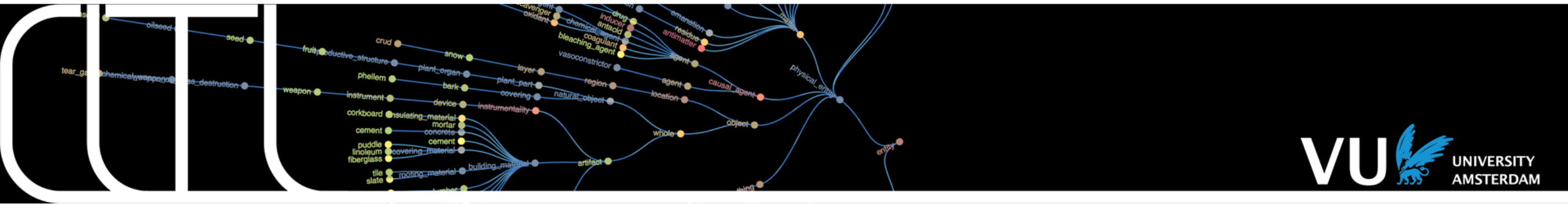
# Text Mining CBS

## Lecture 1: Linguistics and Natural Language Processing

Piek Vossen




VU UNIVERSITY AMSTERDAM

# Logistiek: https://github.com/cltl/text-mining-ba

- 9:30 - 10:30 lecture
- pauze
- 11:00 - 12:00 lecture
- lunch
- 13:00 - 14:00 lab
- pauze
- 14:30 - 15:30 lab

- eigen laptop met voldoende schijfruimte
- downloaden en installeren:
  - Anaconda (Python 3.7): https://anaconda.org
  - https://github.com/cltl/text-mining-ba
- Jupyter notebooks: https://jupyter.org
- NLTK:
- - http://www.nltk.org/book
- SpaCy
- - https://spacy.io
- - https://spacy.io/usage/models
- Skitlearn:
- - https://scikit-learn.org/stable/

There is no such thing as simple text!

# There is no such thing as simple text

twitterer

The new US president has said he believes that vaccines are harmful and has repeatedly and erroneously suggested that they cause autism

# There is no such thing as simple text

The new US president has said he believes that vaccines are harmful and has repeatedly and erroneously suggested that they cause autism

twitterer

The new US president

said he believes

vaccines are harmful

repeatedly and erroneously suggested

they cause autism

# There is no such thing as simple text

The new US president has said he believes that vaccines are harmful and has repeatedly and erroneously suggested that they cause autism

twitterer

The new US president

said he believes

vaccines are harmful

repeatedly and erroneously suggested

they cause autism

Entity phrase detection

Entity Linking

provenance attribution

coreference

semantic parsing

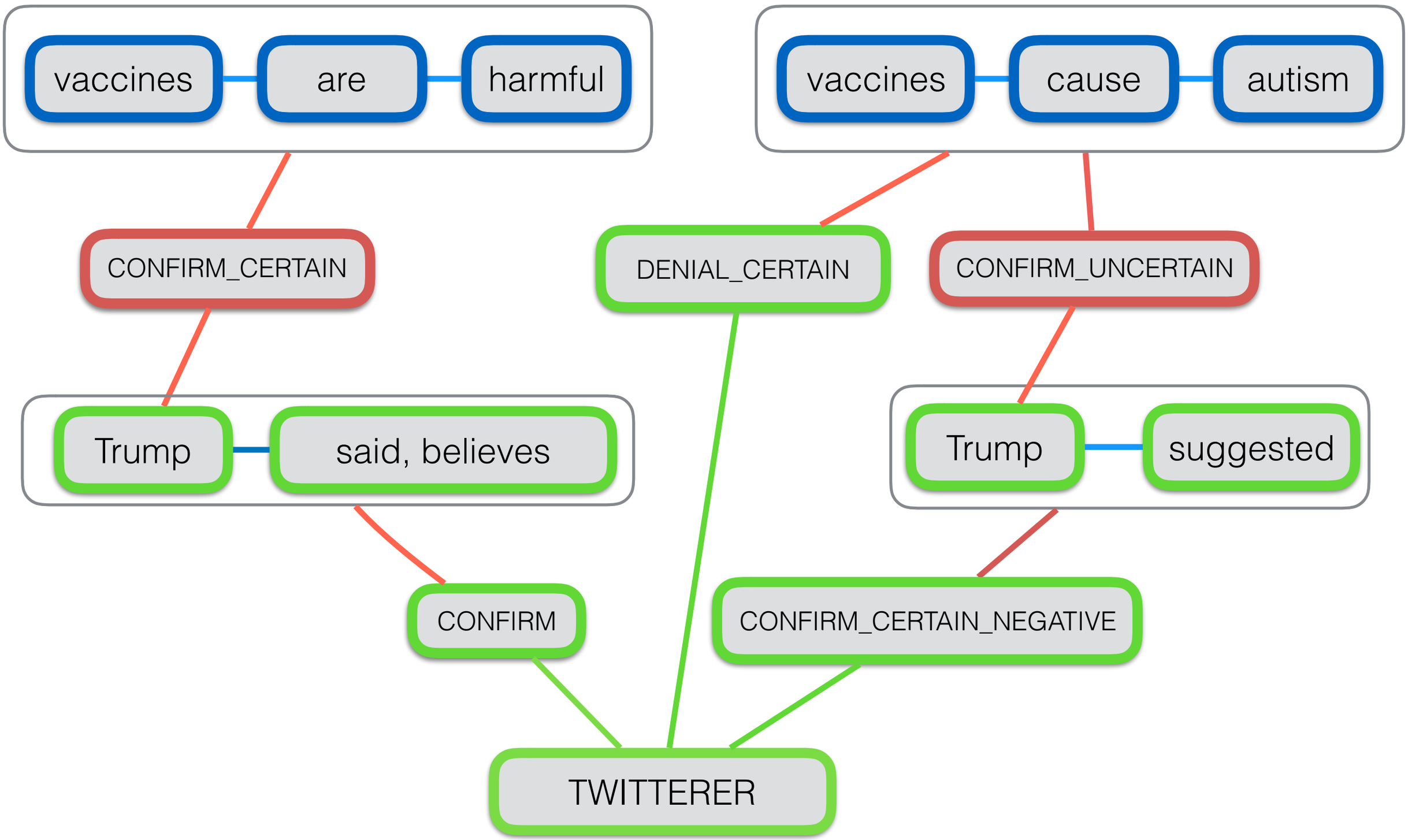
provenance attribution

judgement

semantic parsing

coreference

# Perspective model



# Terminology

- **Computational linguistics**: algorithms that model language data, e.g. *similarity, information value, sequence probabilities*

- **Natural Language processing** (NLP): engineering to address aspects of natural language, e.g. *tokenisation, lemmatisation, compound splitting, syntactic parsing, entity detection, sentiment analysis*, etc.

- **NLP Toolkits**: software packages and resources that provide and/or combine collections of NLP modules

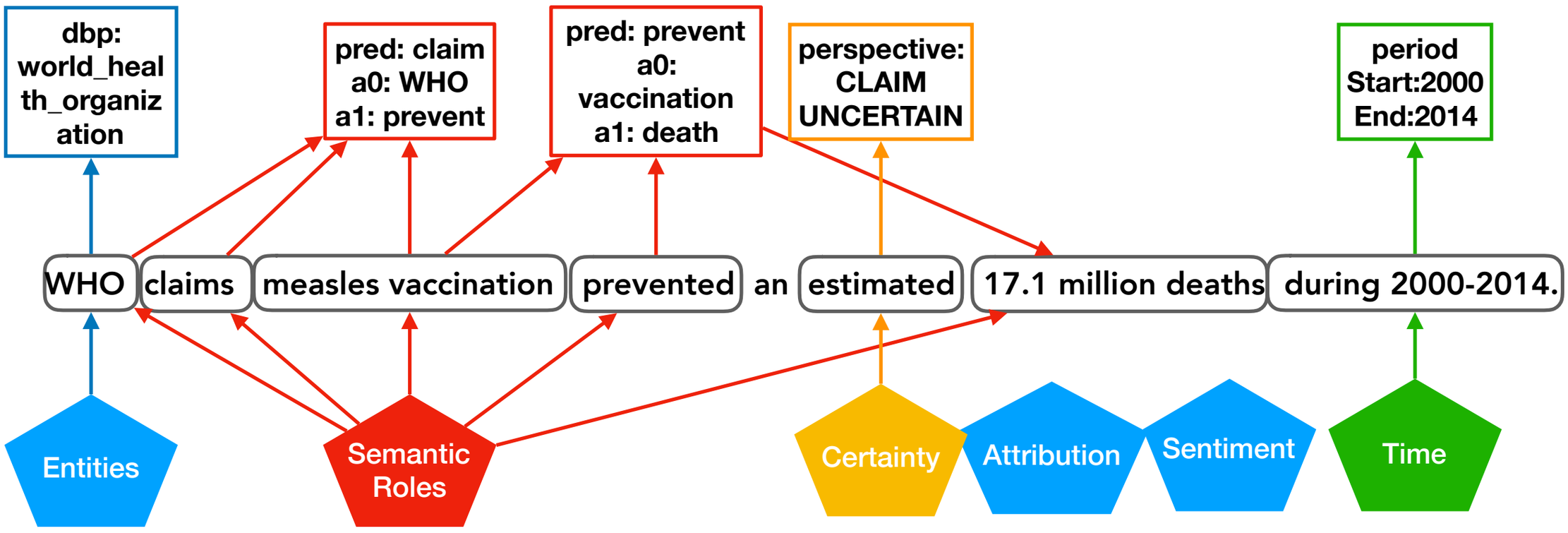
- **Language applications**: machine translation, summarisation, chat bots, text mining

