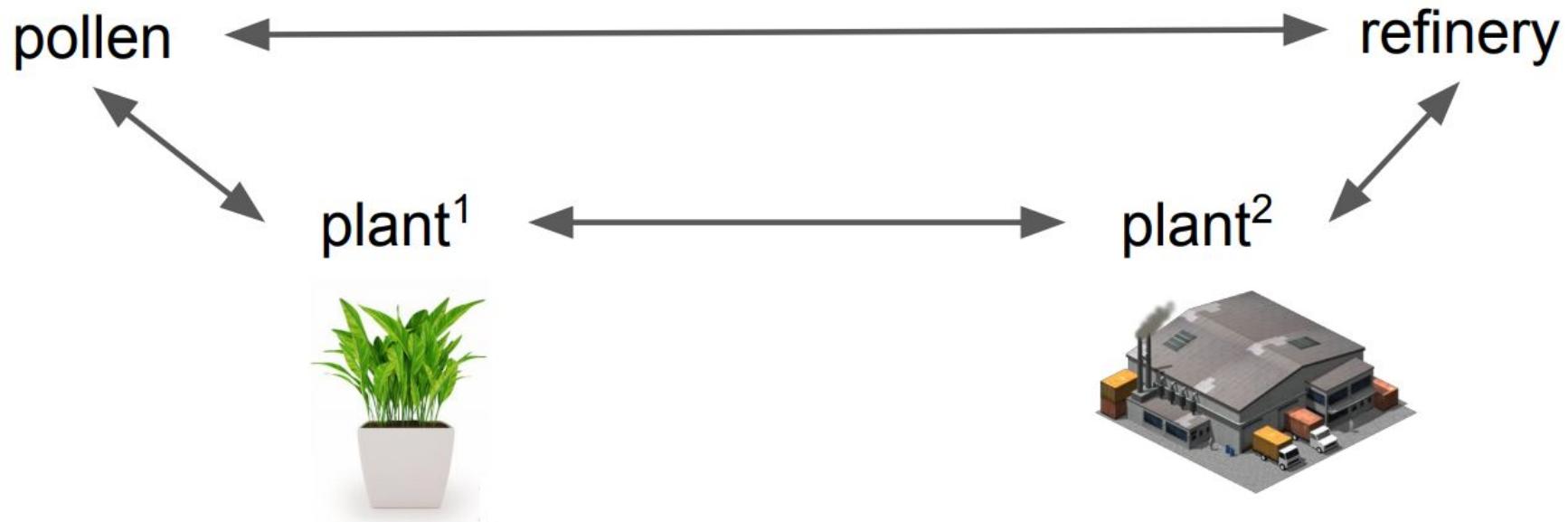
Triangle inequality in word embeddings.



Example from Neelakantan et al. (2014)

# What does it mean for NLP?

Word embeddings => sense embeddings



Example from Neelakantan et al. (2014)

# Text classification/mining

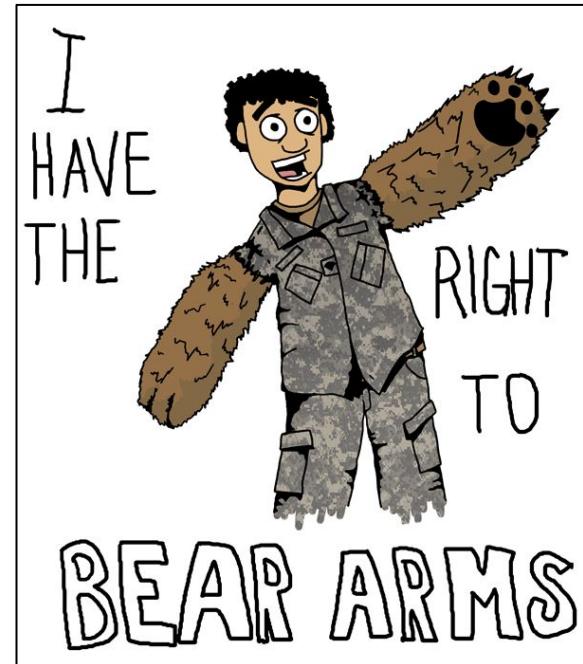
US **sells arms** to countries well-known for violating human rights.

Using recycled prosthesis, a hospital in Tanzania **sells arms** for around \$500 each. There is also high demand for legs.

# Text classification/mining

US **sells arms** to countries well-known for  
violating human rights.

Using recycled prostheses, a hospital in  
Tanzania **sells arms** for around \$500 each.  
There is also high demand for legs.



# Machine translation

The screenshot shows a machine translation interface. On the left, the input text is "I'm a huge metal fan, and I listen to the music all the time!" with a character count of 61/5000. On the right, the translated text is "Я величезний металевий вентилятор, і я слухаю музику весь час!" with a pencil icon for editing. The top bar shows language selection boxes for English, Spanish, French, Detect language, Ukrainian, English, Russian, and a blue "Translate" button.

I'm a huge metal fan, and I listen to the music all the time! 61/5000

Я величезний металевий вентилятор, і я слухаю музику весь час!

Example from Google Translate

# Machine translation

The screenshot shows the Google Translate interface. On the left, the input text is "I'm a huge metal fan, and I listen to the music all the time!" with a character count of 61/5000. On the right, the translated text is "Я огромный металлический поклонник, и я слушаю музыку все время!" Below the text are various interaction icons like star, copy, and share.

English Ukrainian Spanish Detect language ↻ Ukrainian English Russian Translate

I'm a huge metal fan, and I listen to the music all the time! 61/5000

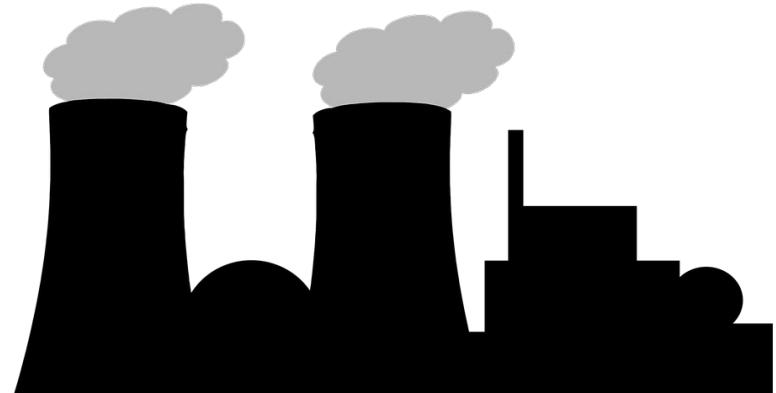
Я огромный металлический поклонник, и я слушаю музыку все время!

Example from Google Translate

# Personal assistants

**You:** I need to buy a big **plant** for my mom. She likes gardening!

**Siri:** Hmm...



# Sentiment analysis

Interest rates are very **high**.

These socks are a little **high**.

This area is **rich** in natural resources.

These comments are a bit **rich** coming from someone with no money worries.

# Sentiment analysis

Interest rates are very **high**.

These socks are a little **high**. (= smelly)

This area is **rich** in natural resources.

These comments are a bit **rich** coming from someone with no money worries.

# Sentiment analysis

Interest rates are very **high**.

These socks are a little **high**. (= smelly)

This area is **rich** in natural resources.

These comments are a bit **rich** coming from someone with no money worries.

# Error correction

**Abstract or concrete?**

**Man** is rapidly destroying the earth.

Do you recognize **man** in the grey suit?

# Error correction

**Abstract or concrete?**

**Man** is rapidly destroying the earth.

Do you recognize **the man** in the grey suit?

# Error correction

## Animate or inanimate?

Then we have 2 laptop stands & 2 **mouses** left.

My cat can easily catch **mouses**.

The **chair** was placed in the museum. He's part of the exhibit now.

The **chair** was awarded for a poem. He's famous now.

# Error correction

## Animate or inanimate?

Then we have 2 laptop stands & 2 **mouses** left.

My cat can easily catch {**mouses=>mice**}.

The **chair** was placed in the museum. He's part of the exhibit now.

The **chair** was awarded for a poem. He's famous now.

# Error correction

## Animate or inanimate?

Then we have 2 laptop stands & 2 **mouses** left.

My cat can easily catch {**mouses=>mice**}.

The **chair** was placed in the museum. {**He=>It**} 's part of the exhibit now.

The **chair** was awarded for a poem. He's famous now.

# What is “sense”?

- senses = domains?
- senses = sentiments?
- senses = animate/inanimate?
- senses = jargon/standard?
- senses = countable/uncountable?
- senses = senses?

# Dictionaries

## bank (*plural* **banks**)

1. (*hydrology*) An **edge** of river, lake, or other **watercourse**. [quotations ▼]
2. (*nautical, hydrology*) An elevation, or rising ground, under the sea; a shallow area of shifting **sand**, **gravel**, **mud**, and so forth (for example, a **sandbox** or **mudbank**).

*the **banks** of Newfoundland*

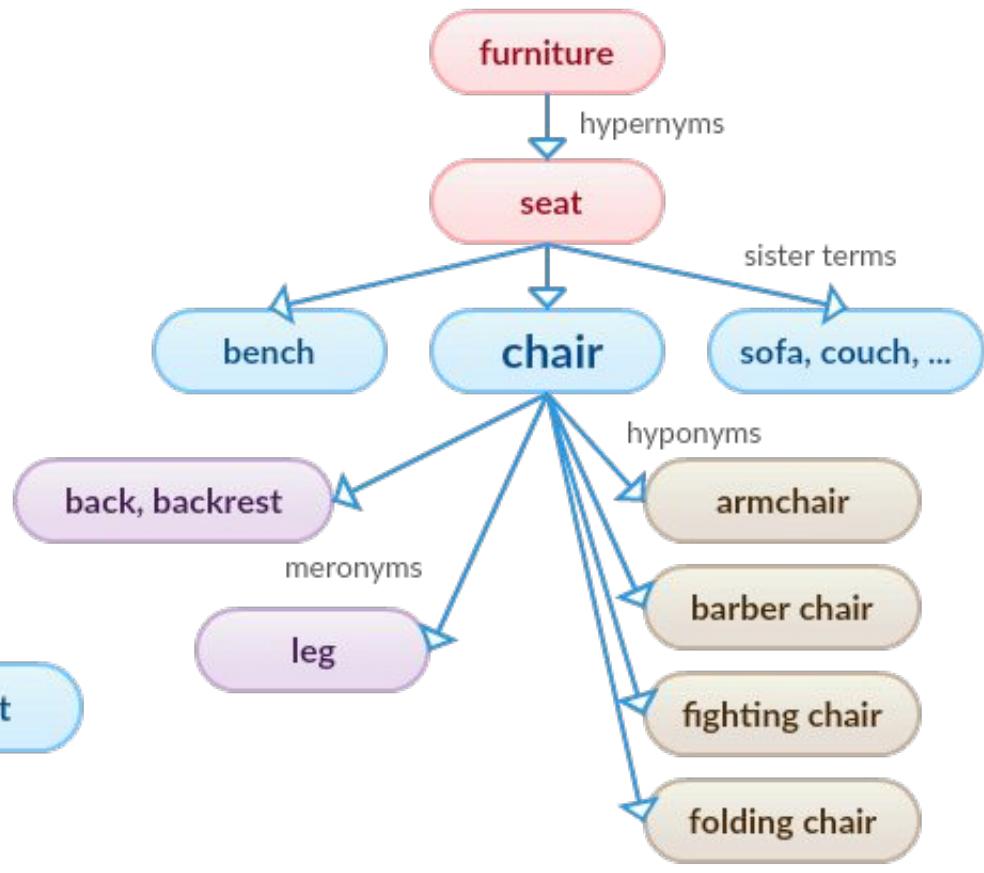
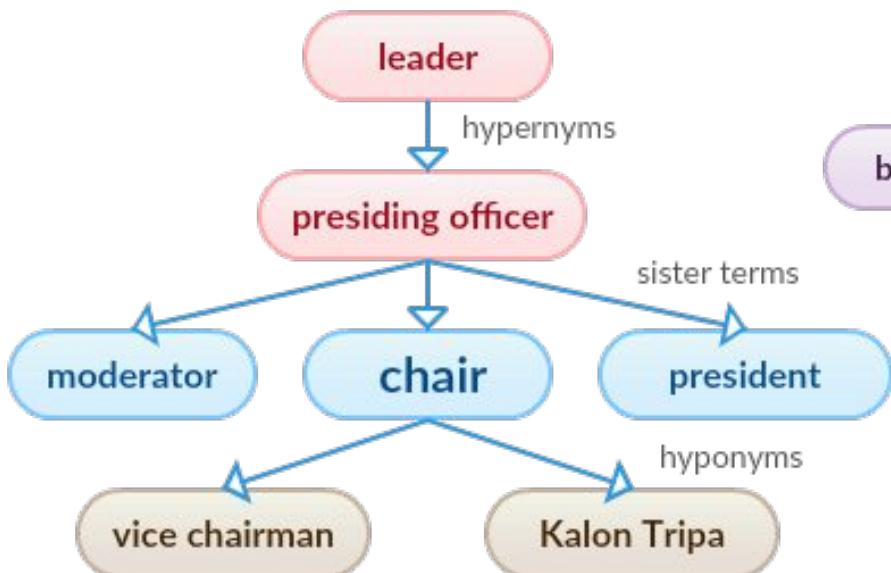
3. (*geography*) A **slope** of earth, sand, etc.; an **embankment**.
4. (*aviation*) The **incline** of an aircraft, especially during a turn.
5. (*rail transport*) An **incline**, a **hill**.