- **Text mining**: from unstructured text to structured data (information or knowledge)

# NLP Toolkits (Python)

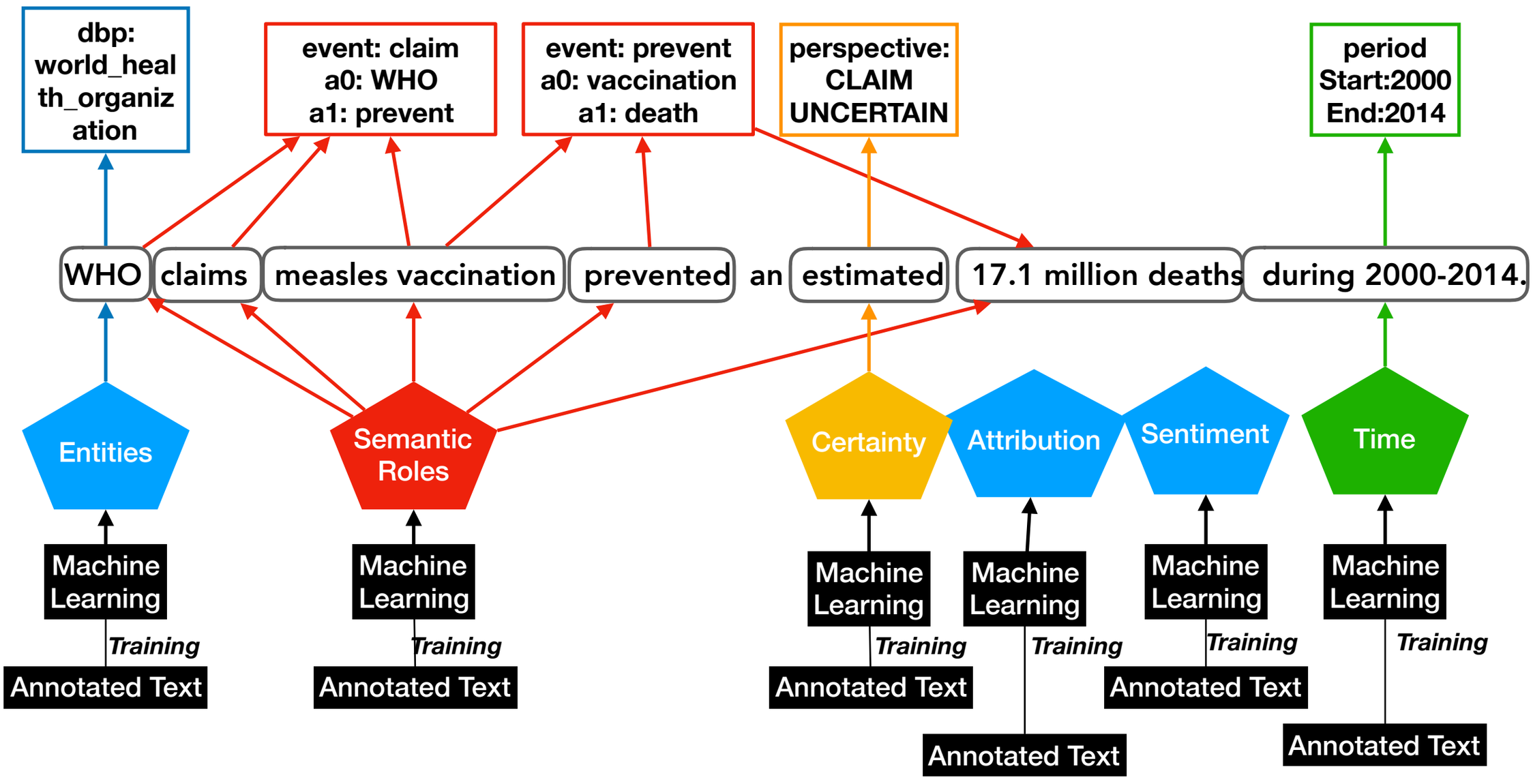
- NLTK: http://www.nltk.org
- spaCy: https://spacy.io
- AllenNLP: https://allennlp.org
- Solutions for: tokenisation, PoS tagging, chunking, parsing, named entity detection in various languages
- **Utility packages:**
- Scikit-learn, Gensim, Pandas, etc…..

# Text Mining


dbp: world_health_organization
pred: claim a0: WHO a1: prevent
pred: prevent a0: vaccination a1: death
perspective: CLAIM UNCERTAIN
period Start:2000 End:2014
WHO
claims
measles vaccination
prevented
an
estimated
17.1 million deaths
during 2000-2014.
Entities
Semantic Roles
Certainty
Attribution
Sentiment
Time

# Text Mining

WHO claims measles vaccination prevented an estimated 17.1 million deaths during 2000-2014.



# CBS Text Mining cursus

| Week | Theorie | Lab sessies | |
|---|---|---|---|
| 1 | Piek Vossen<br>• Kennis van taal<br>• Regels, lexica, machine learning, data annotatie<br>• Sentiment, opinies, en emoties | Piek Vossen<br>• Python toolkits NLTK en SpaCy<br>• Sentiment: lexicon en statistiek<br>• Evaluatie: accuracy, recall, precision, f1 | Basis |
| 2 | Piek Vossen<br>• Van Tekst naar Feature representatie<br>• Entiteiten<br>• Eigenschappen van entiteiten | Piek Vossen<br>• Entiteiten detectie en classificatie<br>• Entiteiten disambiguering/linking<br>• Extractie van eigenschappen | |
| 3 | Antske Fokkens<br>• Word embeddings<br>• Neurale netwerken<br>• Blackbox versus Clearbox | Pia Sommerauer & Antske Fokkens<br>• Similarity and Relatedness<br>• Embeddings in Machine Learning | Verdieping |

# Overview of lecture

- Part I: Linguistics and Natural Language Processing (NLP)
    - morphology
    - syntax
    - semantics
- Part II: NLP Pipelines

# Part I: Linguistics

- Language and Structure

- Language and Meaning

# Linguistics

| Subdiscipline | Medium or unit | Natural language module |
|---|---|---|
| phonetics, phonology | sounds | Automatic Speech Recognition |
| morphology | words, word formation | Part-Of-Speech taggers, lemmatisers, compound splitters |
| syntax | sentences, grammatical structure and function | Syntactic parsers, chunkers |
| semantics | meaning | Semantic parsers |
| pragmatics | language use in context | Context and domain models |
| methods | introspection, behavorism, neuro-cognitive models, empirical (experimental & stochastic), mathematical models | |
| resources | Lexicons, grammars, data collections and annotations, data models, annotations | |

# Morphology

- Study of form and structure of words
- Words are composed of ***morphemes***
- ***Morpheme*** is the smallest meaning-bearing unit:
    - e.g. *talked* contains two morphemes: *talk* (activity) and -*ed* (past)

# Types of morphemes

- **Free Morphemes**: occur independently, e.g. *boy*, *sing*

- **Bound Morphemes**: attached to another morpheme, and cannot be used independently, e.g.
    - English [NUMBER pl] -*s* → boys,
    - Dutch [NUMBER pl] -*s* → appels, [NUMBER pl] -*en* → appelen

- **Affixes**: **prefixes** (e.g. *ge*lopen), **infixes** (e.g. burgemeester*s*post, **suffixes** (e.g. loop*je*)

# Some other basic terms

- **Root or Base**: an un-analysable morpheme, expressing the basic lexical content of a word. Also defined as ‘what is left of a complex form when all affixes are stripped’.
  - “kinderen” —> “kind”, “kinderopvang”
- **Stem**: consists of at least a root. It can contain (a) derivational affix(es). “aardigste” —> “aardig” vs. “aard”
- **Lemma**: an entry in a dictionary,
  - single form for nouns (“stemmetjes” -> “stem”),
  - infinitive form for verbs (“stemde” —> “stemmen”)

# Part of Speech (PoS)

- Words have **part-of-speech** (**PoS**), which specifies the typical phrase structures in which they can be the head (see later)
- **Open class** (open to word formation and neologisms):
  - Noun (N, *boat*), Verb (V, *float*), Adjective (A, *large, fast*), Adverb (*very, largely*)
  - new words invented every day and other words are forgotten, e.g. "***belubberen***";
  - millions of open class words if we include specialised language (chemistry, medicine, product names)
- **Closed class** (you can not invent a new closed class word):
  - Pronoun (PRN, *he, him, this, who*), Preposition (P, *in, at, from, in front of*), etc.
  - relatively fixed, slowly change over generations; small set of less than a hundred words

# Word modification

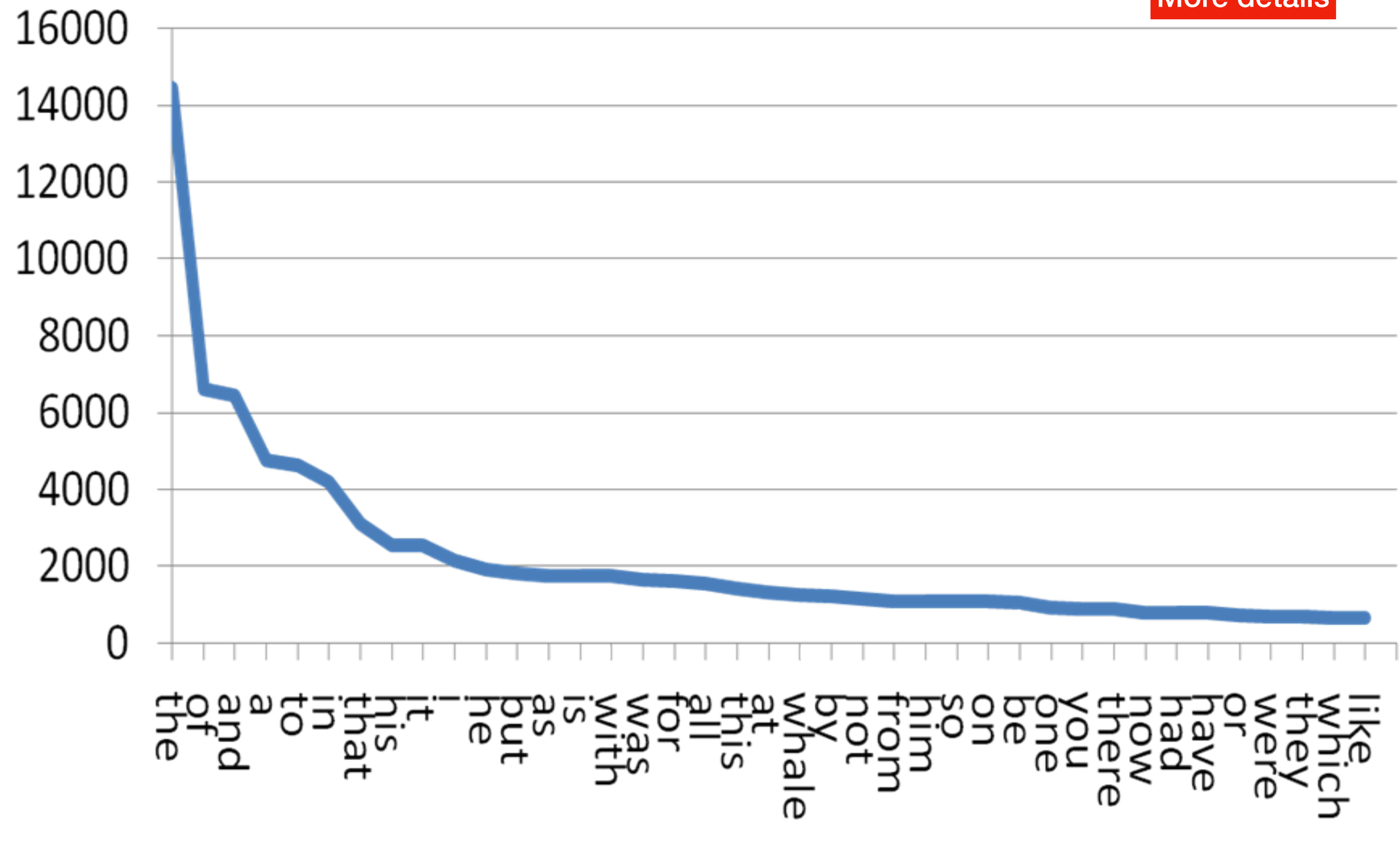
- Given a root, base or stem derive different forms
  - **Inflection**, expresses syntactic properties such as person (1,2,3), number (singular, plural), gender, tense, e.g. “book*s*”, “*her* hair”, “walk*ed*”.
  - **Derivation**: changes semantic and grammatical properties, e.g. “*in*appli*cable”* A, “head” N —> “*be*head” V
    - **Compounding**: “beach head”, “tarwemeel” (oat flour), “kindermeel” (child flour)
    - **Combinations**: aircraft-carriers = ((air+craft)+(carry - er)) (not: air +craftcarrier …)
- Word formation is very productive, our lexicon is potentially infinite:
  - the number of unseen compounds detected in German & Dutch newspapers  grows linearly with the number of newspapers over time
  - the names for new chemical compounds and proteins grow rapidly every year
  - new products (e.g. apps) launched every year

# Forms in language

- A language has:
    - 11-112 phonemes (sound units)
    - 4,000-10,000 morphemes (word units)
    - 50,000 common words, millions of words including terminologies
    - An infinite number of sentences
- But we use small proportion of these forms very frequently even though we recognise and understand most of those

# Power law distribution of word frequency

More details



More details

# Zipfian distributions

$f(w_i)=f(W_1)/r_i(w_i)$

- The frequency of a word in a ranked list is the equal to the frequency of the most frequent word divided by the rank 100, 50, 30, 25, 20, 16, 14, etc….

- Most frequent words also tend to be short and have many different meanings

**George Kingsley Zipf**
**1902 - 1950**

# Lexicon of forms

- Lists all common base forms (a hundred of thousand in a standard dictionary) with:
  - their part-of-speech
  - inflectional paradigm
  - typical (conventional) derived forms
- Inflectional paradigms and derivational morphemes

# Morphology in Computational Linguistics

- Analysing **complex words**, defining their component parts: anti+dis+establish+ment+arian+ism

- Analysis of **grammatical information**, encoded in words:

    - *sings*

        - **Part-of-speech** = VERB

        - **Inflectional information** = [PERSON 3, NUMBER singular,TENSE present

- Obtaining the **stem** or **root**: to reduce size of the data, to find the word in the lexicon

    - Dutch “stemmen” (voice or vote) —> “stem” noun, “stemmen” verb

    - Reduction of lexicon size (English 2:1, Dutch/German 5:1, Finnish/Turkish >200:1) (Crysmann 2006)

# Part-of-Speech tagging

- Task: assign the Part-of-Speech category (e.g. noun, verb, adjective) to every token and add the lemma

- Tagset: no consensus (there are at least 50 different tag sets):
    - http://universaldependencies.org/u/pos/

- PoS tagging is done using machine learning
    - Hidden Markov Models, Decision Trees, SVM, Naive Bayes

- Main challenge:
    - Data sparseness for specific languages and domains

# PoS-tagging

- Assign morphosyntactic categories to words in a specific context:

| The | green | train | runs | down | that | track | . |
|---|---|---|---|---|---|---|---|
| Det | Adj/NN | NNS/ VBZ | NN/ VB | Prep/ Adv/ | SC/ Pron | NN/ VB | . |
| Det | Adj | NNS | VB | Prop | Pron | NN | . |

- Lexical and contextual constraints are used to identify the right tag

More details

# Markov model

- Markov models are used to predict sequences
- They assume that the next state in the sequence is dependent on the current state (only)
- See Section on PoS tagging for more in-depth workings: http://www.cse.unsw.edu.au/~billw/cs9414/notes/nlp/ambiguity/ambiguity-2009.html

# State-of-the-art in PoS-tagging

- Accuracy around ***95-97%*** for all tokens when training and testing on the same domain

- Remaining issues:
  - long distance dependencies
  - genuine ambiguities
  - annotation errors
  - unknown words, data sparseness

# PoS-tagging issues

- Better/richer models? (we would need even more data!)
- ***Coverage for other domains can drop to 75%***
- Morphologically rich languages need far larger training data sets (Finnish & Turkish need up to 10x more data than English)
- 95% sounds good but you don’t know which tokens are wrongly tagged
  - Relatively high proportion of sentences has at least one error
  - Errors propagate: wrong PoS may lead to wrong word sense, named entity, parse tree etc.

# Morphology tooling

- **NLTK**: https://www.nltk.org/book/ch05.html
  - Annotated corpora with part-of-speech tags
  - Dictionaries with word forms, their segmentation, part-of-speech text and inflectional information
    - various morphological parsers: stemming and PoS annotation
- Dutch morphological lexicon **e-Lex**: https://ivdnt.org/downloads/tstc-e-lex
- Some famous stemmers, lemmatisers and taggers:
  - **Porter** stemmer: https://tartarus.org/martin/PorterStemmer/
  - **Snowball**: https://snowballstem.org
  - **Treetagger**: https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