# Dictionaries

**man<sup>1</sup>** /mæn/ ●●● **S1** **W1** noun (*plural men /men/*)  

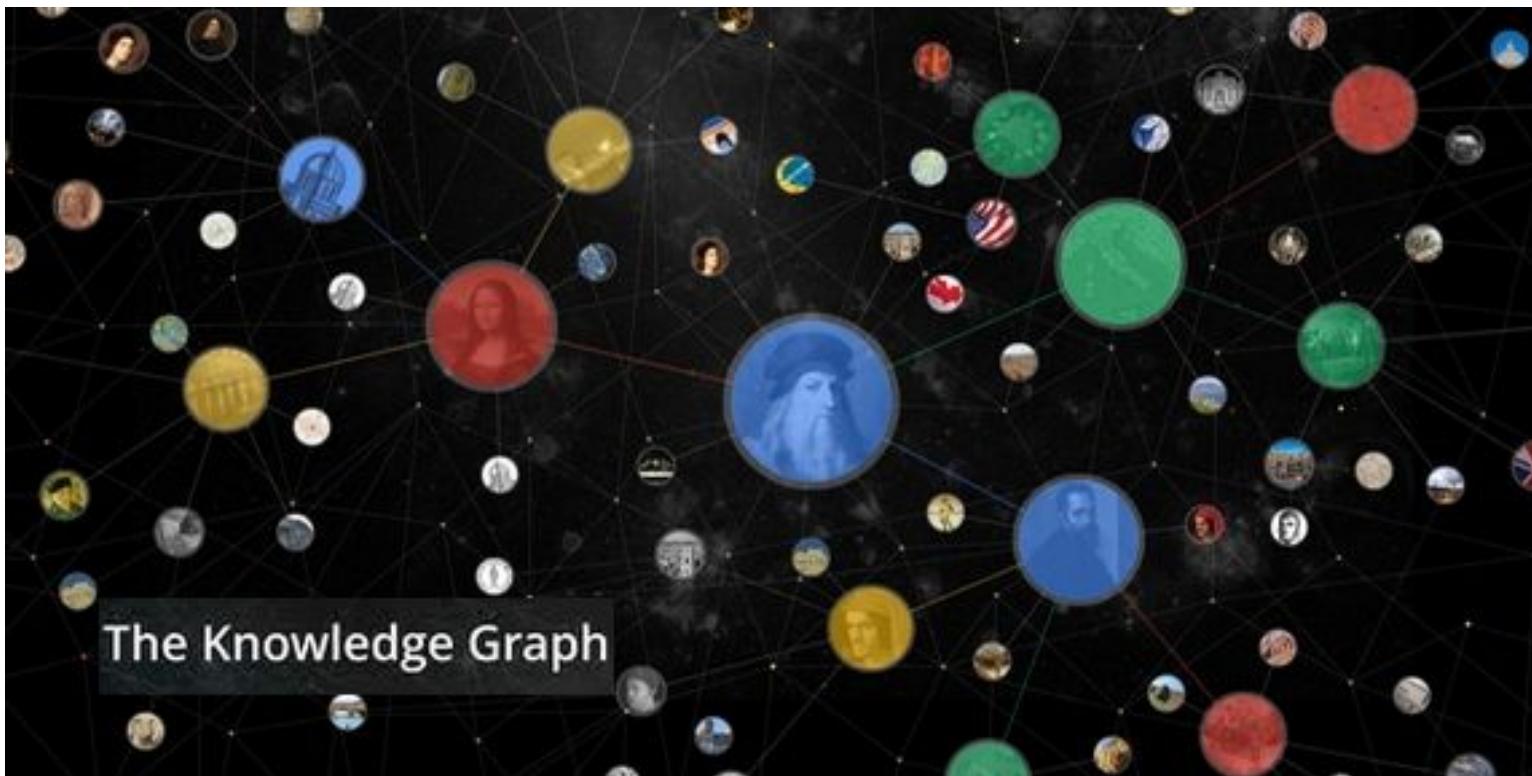
- 1 **MALE PERSON** [countable] an adult male human → **woman**
- 2 **STRONG/BRAVE** [countable usually singular] a man who has the qualities that people think a man should have, such as being brave, strong etc
- 3 **PERSON** [countable] a person, either male or female – used especially in formal situations or in the past
- 4 **PEOPLE** [uncountable] people as a group

# Ontologies



Example of relations in WordNet

# Knowledge Graph



# Wikipedia, Wikidata, DBpedia

## Finance [ edit ]

---

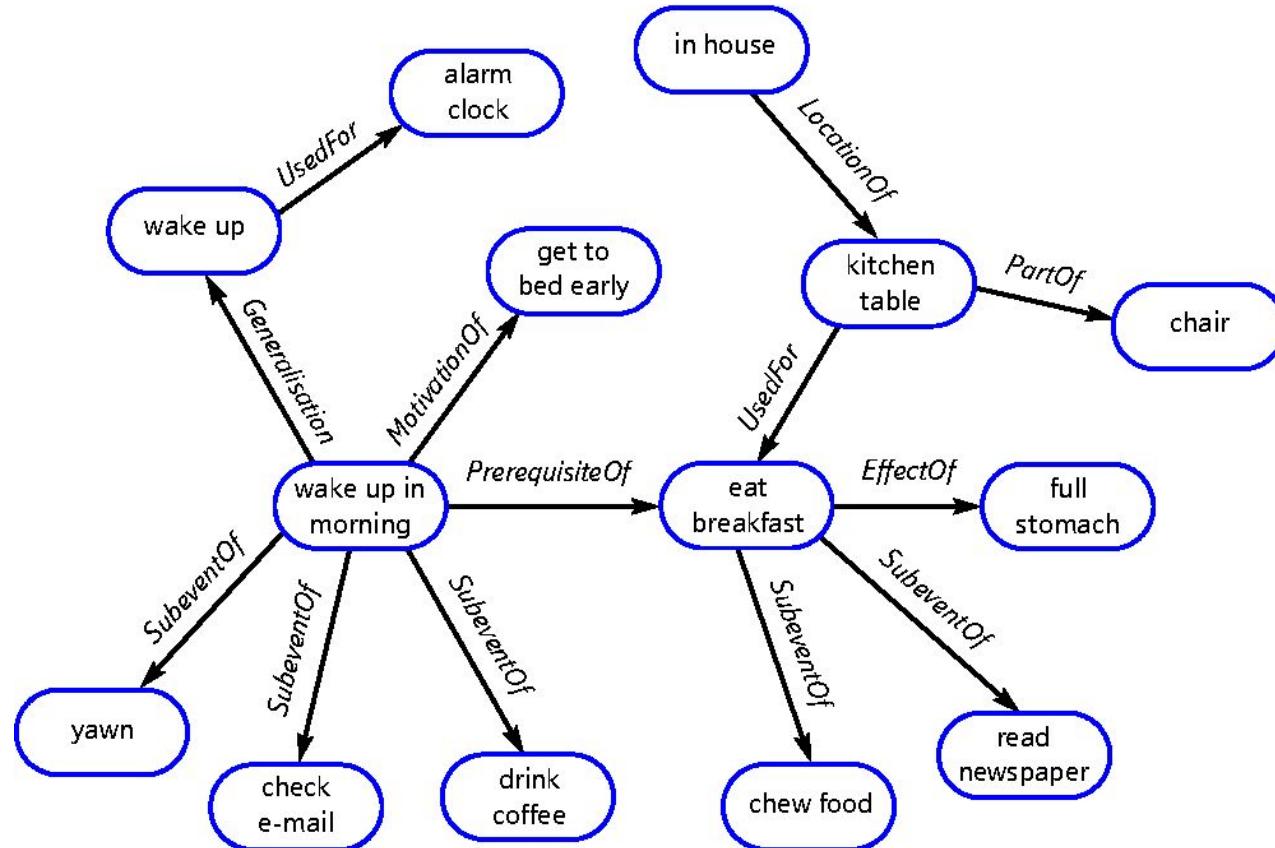
- [Central bank](#)
- [Mutual savings bank](#)
- [Savings bank](#)

## Natural geography [ edit ]

---

- [Bank \(geography\)](#), a raised portion of seabed or sloping ground along the edge of a stream, river, or lake
- [Ocean bank \(topography\)](#)
- [Ocean bank](#), a shallow area in a body of water
- [Stream bank or riverbank](#), a terrain alongside the bed of a river, creek, or stream

# ConceptNet



8 mln nodes  
21 mln edges  
83 languages  
36 relation types

Represents  
human knowledge  
about the world.

# BabelNet

Noun



**fan, mechanical fan, ventilator**

A device for creating a current of air by movement of a surface or surfaces

ID: [00033599n](#) | Concept

13.8 mln nodes

380 mln edges

284 languages



**fan, rooter, sports fan**

An enthusiastic devotee of sports

ID: [00033600n](#) | Concept

# Corpora: SemCor

```
<wf>The</wf>  
<wf lemma="model" wnsn="3">model</wf>  
<wf lemma="quite" wnsn="1">quite</wf>  
<wf lemma="plainly" wnsn="1">plainly</wf>  
<wf lemma="think" wnsn="1">thought</wf>  
<wf lemma="person" wnsn="1">Michelangelo</wf>  
<wf lemma="crazy" wnsn="1">crazy</wf>  
<wf>;</wf>
```

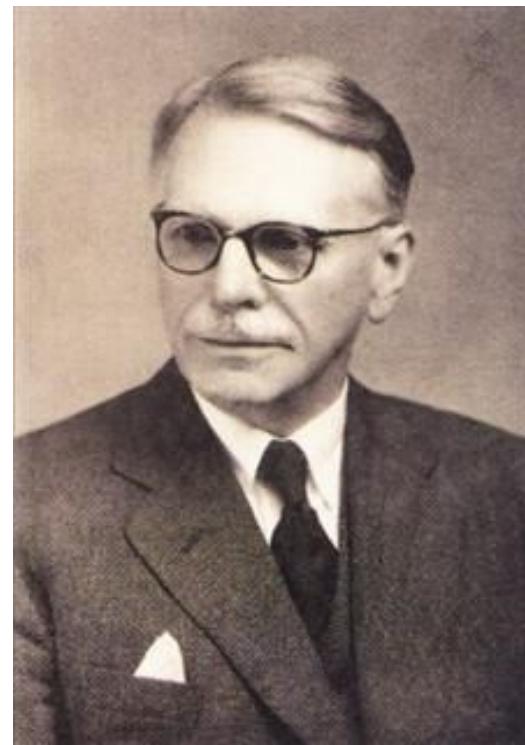
# Corpora: Wikipedia

Beverly Johnson (born October 13, 1952) is an [American|"United States"] [**model**"Model (person)"], [**actress**"Actress"], [**singer**"Singer"], and [**businesswoman**"Businesswoman"].