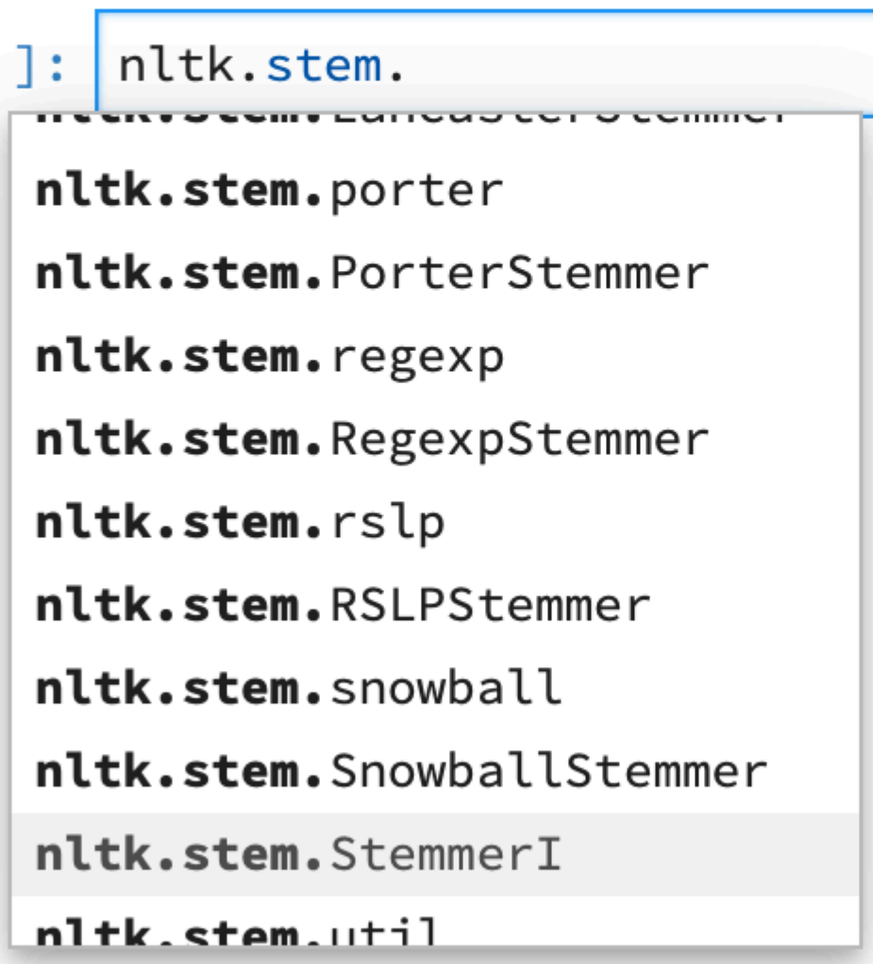
# Morphology tooling in NLTK

```
In [1]: import nltk

In [2]: from __future__ import print_function

In [3]: from nltk.stem import *

In [ ]: nltk.stem.
```



# Annotated texts

## NLTK: nltk_data/corpora/brown/ca11

- The/at Birds/nns-tl got/vbd five/cd hits/nns and/cc all/abn three/cd of/in their/pp$ runs/nns off/in Kunkel/np before/cs Hartman/np took/vbd over/rp in/in the/at top/nn of/in the/at fourth/od ./.

- Hartman/np ,/, purchased/vbn by/in the/at A's/nn from/in the/at Milwaukee/np Braves/nns-tl last/ap fall/nn ,/, allowed/vbd no/at hits/nns in/in his/pp$ scoreless/jj three-inning/jj appearance/nn ,/, and/cc merited/vbd the/at triumph/nn ./.

- Keegan/np ,/, a/at 6-foot-3-inch/jj 158-pounder/nn ,/, gave/vbd up/rp the/at Orioles'/nps$ last/ap two/cd safeties/nns over/in the/at final/jj three/cd frames/nns ,/, escaping/vbg a/at load/nn of/in trouble/nn in/in the/at ninth/od when/wrb the/at Birds/nns-tl threatened/vbd but/cc failed/vbd to/to tally/vb ./.

# Morphological lexicon

E-Lex: https://ivdnt.org/downloads/taalmaterialen/tstc-e-lex

97809\stem\(stem)[N]\\A0;C->N->S4;C->N->Z2\A0;S4;Z2\\295418\stem\N(soort,ev,basis,zijd,stan)\\V\stEm\stEm\stEm\'stEm\V\686\[DET:<de>][HD:<stem>]\\

97809\stem\(stem)[N]\\A0;C->N->S4;C->N->Z2\A0;S4;Z2\\295418\stem\N(soort,ev,basis,zijd,stan)\\V\stEm\stEm\stEm\'stEm\V\686\[DET:<de>][HD:<stem>][PC:[PP:[HD:<op>][OBJ1:NP]]]\\

97809\stem\(stem)[N]\\A0;C->N->S4;C->N->Z2\A0;S4;Z2\\295418\stem\N(soort,ev,basis,zijd,stan)\\V\stEm\stEm\stEm\'stEm\V\686\[DET:<de>][HD:<stem>][PC:[PP:[HD:<tegen>][OBJ1:NP]]]\\

97809\stem\(stem)[N]\\A0;C->N->S4;C->N->Z2\A0;S4;Z2\\295418\stem\N(soort,ev,basis,zijd,stan)\\V\stEm\stEm\stEm\'stEm\V\686\[DET:<de>][HD:<stem>][PC:[PP:[HD:<voor>][OBJ1:NP]]]\\

97809\stem\(stem)[N]\\A0;C->N->S4;C->N->Z2\A0;S4;Z2\\295418\stem\N(soort,ev,basis,zijd,stan)\\V\stEm\stEm\stEm\'stEm\V\686\[DET:<een>][HD:<stem>]\\

97809\stem\(stem)[N]\\A0;C->N->S4;C->N->Z2\A0;S4;Z2\\295418\stem\N(soort,ev,basis,zijd,stan)\\V\stEm\stEm\stEm\'stEm\V\686\[HD:<stem>]\\

…..

97810\stemmen\(stem)[V]\\\\\471636\gestemd\WW(vd,prenom,zonder)\B\C\x@stEmt\G@stEmt\x@stEmt\x@-'stEmt\V\0\\\

97810\stemmen\(stem)[V]\\\\\107427\gestemd\WW(vd,vrij,zonder)\B\V\x@stEmt\G@stEmt\x@stEmt\x@-'stEmt\V\108\[SU:NP][HD:<hebben>][VC:[HD:<gestemd>]]\\

97810\stemmen\(stem)[V]\\\\\107427\gestemd\WW(vd,vrij,zonder)\B\V\x@stEmt\G@stEmt\x@stEmt\x@-'stEmt\V\108\[SU:NP][HD:<hebben>][VC:[PPART:[HD:<gestemd>]]]\\

97810\stemmen\(stem)[V]\\\\\107427\gestemd\WW(vd,vrij,zonder)\B\V\x@stEmt\G@stEmt\x@stEmt\x@-'stEmt\V\108\[SU:NP][HD:<hebben>][VC:[PPART:[OBJ1:NP][HD:<gestemd>]]]\\

97810\stemmen\(stem)[V]\\\\\107427\gestemd\WW(vd,vrij,zonder)\B\V\x@stEmt\G@stEmt\x@stEmt\x@-'stEmt\V\108\[SU:NP][HD:<hebben>][VC:[PPART:[OBJ1:NP][PC:[PP:[HD:<tot>][OBJ1:INF]]][HD:<gestemd>]]]\\

97810\stemmen\(stem)[V]\\\\\107427\gestemd\WW(vd,vrij,zonder)\B\V\x@stEmt\G@stEmt\x@stEmt\x@-'stEmt\V\108\[SU:NP][HD:<hebben>][VC:[PPART:[OBJ1:NP][PC:[PP:[HD:<tot>][OBJ1:NP]]][HD:<gestemd>]]]\\

…..

# Multiword expressions

- Fixed Idioms
  - *An apple a day keeps the doctor away*
  - *kick the bucket, Raining cats and dogs*
- Less fixed idioms
  - *shooting from the hip*
- Slots
  - *X, let alone Y*
- Collocations
  - *running engine, strong coffee, count on, treat for*
- Selectional restrictions:
  - *essen/fressen, a glass of …, eat edibles, “poten” or “benen”*
  - *blow your nose, neus snuiten en niet blazen*

# Syntax

- We experience a sentences as a complete grammatical structure.
- We can freely combine words into **phrases** or **constituents** and we have a strong intuition about the grammaticality of these structures within a sentence.
- What is a **phrase** or **constituent**?
  - A phrase is a word or a group of words which functions as single unit within a grammatical hierarchy
  - A phrase is built around a **head** lexical item and has a certain syntactic behaviour
    - she ⇒ Noun Phrase or NP (the **head** is a pronoun)
    - a very beautiful morning ⇒ NP (the **head** is a Noun)
    - chases the cat ⇒ Verb Phrase or VP (**head** is a Verb)

# Syntactic elements

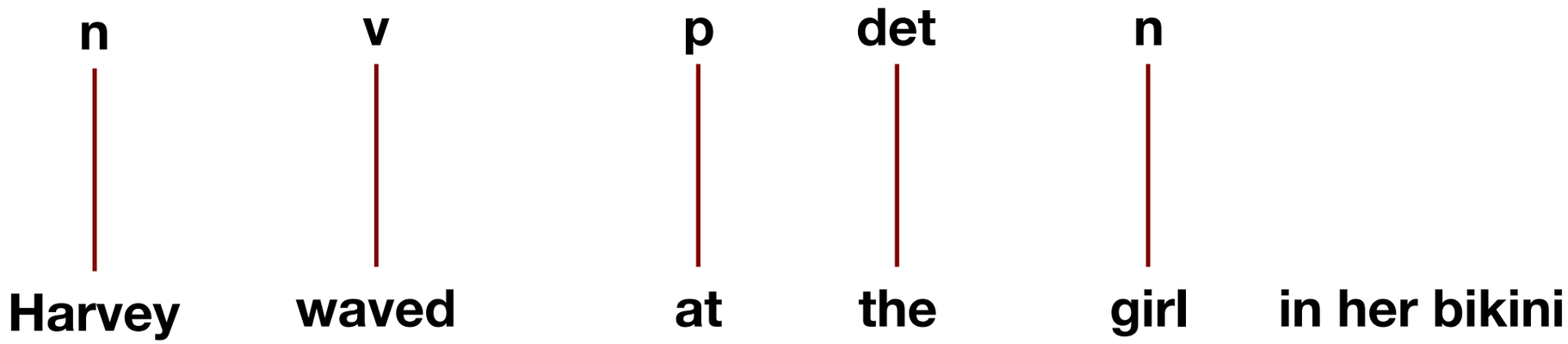
- Phrasal categories: Noun Phrase (NP)
    - Prepositional Phase (PP)
    - Verb Phrase (VP)
    - Adverbial Phrase (AdvP)
    - Adjectival Phrase (AP)
- Lexical categories: Noun (N)
    - Pronoun (Pr)
    - Adjective (A)
    - Adverb (Adv)
    - Verb (V)
    - Preposition (P)

# Syntax

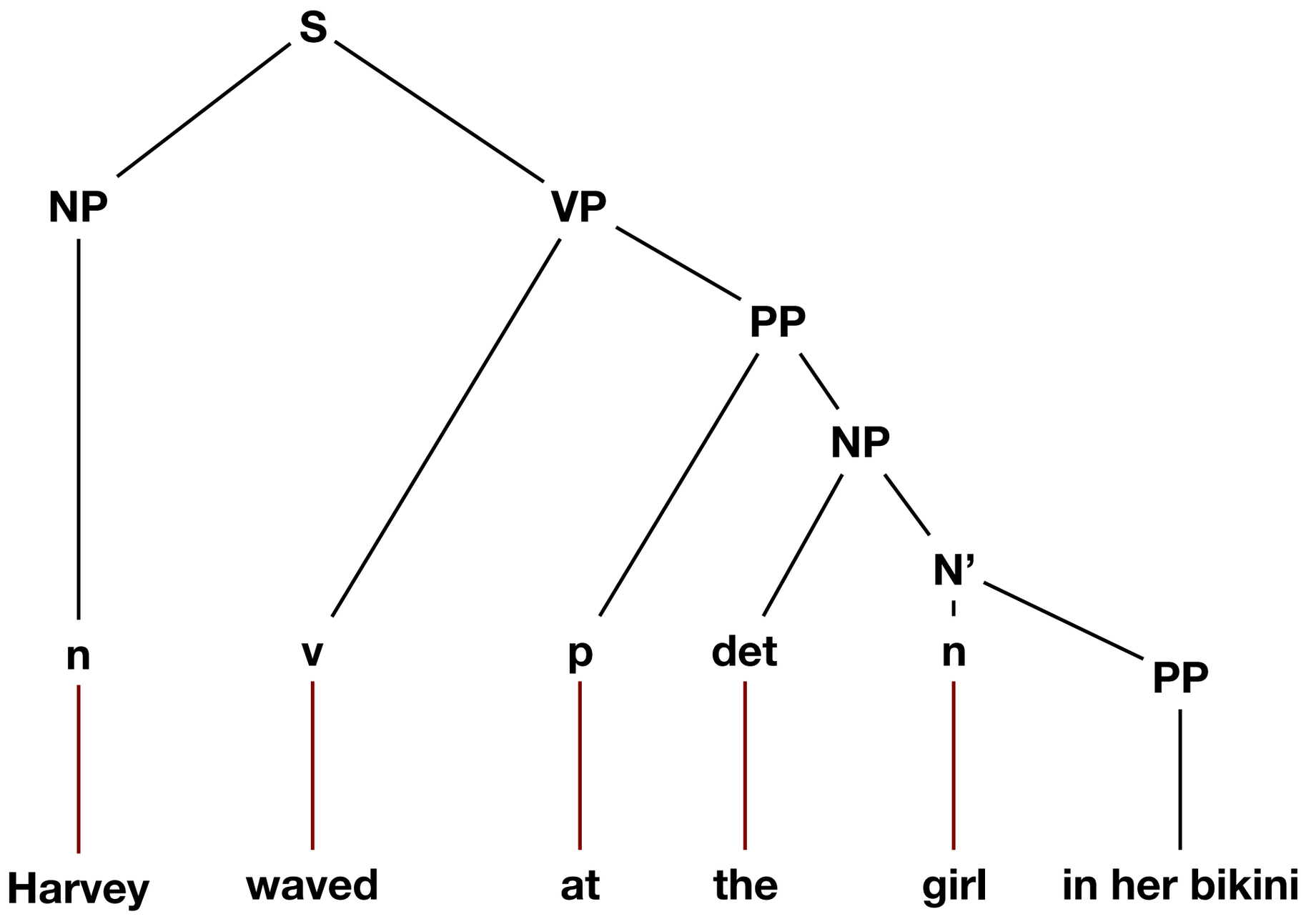
- Phrases can be nested hierarchically:
  - very nice = Ajective Phrase or AP (head is an adjective (A))
  - a very nice looping = NP (head is noun (N)
    - performs a nice looping = VP (head is a verb (V))
    - with a long stick = Prepositional Phrase or PP (head is preposition (P))
    - the cow performs a very nice looping with a long stick = Sentence (S)
- Functions
- Dependency relations between the heads of the constituents: subject (cow), object (looping), main verb (perform), modifier (nice), adjunct (stick)