# How to determine the sense?

*You shall know a word by the  
company it keeps.*

John Rupert Firth, 1957



# 1. Lesk

With which sense **signature** does your **context** overlap the most?

```
function SIMPLIFIED LESK(word, sentence) returns best sense of word
```

```
best-sense  $\leftarrow$  most frequent sense for word
```

```
max-overlap  $\leftarrow$  0
```

```
context  $\leftarrow$  set of words in sentence
```

```
for each sense in senses of word do
```

```
signature  $\leftarrow$  set of words in the gloss and examples of sense
```

```
overlap  $\leftarrow$  COMPUTEOVERLAP(signature, context)
```

```
if overlap  $>$  max-overlap then
```

```
    max-overlap  $\leftarrow$  overlap
```

```
    best-sense  $\leftarrow$  sense
```

```
end
```

```
return(best-sense)
```

# Lesk

Simon works at an industrial plant as an engineer.

- S: (n) **plant**, works, industrial plant (buildings for carrying on industrial labor) "*they built a large plant to manufacture automobiles*"
- S: (n) **plant**, flora, plant life ((botany) a living organism lacking the power of locomotion)
- S: (n) **plant** (an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience)
- S: (n) **plant** (something planted secretly for discovery by another) "*the police used a plant to trick the thieves*"; "*he claimed that the evidence against him was a plant*"

# Lesk

How to find context words?

- filter functional words
- take lemmas
- for *signature* of each sense, use
  - examples
  - definitions
  - related terms
  - synonyms, hyponyms, hypernyms, holonyms, meronyms...
  - sentences from corpora, etc.

# Lesk

How to compute overlap?

- number of overlapping words
- weighed by the number of occurrences
- weighed by  $-\log(P(w))$
- weighed by IDF score:  $\log(C(\text{docs}) / C(\text{docs with word } i))$
- ...

# Important linguistic hypothesis

One sense per discourse!

*I bought a **plant** yesterday and put it in my small tank with some inch long baby cichlids. Lost 3 fish over night i never lose fish. i dont see any nibbles on the **plant** though.. any advice?*



# Results

Pros:

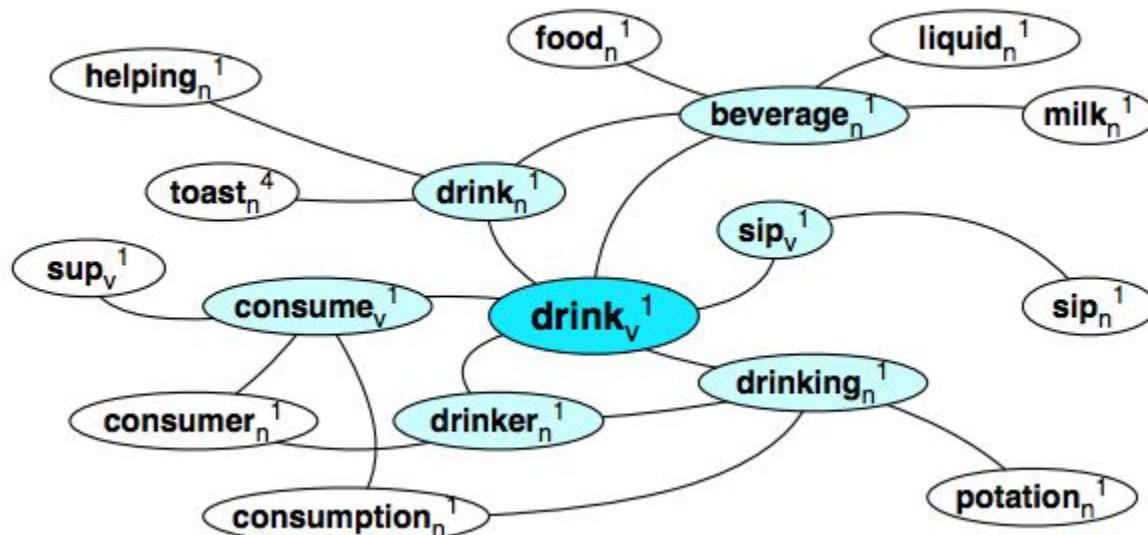
- good for partially annotating corpora
  - can be continued in a semi-supervised fashion
- links to the KB can be preserved
- unreasonably effective: up to ~0.7% prec and ~0.7% recall

Cons:

- some senses are poorly covered
- mapping (e.g., WordNet to Wikipedia) is a tricky task

## 2. Graph-Based

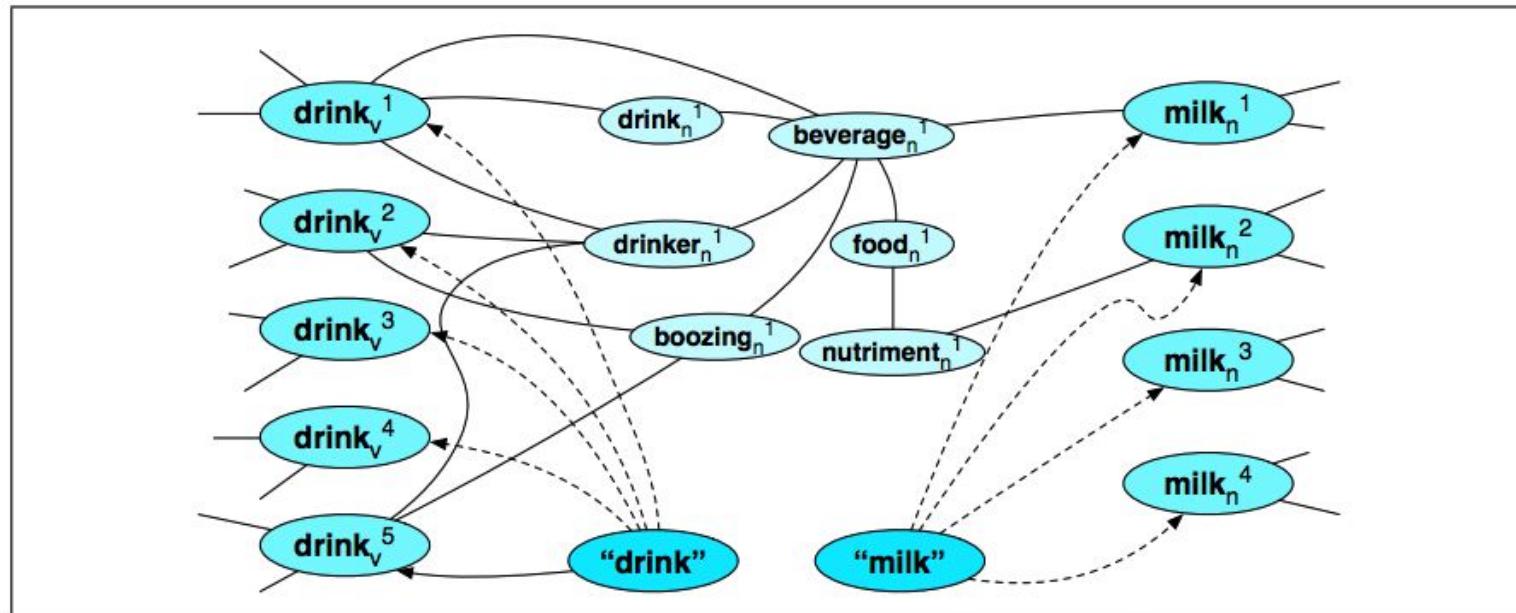
Which sense is the closest to context words?



Example from Navigli and Lapata (2010)

# Graph-Based

Which sense of the context word to choose?



Example from Navigli and Lapata (2010)

# Graph-Based

*Simon works at an industrial **plant** as an engineer.*

```
>>> plant_1 = wn.synset('plant.n.01')
>>> plant_1.definition()
u'buildings for carrying on industrial labor'

>>> plant_2 = wn.synset('plant.n.02')
>>> plant_2.definition()
u'(botany) a living organism lacking the power of locomotion'

>>> engineer = wn.synset('engineer.n.01')
```

# Graph-Based

*Simon works at an industrial **plant** as an engineer.*

```
>>> plant_1 = wn.synset('plant.n.01')
>>> plant_1.definition()
u'buildings for carrying on industrial labor'

>>> plant_2 = wn.synset('plant.n.02')
>>> plant_2.definition()
u'(botany) a living organism lacking the power of locomotion'

>>> engineer = wn.synset('engineer.n.01')
>>> plant_1.path_similarity(engineer)
0.1111111111111111
>>> plant_2.path_similarity(engineer)
0.25
```



# Graph-Based

Input the two lexical items [?](#)

plant#n#1

Input type:  [?](#)

engineer#n#1

Input type:  [?](#)

Alignment-based disambiguation?  Yes  No [?](#)

The similarity of the two items is: 0.182 [?](#)

unrelated (0)  (1) synonymous

Input the two lexical items [?](#)

plant#n#2

Input type:  [?](#)

engineer#n#1

Input type:  [?](#)

Alignment-based disambiguation?  Yes  No [?](#)

The similarity of the two items is: 0.052 [?](#)

unrelated (0)  (1) synonymous

# Babelfy

I need to buy a big plant for my mom .

**need**

Have need of



**buy**

Obtain by purchase;  
acquire by means of a  
financial transaction



**flora**

(botany) a living  
organism lacking the  
power of locomotion



**mommy**

Informal terms for a  
mother

# Babelfy algorithm

- *[done once]* create semantic signatures for BabelNet concepts
  - assign weights to edges; higher weights in more densely connected areas

# Babelfy algorithm

- [done once] create semantic signatures for BabelNet concepts
  - assign weights to edges; higher weights in more densely connected areas
- extract all linkable fragments from the text
- list possible meanings for the fragments
- create a graph-based semantic interpretation of the whole text by linking the possible meanings of the fragments

# Babelfy algorithm

- *[done once]* create semantic signatures for BabelNet concepts
  - assign weights to edges; higher weights in more densely connected areas
- extract all linkable fragments from the text
- list possible meanings for the fragments
- create a graph-based semantic interpretation of the whole text by linking the possible meanings of the fragments
- extract a dense subgraph of this representation
  - connect meanings if they are in each other's signature
- select the best candidate meaning for each fragment

# Babelfy

Simon

works at

a

plant

as an

engineer

.