# Syntactic trees

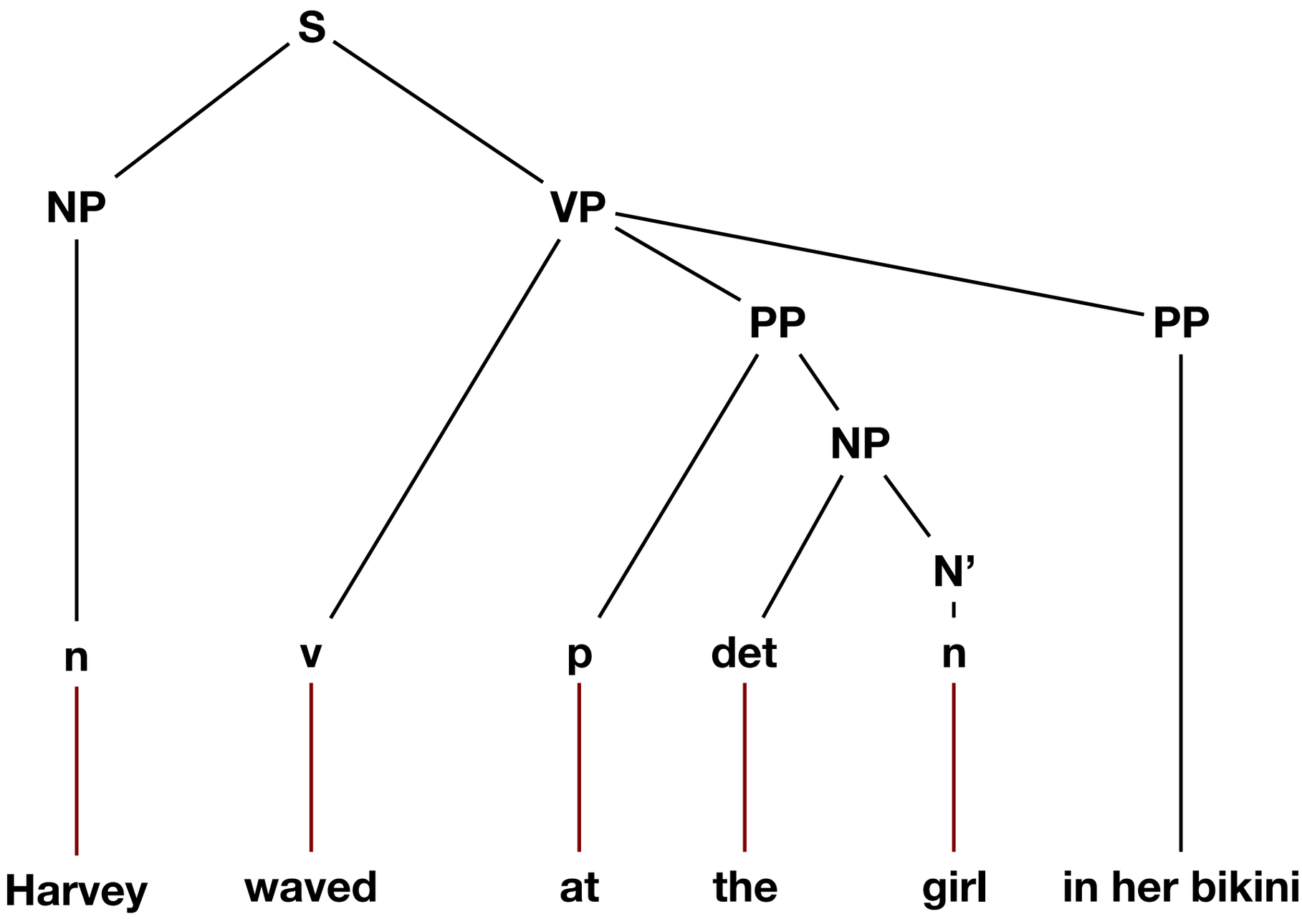
# Syntactic trees



# Syntactic trees



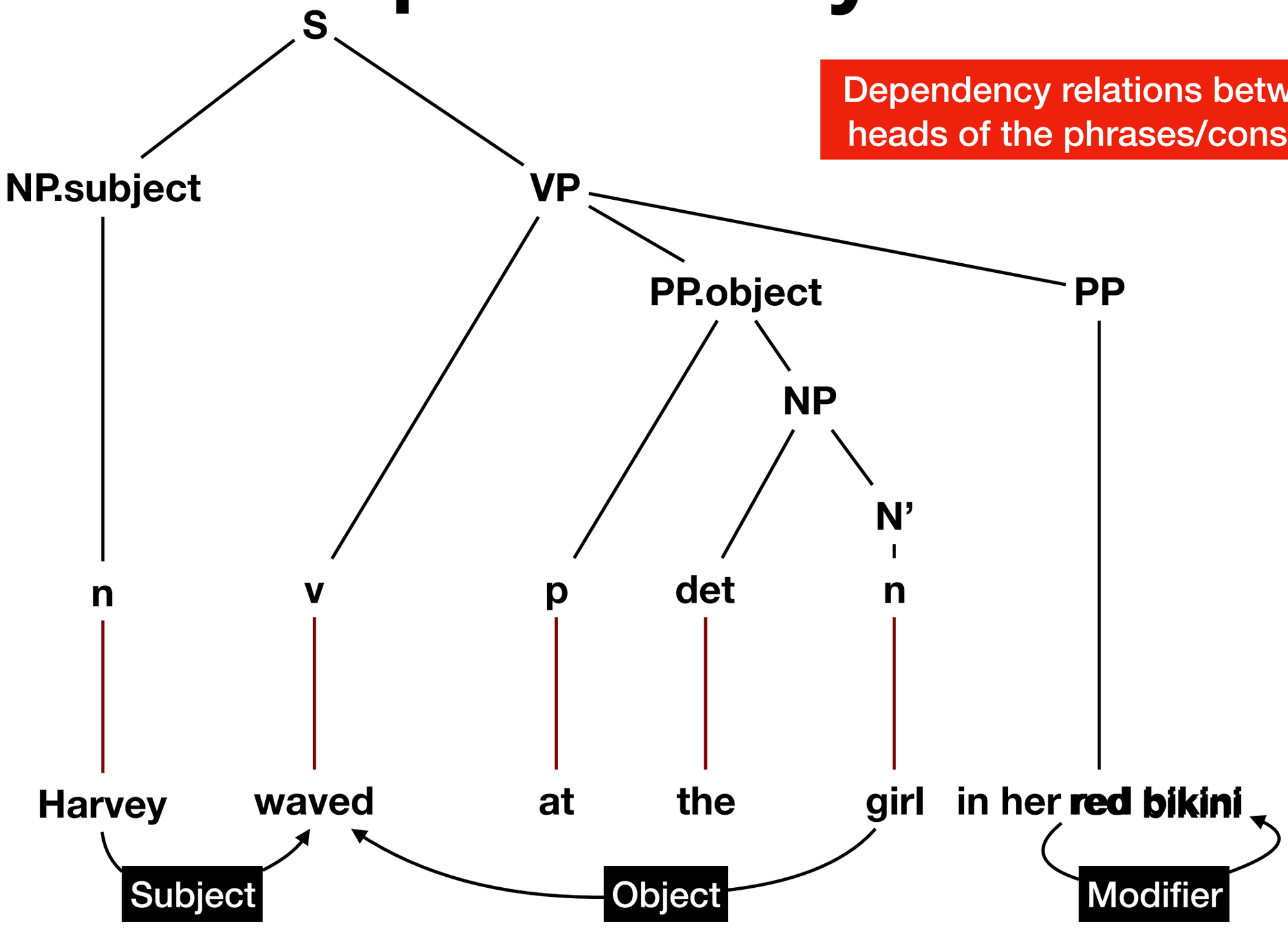
# Syntactic trees



# Syntactic functions

- Grammatical **Subject**: agreement with the main verb
    - the *boys* wave at the girl
    - the *books* were given to me
    - the *boys* were hit by the girl
    - the *girl* hits the boys
- Grammatical **Objects**: obligatory NPs or PPs to form a grammatical sentence
    - *the boys give the girl
    - *the boys fancy
    - *the boys treat the girl
- https://universaldependencies.org/u/dep/

# Syntax Tree with dependency labels




Dependency relations between the heads of the phrases/constituents

# Syntax

- Most important types of predicates in terms of obligatory arguments (the complementation=that what is needed to obtain a grammatical structure)

| Valency | Predicate | Complementation | | | Example |
|---|---|---|---|---|---|
| **Intransitive** | walk.v | NP.subject | | | The cow walks |
| **Transitive** | perform.v | NP.subject | NP.direct object | | The cow performs a looping |
| | count.v | NP.subject | PP(on).pp-object | | The cow is hoping for a big applause |
| | be.v | NP.subject | NP.object \| AP.object | | This cow is a phenomenon. / This cow is phenomenal |
| **Ditransitive** | give.v | NP.subject | NP.direct object | NP.indirect object | The cow gives the spectators an unforgettable day |

- A lexicon provides a list of verbs with their complementation patterns

# Phrase structure parser

- Lookup words from a sentence in a sentence to find a candidate for a main verb
- Get the obligatory arguments of the verb
- Match the structure of surrounding phrases with the structure of the arguments (taken word order into account)
- Match the remaining phrases as non-obligatory elements
- If nothing left, a potential sentence structure is found

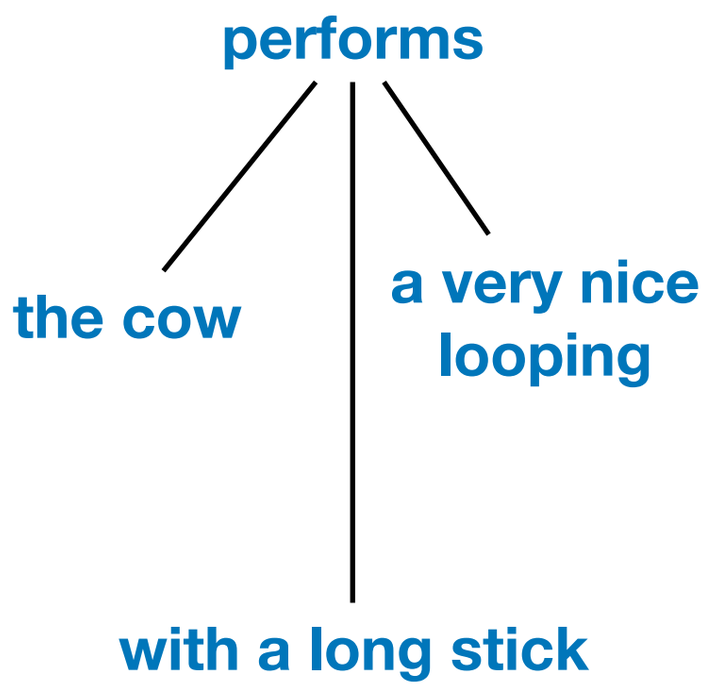
# Syntax in short

- *The cow performs a very nice looping with a long stick.*

- The main verb is the centre of the sentence.

- The main verb gives you the obligatory arguments to make a grammatical sentence

- Next to the obligatory arguments there can also be optional adjuncts



More details

# Syntax: some issues

- Constituents (S, VP, NP, PP, AP, AdvP) can be very small and infinitely large
    - He (NP); The nice green dogs with hats on their head (NP)
- PP-attachment ambiguity is often semantic or context dependent
    - Groucho shot an elephant in his pants.
- Scope is often semantic or context dependent
    - Old men and women can be annoying.
- Argument or adjunct depends on semantics or context
    - I count on your computer.
- A these can be easily combined in one sentence
    - Old men and women may be counting on their computers in their pants.

More details

# Syntax: some issues

- Sentences are often ungrammatical!
    - My trainer don't tell me nothing between rounds. I don't allow him to. All I want to know is did I win the round.
        - Typo’s and somtimez not even a verb
- It's out of the date and lacks validity, so formats of late exams are quite different from its. But for foreign language learners, 'there-insertion' is quite handful.
- => A parser should be robust.

# Parsers

- https://www.nltk.org/book/ch08.html
- Context free grammars use rewrite rules:

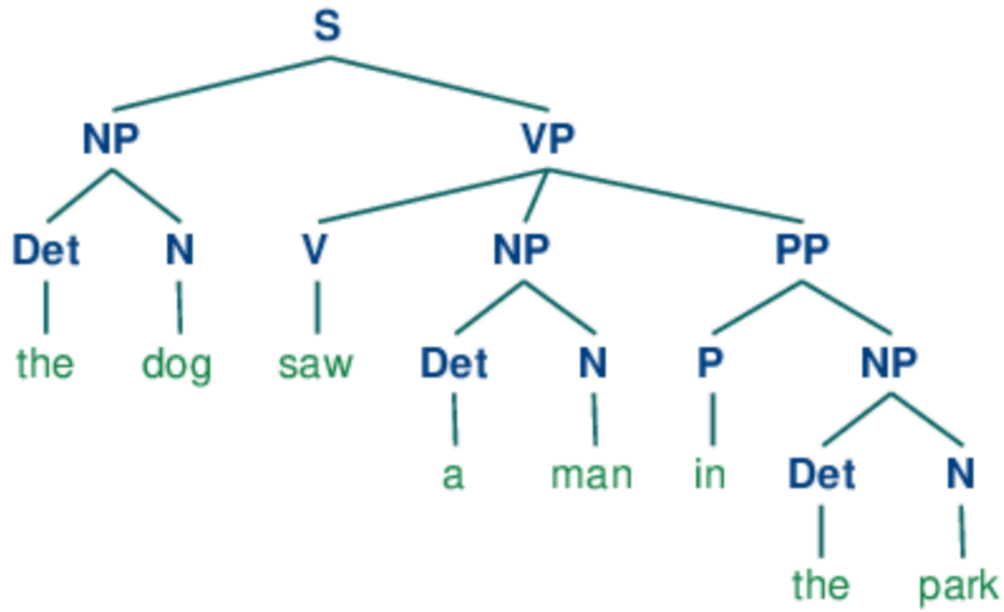
```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "ate" | "walked"
NP -> "John" | "Mary" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "dog" | "cat" | "telescope" | "park"
P -> "in" | "on" | "by" | "with"
```

# Context Free Grammar

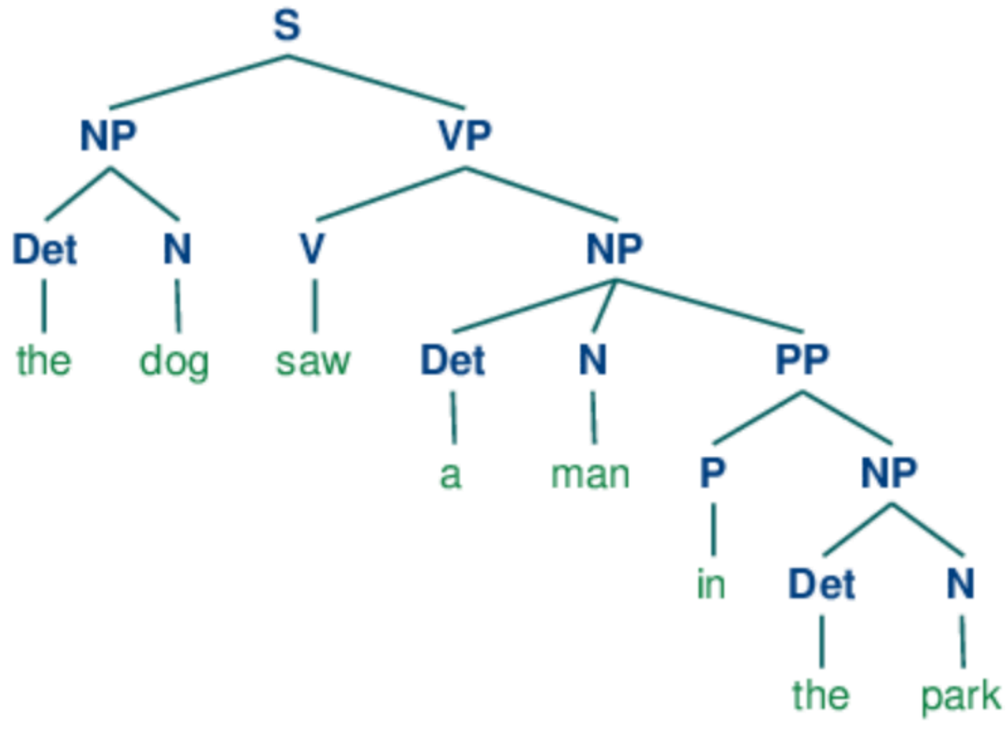
```
S -> NP VP
  VP -> V NP | V NP PP
  PP -> P NP
  V -> "saw" | "ate" | "walked"
  NP -> "John" | "Mary" | "Bob" | Det N | Det N PP
  Det -> "a" | "an" | "the" | "my"
  N -> "man" | "dog" | "cat" | "telescope" | "park"
  P -> "in" | "on" | "by" | "with"
```

| Symbol | Meaning | Example |
|---|---|---|
| S | sentence | *the man walked* |
| NP | noun phrase | *a dog* |
| VP | verb phrase | *saw a park* |
| PP | prepositional phrase | *with a telescope* |
| Det | determiner | *the* |
| N | noun | *dog* |
| V | verb | *walked* |
| P | preposition | *in* |

(9) a.




b.



# Stochastic parser

- Most parsers are trained from so-called treebanks: large collection of manually created parser trees

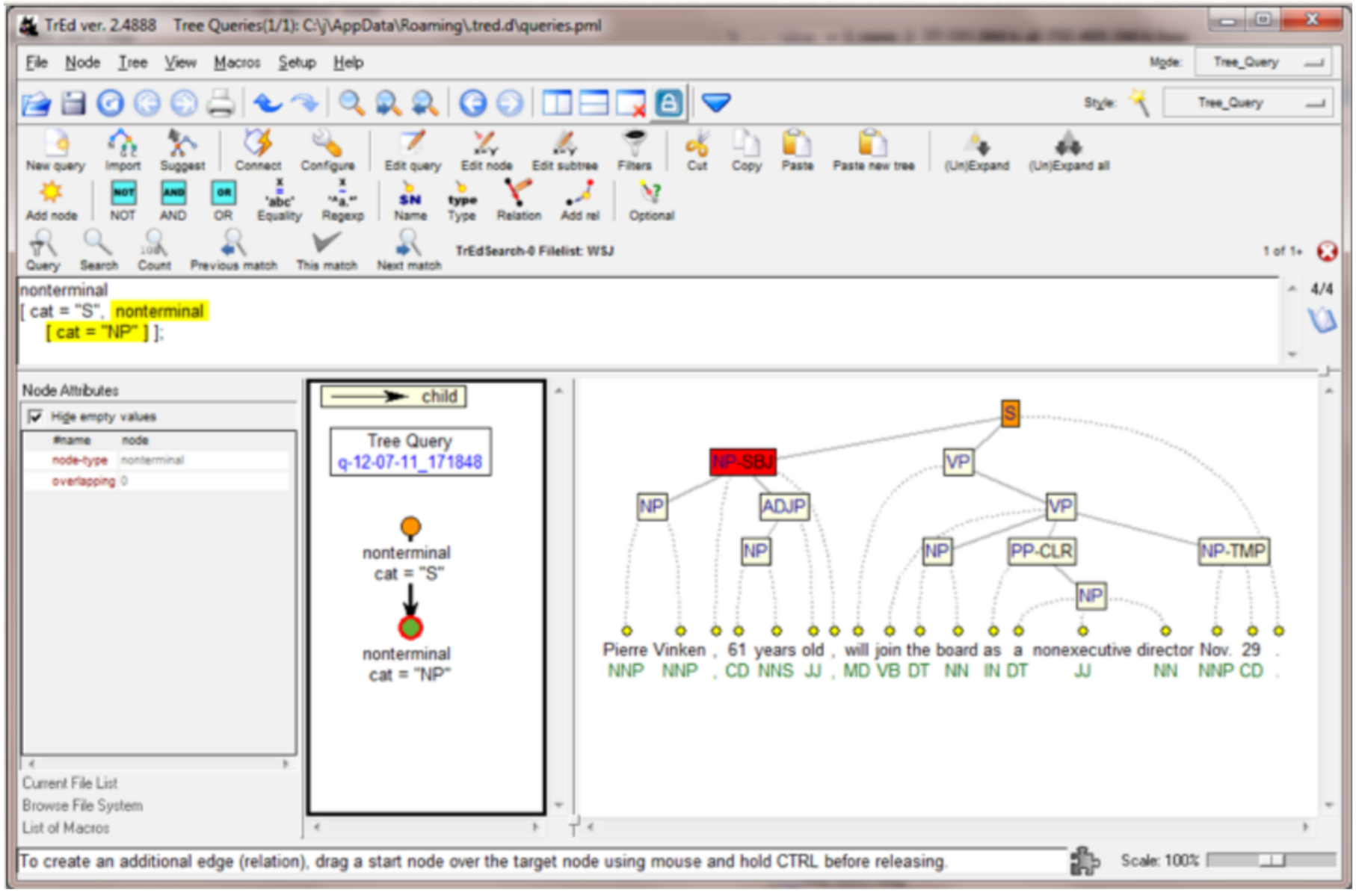
- Usually take tokenised text as input that is also lemmatised and tagged with parts-of-speech
    - Chunkers: only mark phrase structure boundaries
    - Phrase structure parsers add dependency relations
    - Dependency parser label dependencies

- Large quantities of annotated texts are needed

# Treebanks

- Big set of parse trees, often created to train parsers on
  - Penn treebank: http://www.cis.upenn.edu/~treebank
  - Prague dependency treebank: http://ufal.mff.cuni.cz/pdt2.0
  - TiGer treebank: http://www.ims.uni-stuttgart.de/forschung/resourcen/korpora/tiger.en.html
- Penn Treebank:
  - originally just phrase structure, converted to dependencies by Collins (1996)
  - Currently contains basic predicate-argument structure