**Herb Simon**

United States economist and psychologist who pioneered in the development of cognitive science (1916-2001)

**work on**

To exert effort in order to do, make, or perform something



**work**

Exert oneself by doing mental or physical work for a purpose or out of necessity



**industrial plant**

Buildings for carrying on industrial labor



**Engineer**

An engineer is a professional practitioner of engineering, concerned with applying scientific knowledge, mathematics, and ingenuity to develop solutions for technical, societal

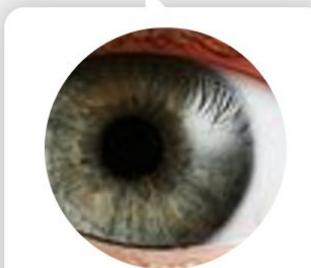
# Babelfy

The teacher and the pupils entered the classroom.



teacher

A person whose occupation is teaching



pupil

The contractile aperture in the center of the iris of the eye; resembles a large black dot

enroll

Register formally as a participant or member



classroom

A room in a school where lessons take place

# Babelfy

у дівчини гарна коса .

girl  
A friendly informal reference to a grown woman



scythe  
An edge tool for cutting grass; has a long handle that must be held with both hands and a curved blade that moves parallel to the ground

# Babelfy

У дівчини гарна коса .

girl

A friendly informal reference to a grown woman



scythe

An edge tool for cutting grass; has a long handle that must be held with both hands and a curved blade that moves parallel to the ground

У дівчини розплетена коса .

girl

A friendly informal reference to a grown woman



scythe

An edge tool for cutting grass; has a long handle that must be held with both hands and a curved blade that moves parallel to the ground

# Babelfy

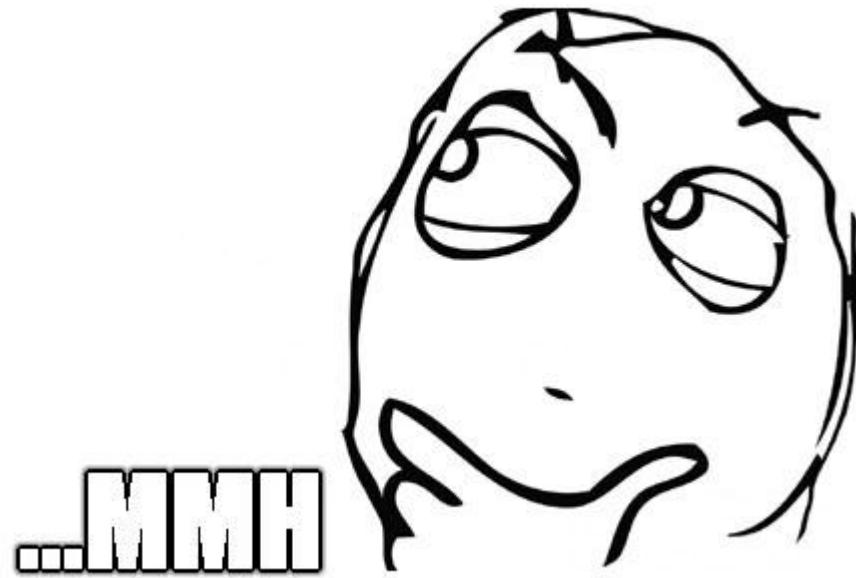
Дівчина      плете      косу .

girl  
A young woman

plait  
A hairdo formed by braiding or twisting the hair

# What else can we do?

- build a classifier
- build sense embeddings



# 3. Classification

Features:

- collocations
- bag of words

Containing:

- word, lemma, part of speech, dependencies

Classes:

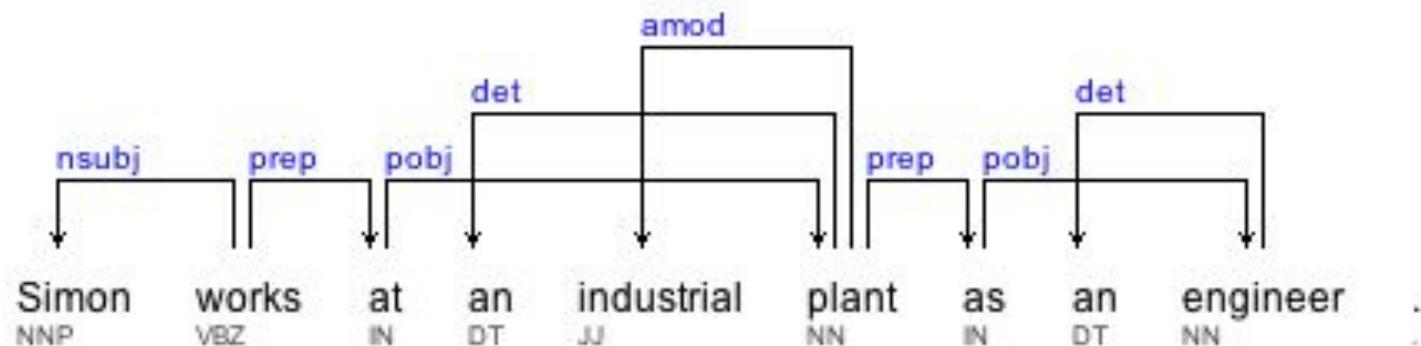
- senses from a sense inventory

# Collocations

*Simon works at an industrial **plant.n.1** as an engineer.*

**Ngrams:** [industrial plant, plant as, industrial plant as,...]

**Syngrams:** [works:prep\_at:plant, work:prep:as, plant:amod:industrial,...]



# Bag of words

Idea

- use a predefined set of context words for each word
- (this can add noise, though)

*Simon works at an industrial **plant** as an engineer.*

**plant:** [soil, assembly, root, industrial, contraband, agent, work...]  
[0, 0, 0, 1, 0, 0, 1...]

# Results

Pros:

- good for disambiguating specific words
- links to the KB can be preserved

Cons:

- not very successful for all-words task; more successful for disambiguating among larger senses
- needs a lot of data to generalize

# OpenAI: Type-based Neural Entity Disambiguation



The man saw a **Jaguar** speed on the highway.

Jaguar Cars 🚗 0.70

jaguar 🐾 0.12



The prey saw the **jaguar** cross the jungle.

Jaguar Cars 🚗 0.03

jaguar 🐾 0.89

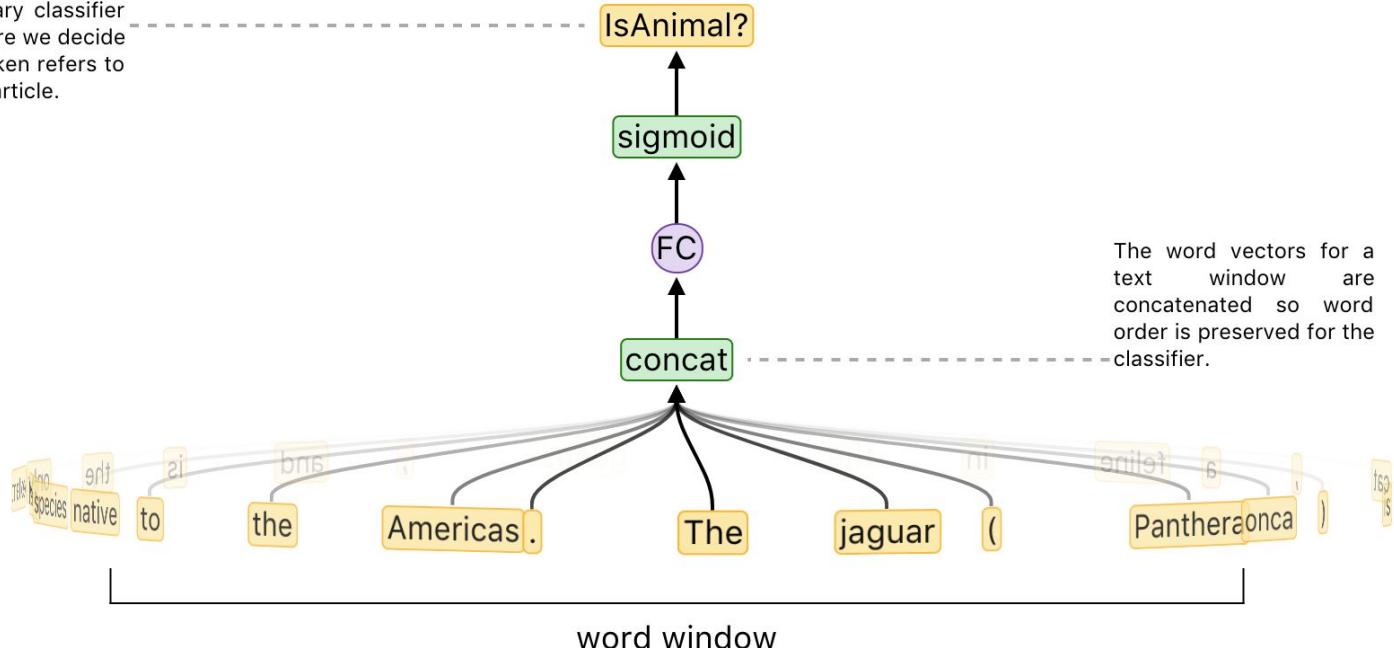
Example from blog.openai.com

# OpenAI: Type-based Neural Entity Disambiguation

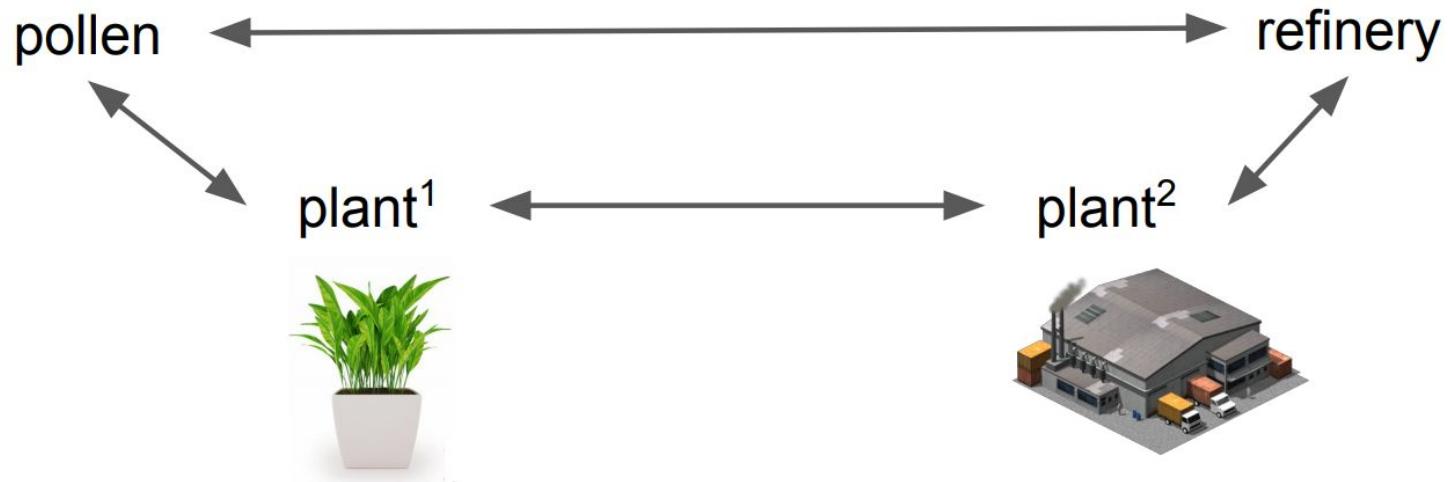
- Extract entities from Wikipedia
- Extract the set of categories for each entity from Wikipedia
- Pick a list of ~100 categories to be your “type” system
- Generate train data: word contexts mapped to 100-dimensional vectors
- Train: 400M tokens, bidirectional LSTM, F1 - 0.91

# OpenAI: Type-based Neural Entity Disambiguation

For each possible type  $a$  a different binary classifier is trained: here we decide whether a token refers to an "Animal" article.



# 4. Sense Embeddings



# Sense Representations

*Simon works at an industrial **plant.n.1** as an engineer.*

*I need to buy a big **plant.n.2** for my mom. She likes gardening!*

# SensEmbed vectors

- use *BabelNet* as a sense inventory
- use a dump of the *English Wikipedia* for corpus
- do disambiguation with *Babelfy*
- train *word2vec CBOW* to obtain sense embeddings  
(window = 5, size = 400)

# SensEmbed vectors

$bank_1^n$ (geographical)	$bank_2^n$ (financial)	$number_4^n$ (phone)	$number_3^n$ (acting)
upstream <sup>r</sup> <sub>1</sub>	commercial_bank <sup>n</sup> <sub>1</sub>	calls <sup>n</sup> <sub>1</sub>	appearing <sup>v</sup> <sub>6</sub>
downstream <sup>r</sup> <sub>1</sub>	financial_institution <sup>n</sup> <sub>1</sub>	dialled <sup>v</sup> <sub>1</sub>	minor_roles <sup>n</sup> <sub>1</sub>
runs <sup>v</sup> <sub>6</sub>	national_bank <sup>n</sup> <sub>1</sub>	operator <sup>n</sup> <sub>20</sub>	stage_production <sup>n</sup> <sub>1</sub>
confluence <sup>n</sup> <sub>1</sub>	trust_company <sup>n</sup> <sub>1</sub>	telephone_network <sup>n</sup> <sub>1</sub>	supporting_roles <sup>n</sup> <sub>1</sub>
river <sup>n</sup> <sub>1</sub>	savings_bank <sup>n</sup> <sub>1</sub>	telephony <sup>n</sup> <sub>1</sub>	leading_roles <sup>n</sup> <sub>1</sub>
stream <sup>n</sup> <sub>1</sub>	banking <sup>n</sup> <sub>1</sub>	subscriber <sup>n</sup> <sub>2</sub>	stage_shows <sup>n</sup> <sub>1</sub>

# Quality

Measure	Dataset					
	RG-65	WS-Sim	WS-Rel	YP-130	MEN	Average
Pilehvar et al. (2013)	0.868	0.677	0.457	0.710	0.690	0.677
Zesch et al. (2008)	0.820	—	—	0.710	—	—
Collobert and Weston (2008)	0.480	0.610	0.380	—	0.570	—
Word2vec (Baroni et al., 2014)	0.840	0.800	0.700	—	0.800	—
GloVe	0.769	0.666	0.559	0.577	0.763	0.737
ESA	0.749	—	—	—	—	—
PMI-SVD	0.738	0.659	0.523	0.337	0.726	0.695
Word2vec	0.732	0.707	0.476	0.343	0.665	0.644
SENSEMBED <sub>closest</sub>	<b>0.894</b>	0.756	0.645	<b>0.734</b>	0.779	0.769
SENSEMBED <sub>weighted</sub>	0.871	<b>0.812</b>	<b>0.703</b>	0.639	<b>0.805</b>	<b>0.794</b>

Table 3: Spearman correlation performance on five word similarity and relatedness datasets.

# Nasari vectors

Bank (financial institution)

English	French	Spanish
bank	banque	banco
banking	bancaire	bancario
deposit	crédit	banca
credit	financier	financiero
money	postal	préstamo
loan	client	entidad
commercial_bank	dépôt	déposito
central_bank	billet	crédito

Bank (geography)

English	French	Spanish
river	eau	banco
stream	castor	limnología
bank	berge	ecología
riparian	canal	barrera
creek	barrage	estuarios
flow	zone	isla
water	perchlorate	interés
watershed	humide	laguna

# Results

Pros:

- good for word similarity tasks
- easy to train
- links to the KB can be preserved

Cons:

- some senses are poorly covered
- needs a lot of data to generalize

# 5. Word sense induction

Idea:

- for each word occurrence, compute a context vector
- cluster these context vectors
- compute the sense vector in each cluster
- map sense vectors to senses

The number of clusters should be predefined. Or not.

2.

# Semantic role labeling

# Who did what to whom

- Causer (agent/force)
- Instrument
- Result
- Patient
- Theme
- Source, path, goal/recipient, location
- Experiencer
- Stimulus
- Beneficiary
- Time, manner, reason...

The police officer detained the suspect at the scene of the crime

Agent

Predicate

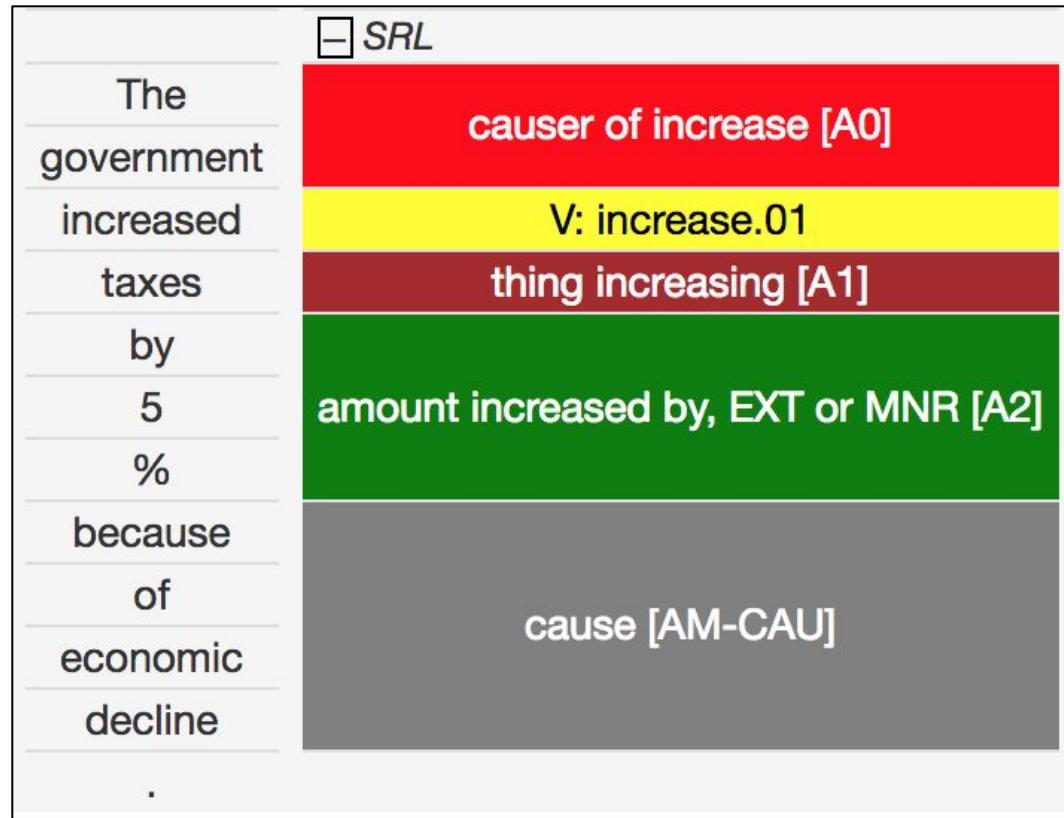
Theme

Location

# PropBank

- *increase.01 “go up incrementally”*
  - Arg0: *causer of increase*
  - Arg1: *thing increasing*
  - Arg2: *amount increased by*
  - Arg3: *start point*
  - Arg4: *end point*
- ***The government increased taxes by 5%.***
- ***Taxes increased.***

# PropBank



# FrameNet

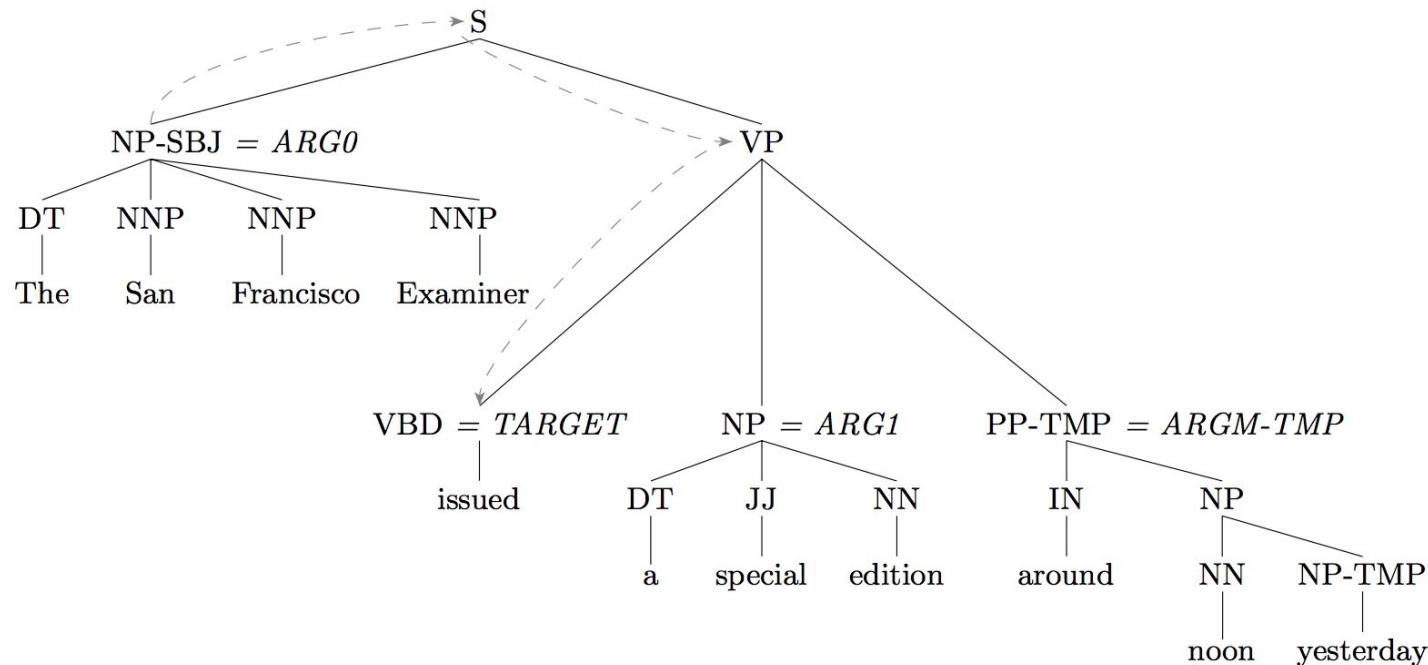
- Abandonment:
  - *abandon, abandoned, abandonment, leave, forget*
- An **Agent** leaves behind a **Theme** effectively rendering it no longer within their control or as one's property...
- examples:
  - **Carolyn** abandoned **her car** and jumped on a bus.
  - Abandonment of **a child** is considered to be a serious crime in many jurisdictions.
  - Perhaps **he** left **the key** in the ignition.

# Basic SRL algorithm

- parse the sentence
- find all predicates
- for each predicate in the sentence
  - for each node in the parse tree
    - assign the semantic role (if any) for the predicate

*Logistic regression, SVM, Perceptron, CRF, etc.*

# Basic SRL example



What features would you use?

# Basic SRL features

- predicate
  - lemma, POS, active/passive voice, etc.
  - syntactic frame
    - e.g., VP → VBD NP PP
- node
  - label
  - headword lemma, headword POS, etc.
  - linear position
- path from the constituent to the predicate
  - e.g., NP↑S↓VP↓VBD
- more: ngrams, dependencies, etc.

# Basic SRL notes

- Evaluation:
  - precision, recall, f-measure
- Also possible two steps:
  - identification
    - to prune unlikely constituents
    - to avoid nested constituents
  - classification

3.

# Semantic parsing

# Combinatory Categorial Grammar

Language can be presented as a logical structure:

*Which state borders with Kansas?*

$\Rightarrow \lambda x. \text{state}(x) \wedge \text{border}(x, \text{Kansas})$

*Texas borders with Kansas.*

$\Rightarrow \text{border}(\text{Texas}, \text{Kansas})$

# Combinatory Categorial Grammar

Language can be presented as a logical structure:

*Which state borders with Kansas?*

$$\Rightarrow \lambda x. \text{state}(x) \wedge \text{border}(x, \text{Kansas})$$

*Texas borders with Kansas.*

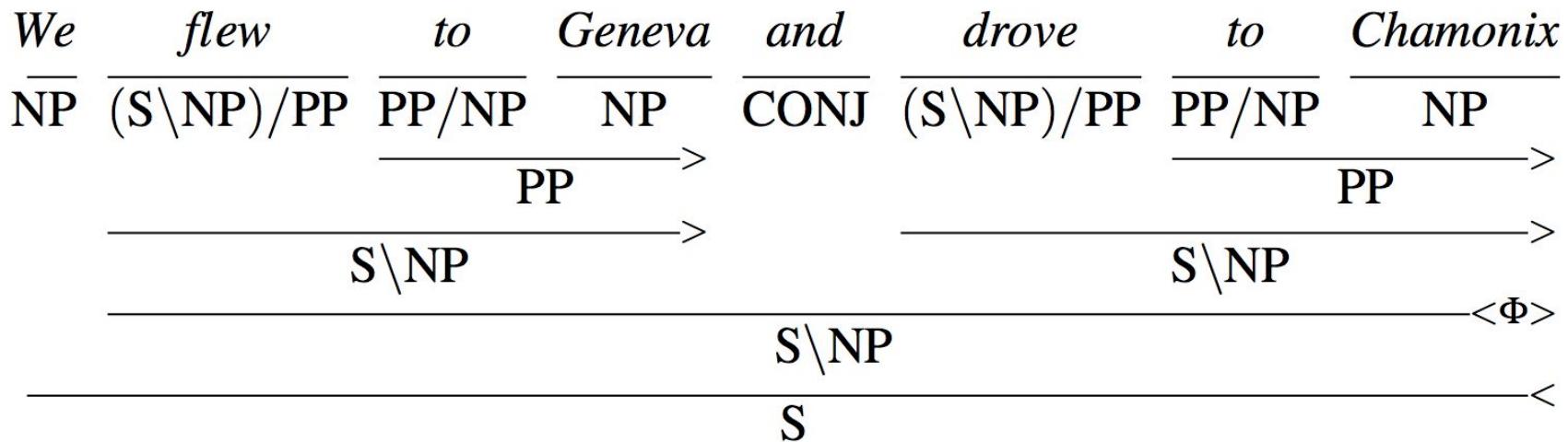
$$\Rightarrow \text{border}(\text{Texas}, \text{Kansas})$$

Texas	borders	Kansas
NP texas	(S\NP) / NP $\lambda x. \lambda y. \text{borders}(y, x)$	NP kansas
<hr/>		
$S \setminus NP$ $\lambda y. \text{borders}(y, kansas)$		
<hr/>		
S $\text{borders}(\text{texas}, \text{kansas})$		

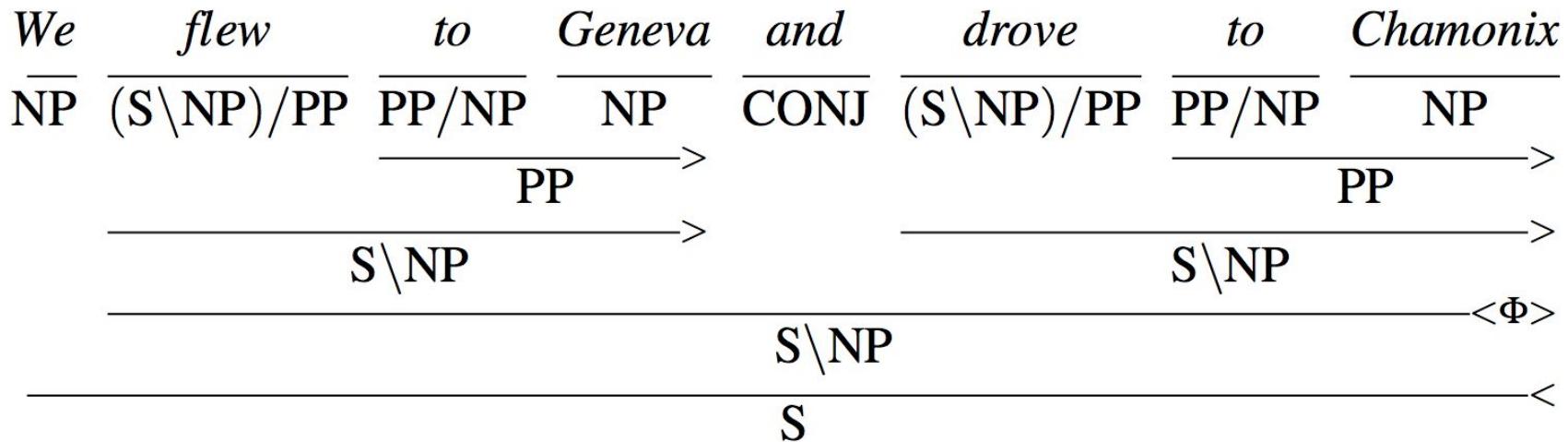
# Combinatory Categorial Grammar

- Categories
  - atomic elements:  $NP$ ,  $ADJ$ ,  $S$
  - functions:  $(S \setminus NP)/ADJ$
- Lexicon
  - Tom:  $NP$
  - cancel:  $(S \setminus NP)/NP$
  - give:  $((S \setminus NP)/NP)/NP$
  - be:  $(S \setminus NP)/ADJ$
- Rules
  - $X/Y \ Y \Rightarrow X$
  - $Y \ X \setminus Y \Rightarrow X$
  - $X \ CONJ \ X \Rightarrow X$

# Combinatory Categorial Grammar

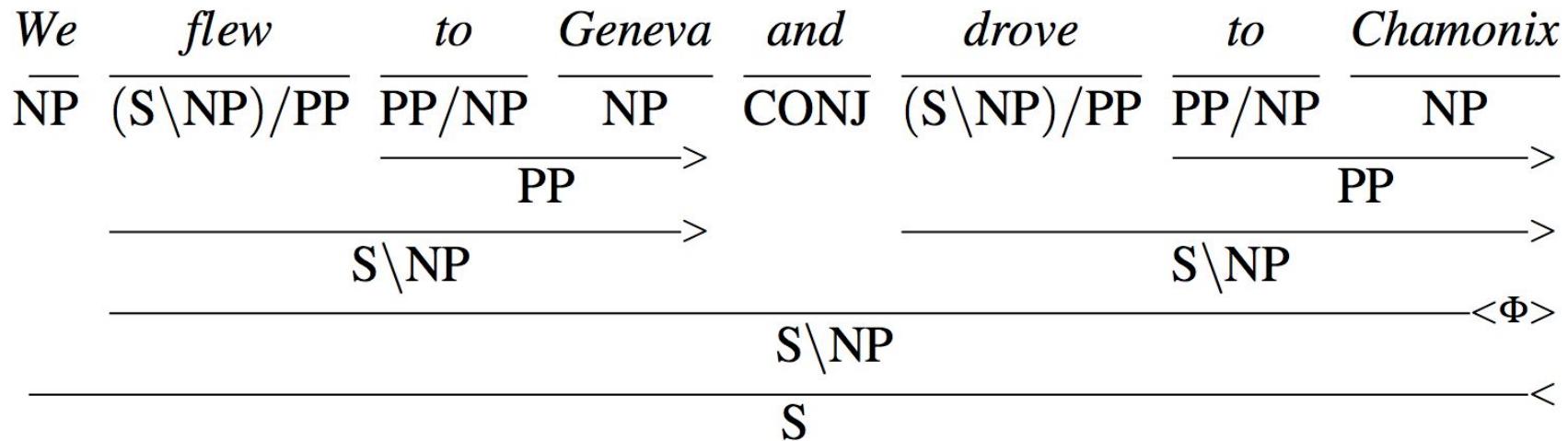


# Combinatory Categorial Grammar



*flew(we, to(Geneva))*  $\wedge$  *drove(we, to(Chamonix))*

# Combinatory Categorial Grammar

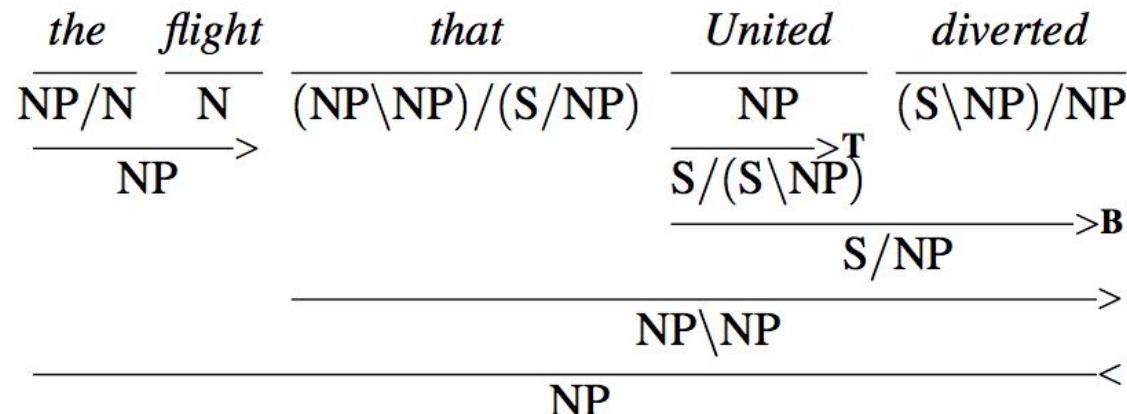


More nice examples from NLTK:

[https://github.com/nltk/nltk/blob/develop/nltk/test/ccg\\_semantics.doctest](https://github.com/nltk/nltk/blob/develop/nltk/test/ccg_semantics.doctest)

# Combinatory Categorial Grammar

- More rules
  - backward/forward composition
    - $X/Y \ Y/Z \Rightarrow X/Z$
    - $Y\backslash Z \ X\backslash Y \Rightarrow X\backslash Z$
  - type raising
    - $X \Rightarrow T/(T\backslash X)$
    - $X \Rightarrow T\backslash(T/X)$



# CCG Bank

# CCG parsing

- tagging
  - a sequence labeling task (425 categories)
- A\* algorithm
  - a modification of the probabilistic CKY
  - instead of filling in the whole chart, rerank elements by probability and proceed with the most probable ones
- Evaluation
  - Parseval
  - Recovery of word-word dependencies

# Abstract Meaning Representation

*The boy desires the girl to believe him.*

*The boy wants the girl to believe him.*

*The boy desires to be believed by the girl.*

*The boy has a desire to be believed by the girl.*

*The boy's desire is for the girl to believe him.*

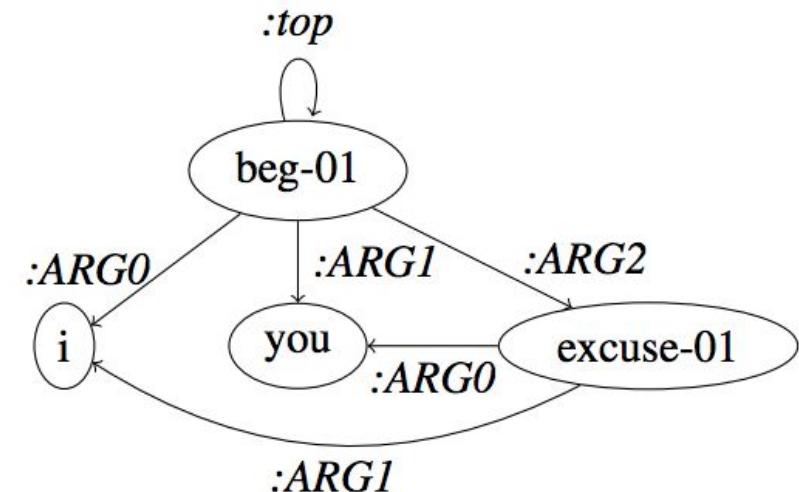
*The boy is desirous of the girl believing him.*

```
(w / desire-01
  :ARG0 (b / boy)
  :ARG1 (b2 / believe-01
    :ARG0 (g / girl)
    :ARG1 b))
```

# Abstract Meaning Representation

Each sentence is a **rooted, directed, acyclic** graph with labels on edges (relations) and leaves (concepts). Functional words are omitted.

```
(b / beg-01
  :ARG0 (i / i
  :ARG1 (y / you)
  :ARG2 (e / excuse-01
    :ARG0 y
    :ARG1 i))
```



# AMR: more logical than syntax

*his boat*

*his own boat*

(b / boat  
:poss (h / he))

*The boy must not go.*

*It is obligatory that the boy not go.*

(o / obligate-01  
:ARG2 (g / go-02  
:ARG0 (b / boy)  
:polarity -))

# AMR: more logical than syntax

*He described the mission as a failure.*

*As he described it, the mission was a failure.*

*His description of the mission: failure.*

```
(d / describe-01
  :ARG0 (h / he)
  :ARG1 (m / mission)
  :ARG2 (f / fail-01))
```

# AMR: more logical than syntax

*Which state borders with Kansas?*

```
(b / border-01
  :ARG0 (s / state
           :name (n / name :op1 (a / amr-unknown)))
  :ARG1 (s2 / state
           :name (n2 / name :op1 "Kansas")))
```

*Does Texas border with Kansas?*

```
(b / border-01
  :ARG0 (s / state
           :name (n / name :op1 "Texas"))
  :ARG1 (s2 / state
           :name (n2 / name :op1 "Kansas"))
  :polarity (a / amr-unknown))
```

# AMR: more logical than syntax

*About 14,000 people fled their homes at the weekend after a local tsunami warning was issued, the UN said on its Web site.*

```
(s / say-01
  :ARG0 (g / organization
          :name (n / name :op1 "UN"))
  :ARG1 (f / flee-01
    :ARG0 (p / person
      :quant (a / about :op1 14000))
    :ARG1 (h / home :poss p)
    :time (w / weekend)
    :time (a2 / after
      :op1 (w2 / warn-01
        :ARG1 (t / tsunami)
        :location (l / local))))
  :medium (s2 / site
    :poss g
    :mod (w3 / web)))
```

# AMR parsing

- Data - AMR Bank
  - 39,260 sentences of newswire, discussion forum and other web logs, television transcripts
  - plus, *Little Prince* :)
  - need alignment
- Parsing algorithms:
  - graph-based
  - transition-based

# AMR alignment

JAMR Aligner (2014):

- align graph fragments to spans of words
- uses a set of handcrafted rules in a specific order:
  - dates
  - named entities
  - predicate
  - single concepts
  - fuzzy single concepts
  - etc.
- Precision: 92%, recall: 89%

# AMR alignment

JAMR Aligner (2014)

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
  :ARG0 (o / organization
    :ARG0 (n / name
      :op1 "IAEA"))
  :ARG1 (t2 / thing
    :ARG1-of (p / propose-01
      :ARG0 (c / country
        :name (n2 / name
          :op1 "North"
          :op2 "Korea")))))
  :time (d / date-entity
    :month 11))
```

# AMR alignment

ISI Aligner (2014):

- convert the graph to a linear string
- remove stop words from the sentence and auxiliary tags from the graph
- stem to the first 4 letters both graph and sentence
- use word alignment model

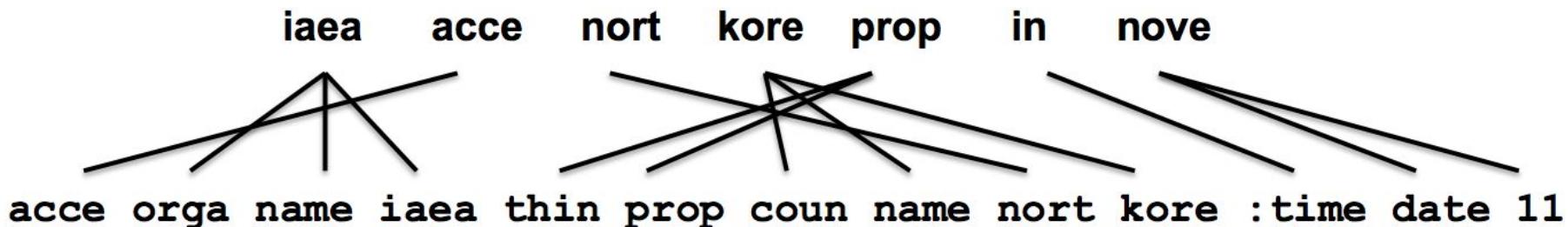
**IAEA accepted North Korea proposal in November**

```
accept organization name IAEA thing propose-01
country name North Korea :time date-entity 11
```

# AMR alignment

ISI Aligner (2014):

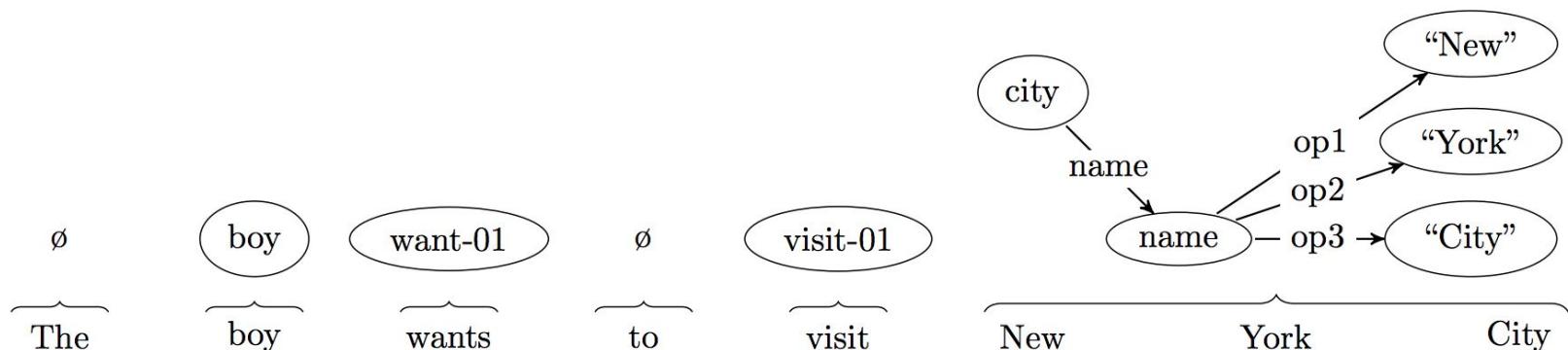
- convert the graph to a linear string
- remove stop words from the sentence and auxiliary tags from the graph
- stem to the first 4 letters both graph and sentence
- use word alignment model



# AMR: graph-based parsing

Algorithm

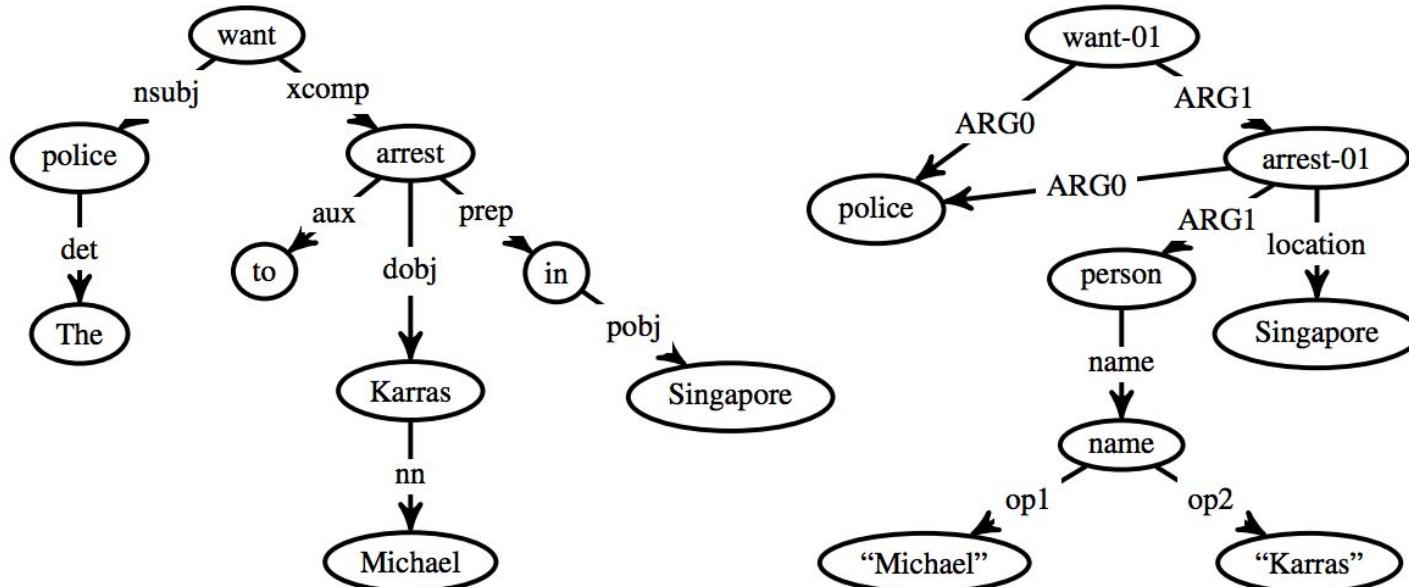
- identify concepts
  - *a sequence labeling task*
- identify relations (maximum spanning connected subgraph)
  - *similar to graph-based dependency parsing*



Example from Flanigan et al (2014)

# AMR: transition-based parsing

Idea: convert dependency tree into AMR graph.



Example from Wang et al (2015)

# AMR: transition-based parsing

Setup:

- $\sigma$  - a buffer that stores indices of the nodes which have not been processed;  $\sigma_0$  - top of the buffer
- $\beta$  - a buffer  $[\beta_0, \beta_1, \dots, \beta_j]$  and each element  $\beta_i$  of  $\beta$  indicates the edge  $(\sigma_0, \beta_i)$  which has not been processed in the partial graph;  $\beta_0$  - top of the buffer
- $G$  - span graph that stores the partial parses

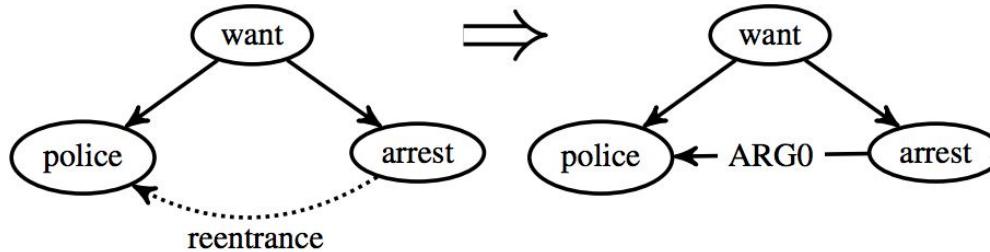
# AMR: transition-based parsing

Actions:

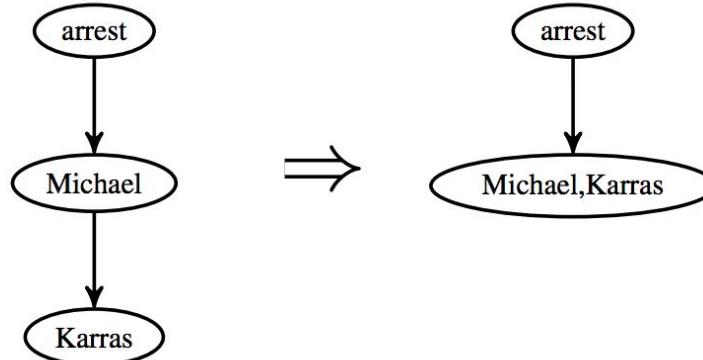
- NEXT-EDGE- $l_r$  - attach edge  $(\sigma_0, \beta_0)$  and move to next word
- SWAP- $l_r$  - swap nodes and attach with edge
- REATTACH $_k-l_r$  - delete edge and reattach to already processed node
- REPLACE-HEAD - replace node with another node
- REENTRANCE $_k-l_r$  - attach edge to already processed node
- MERGE - merge two nodes
- NEXT-NODE- $l_c$  - label with concept and move to next word
- DELETE-NODE - delete a word

# AMR: transition-based parsing

REENTRANCE<sub>k-l<sub>r</sub></sub>



MERGE



# AMR evaluation: smatch

```
(w / want-01
  :ARG0 (b / boy)
  :ARG1 (b2 / believe-01
    :ARG0 (g / girl)
    :ARG1 b))
```

instance(w, want-01)	/* w is an instance of wanting */
instance(b, boy)	/* b is an instance of boy */
instance(b2, believe-01)	/* b2 is an instance of believing */
instance(g, girl)	/* g is an instance of girl */
ARG0(w, b)	/* b is the wanter in w */
ARG1(w, b2)	/* b2 is the wantee in w */
ARG0(b2, g)	/* g is the believer in b2 */
ARG1(b2, b)	/* b is the believee in b2 */

# Minimal Recursive Semantics

*The jungle lion was chasing a small giraffe.*

```
 $\langle h_1, e_3,$ 
 $h_4:\text{the\_q}(x_6, h_7, h_5),$ 
 $h_8:\text{compound}(e_{10}, x_6, x_9),$ 
 $h_{11}:\text{udef\_q}(x_9, h_{12}, h_{13}),$ 
 $h_{14}:\text{jungle\_n\_1}(x_9),$ 
 $h_8:\text{lion\_n\_1}(x_6),$ 
 $h_2:\text{chase\_v\_1}(e_3, x_6, x_{15}),$ 
 $h_{16}:\text{a\_q}(x_{15}, h_{18}, h_{17}),$ 
 $h_{19}:\text{small\_a\_1}(e_{20}, x_{15}),$ 
 $h_{19}:\text{giraffe\_n\_1}(x_{15})$ 
 $\{ h_1 =_q h_2, h_7 =_q h_8, h_{12} =_q h_{14}, h_{18} =_q h_{19} \} \rangle$ 
```

4.

**Textual entailment**

# Textual entailment

We say that text  $T$  entails hypothesis  $H$  (i.e.,  $T \Rightarrow H$ ) if the meaning of  $H$  can be inferred from the meaning of  $T$ , as would typically be interpreted by people.

- Positive TE (text entails hypothesis):
  - *A girl swings high in the air.*
  - *A girl is on a swing.*
- Negative TE (text contradicts hypothesis):
  - *A girl swings high in the air.*
  - *A girl is laying in the pool.*
- Non-TE (text does not entail nor contradict):
  - *A girl swings high in the air.*
  - *A girl is looking for her mother.*

# Textual entailment

**Text:** The purchase of Houston-based LexCorp by BMI for \$2Bn prompted widespread sell-offs by traders as they sought to minimize exposure. LexCorp had been an employee-owned concern since 2008.

**Hyp 1:** BMI acquired an American company.

**Hyp 2:** BMI bought employee-owned LexCorp for \$3.4Bn.

**Hyp 3:** BMI is an employee-owned concern.

# Textual entailment

**Text:** The purchase of Houston-based LexCorp by BMI for \$2Bn prompted widespread sell-offs by traders as they sought to minimize exposure. LexCorp had been an employee-owned concern since 2008.

**Hyp 1:** BMI acquired an American company.

**Hyp 2:** BMI bought employee-owned LexCorp for \$3.4Bn.

**Hyp 3:** BMI is an employee-owned concern.

- NER: *LexCorp, BMI, \$2Bn*
- WordNet: *purchase (n.) => purchase (v.), acquire, buy*
- Knowledge base: *Houston-based => American*
- Syntactic analysis + semantic role labeling / semantic parsing

# Textual entailment

**Text:** The purchase of Houston-based LexCorp by BMI for \$2Bn prompted widespread sell-offs by traders as they sought to minimize exposure. LexCorp had been an employee-owned concern since 2008.

**Hyp 1:** BMI acquired an American company.

**Hyp 2:** BMI bought employee-owned LexCorp for \$3.4Bn.

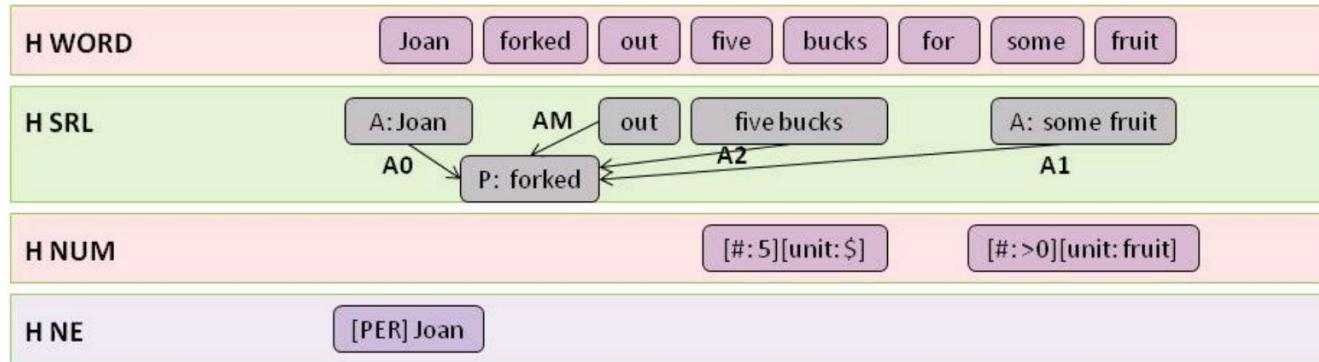
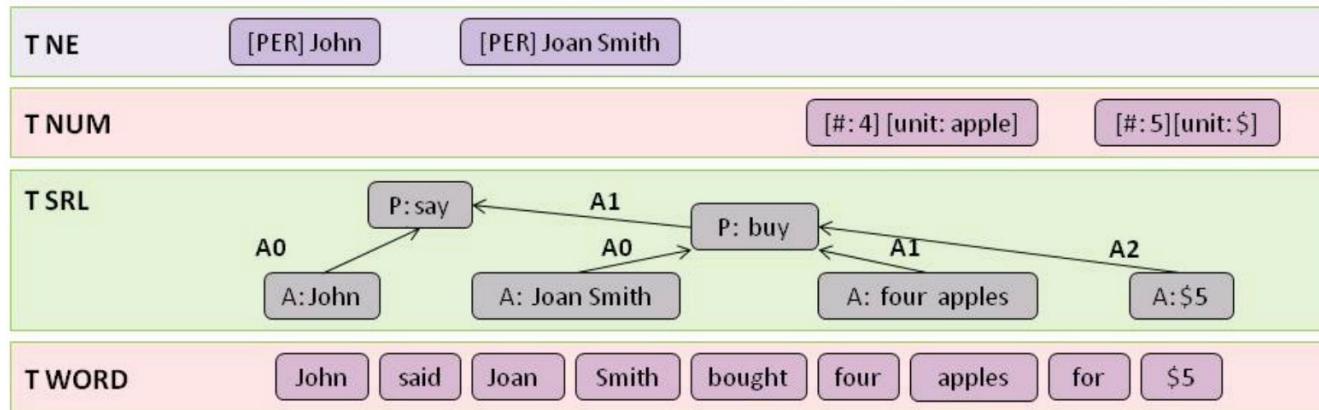
**Hyp 3:** BMI is an employee-owned concern.

- Normalization of quantities:  $\$2Bn \Rightarrow 2,000,000,000$
- Fuzzy match of NE: *BMI*  $\Rightarrow$  *Business Machine Inc.*
- Coreference resolution
- Scope of negation

# Textual entailment pipeline

- Text and hypothesis preprocessing
- Align syntactic/semantic representations of  $T$  and  $H$ 
  - only a small portion of  $T$  is relevant to  $H$
- Compare the constituents
  - match of NEs and numerical measures
  - lexico-semantic relations (synonyms, antonyms, hypernyms, meronyms, entailment)
  - word similarity
  - noun-verb relations (“*is a winner*” => “*won*”)
  - idioms (“*kick the bucket*” => “*die*”)
  - paraphrases (“*has been unable to*” => “*could not*”)
  - syntactic mapping (active => passive)

# Textual entailment pipeline



# Textual entailment pipeline

- Inference
  - compute the overlap  
or
  - train a classifier

# Textual entailment: TEASE

- Extract entailment paraphrases from web using a dependency parser
  - both lexical and grammatical
- Apply iteratively to transform  $T$  to  $H$

$X \text{ write } Y$	$X \text{ who write } Y$	$X \text{ produce } Y$
	$X \text{ publish } Y$	$X \text{ pen } Y$
	$X \text{ compose } Y$	$X \text{ create } Y$
	$\text{read } Y \text{ by } X$	$X \text{ s } Y$
	$Y \text{ attributed to } X$	$X \text{ complete } Y$
	$\text{perform } Y \text{ by } X$	$X \text{ book of } Y$
	$X \text{ writer of } Y$	$X \text{ say in } Y$
	$\text{selected } Y \text{ of } X$	$X \text{ work include } Y$

# References

- Neelakantan et al. (2014), [Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space](#)
- Iacobacci et al. (2015), SENSEMBED: [Learning Sense Embeddings for Word and Relational Similarity](#)
- Camacho-Collados et al. (2016), [Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities](#)
- Navigli and Lapata (2010), [An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation](#)
- Athiwaratkun and Wilson (2017), [Multimodal Word Distributions](#)
- Nematzadeh, Meylan and Griffiths (2017), [The Unreasonable Effectiveness of Counting Words Near Other Words](#)
- Raiman and Raiman (2018), [Discovering Types for Entity Disambiguation](#)

# References

- [Babelfy: Entity Linking meets Word Sense Disambiguation](#), Moro, Raganato, and Navigli (2015)
- [ConceptNet 5.5: An Open Multilingual Graph of General Knowledge](#), Robert Speer (2016)
- [Speech and Language Processing](#), Chapters 13 and 22, Jurafsky and Martin (2017)
- [CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank](#), Hockenmaier and Steedman (2007)
- CCG Software: <http://groups.inf.ed.ac.uk/ccg/software.html>
- AMR: <https://amr.isi.edu/index.html>
- [A Transition-based Algorithm for AMR Parsing](#), Wang et al. (2015)
- [A Discriminative Graph-Based Parser for AMR](#), Flanigan et al. (2014)
- [Textual Entailment](#), Alicia Ageno (2014)
- [Recognizing Textual Entailment](#), Mark Sammons (2012)