# Penn treebank



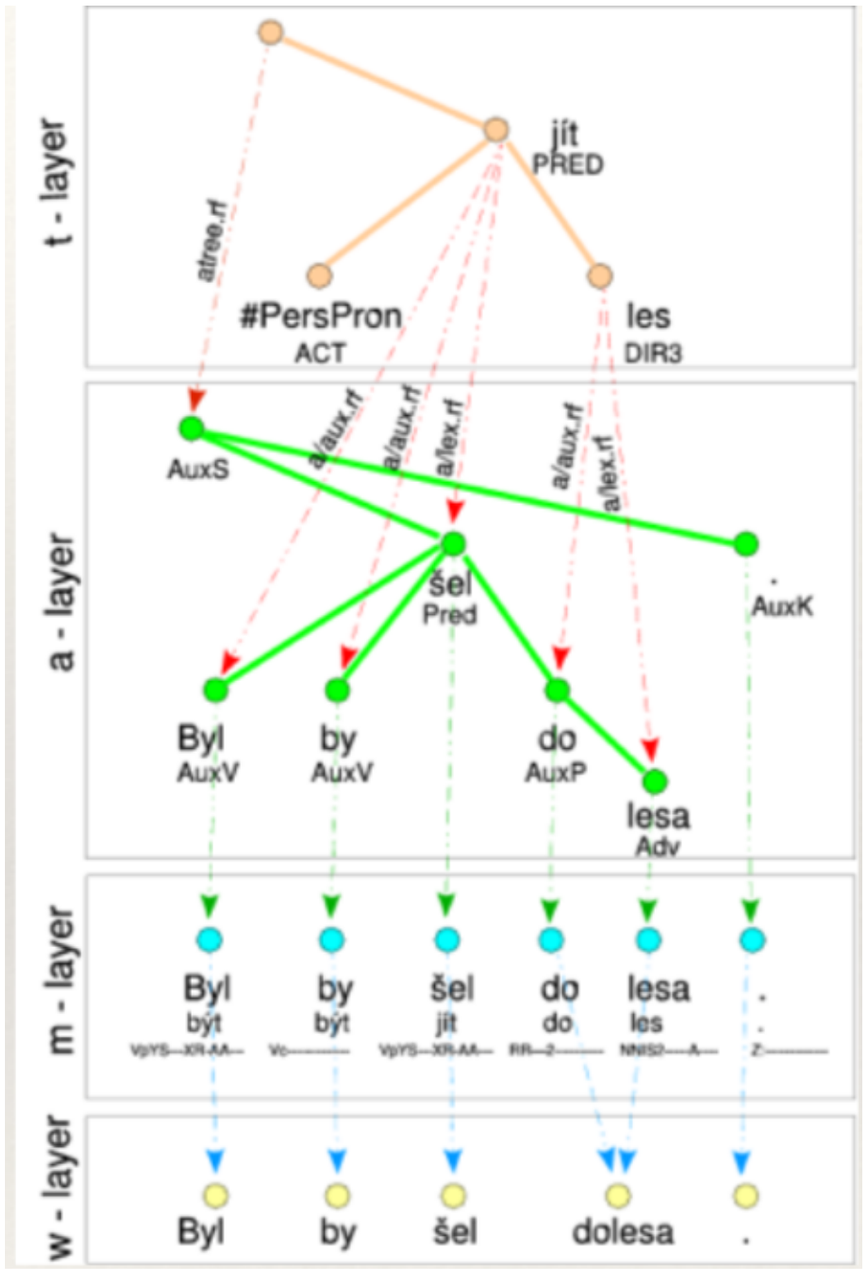

http://faculty.washington.edu/fxia/LAWVI/workshop_presentation_slides/special_session/pml/bak06-pmltq-q.PNG

# Penn Treebank format

**nltk_data/corpora/treebank/parsed/wsj_0003.prd**

```
( (S (PP-TMP In
           (NP July))
     ,
     (NP-SBJ the Environmental Protection Agency)
     (VP imposed
        (NP a gradual ban)
      (PP-CLR on
           (NP (NP (ADJP virtually all)
                 uses)
               (PP of
                (NP asbestos)))))
.))
```

# Prague Dependency Treebank




http://faculty.washington.edu/fxia/LAWVI/workshop_presentation_slides/special_session/pml/dolesa.png

# Dependency Treebank format

**nltk_data/corpora/dependency_treebank/wsj_0016.dp**

```
1   The     DT 3
2   monthly   JJ 3
3   sales   NNS   4
4   have   VBP   0
5   been   VBN   4
6   setting     VBG   5
7   records     NNS   6
8   every   DT 9
9   monthNN     6
10 since   IN 9
11 MarchNNP   10
12 .    .    4
```

# Stanford Parser

- Widely used statistical parser trained on the Penn Treebank
- PCFG: Probabilistic Context Free Grammar
- Also provides labels of dependencies
- Try it out: http://nlp.stanford.edu:8080/parser/index.jsp
- Performance:
    - $F_1$ = .85 for phrase structures
    - $F_1$ = .80 for dependency labelling

# Stanford output

**Tagging**

```
Krelis/NNS  waved/VBD  at/IN  the/DT  girl/NN  with/IN  the/DT  bikini/NN  ./.
```

**Parse**

```
(ROOT
  (S
    (NP (NNS Krelis))
    (VP (VBD waved)
      (PP (IN at)
        (NP (DT the) (NN girl)))
      (PP (IN with)
        (NP (DT the) (NN bikini))))
    (. .)))
```

# Stanford output

**Typed dependencies**

```
nsubj(waved-2, Krelis-1)
root(ROOT-0, waved-2)
prep(waved-2, at-3)
det(girl-5, the-4)
pobj(at-3, girl-5)
prep(waved-2, with-6)
det(bikini-8, the-7)
pobj(with-6, bikini-8)
```

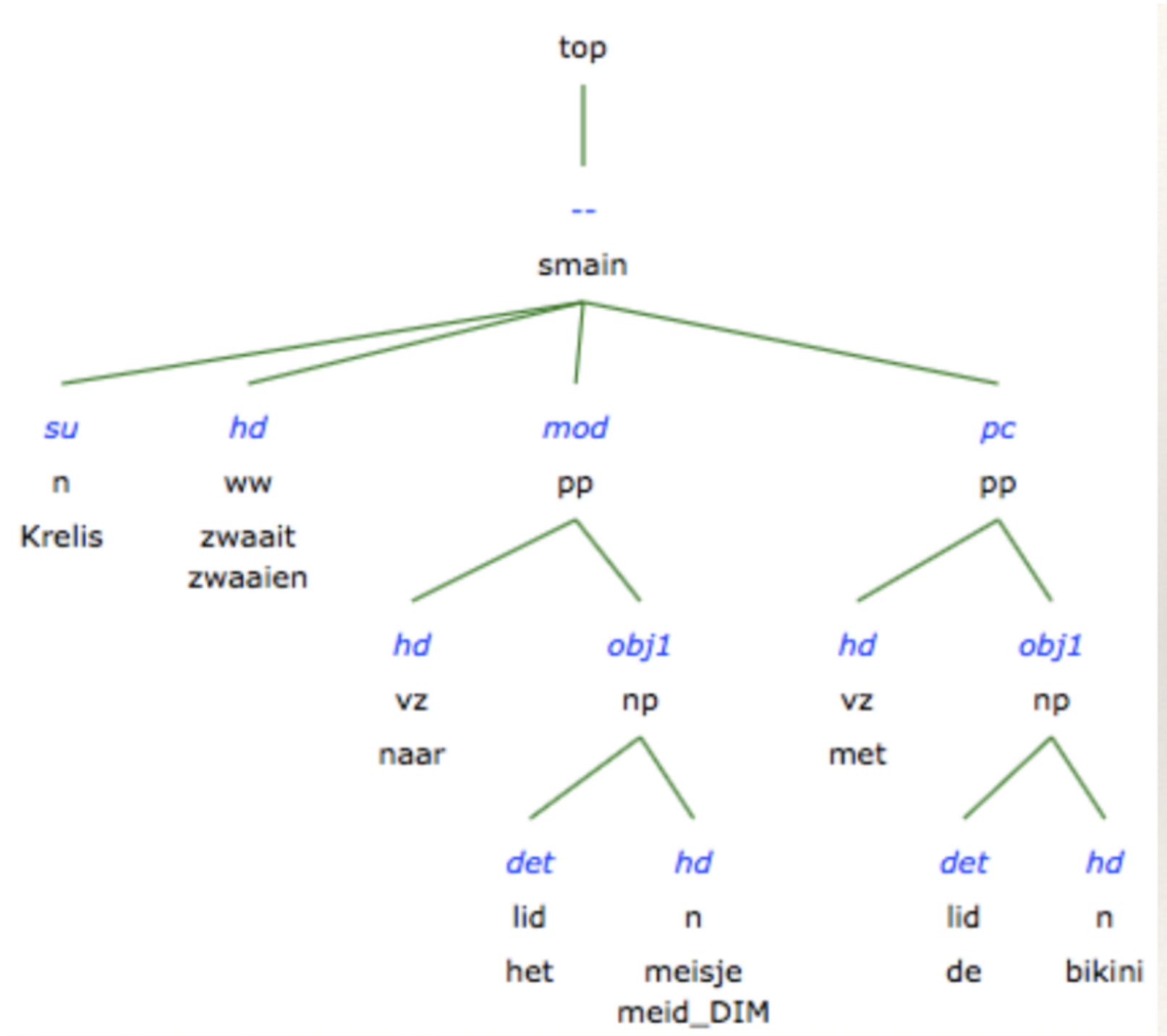
**Typed dependencies, collapsed**

```
nsubj(waved-2, Krelis-1)
root(ROOT-0, waved-2)
det(girl-5, the-4)
prep_at(waved-2, girl-5)
det(bikini-8, the-7)
prep_with(waved-2, bikini-8)
```

# Alpino

- HPSG-based grammar for Dutch
- available at: http://www.let.rug.nl/vannoord/alp/Alpino
- Provides a set of all possible solutions
- Probabilistic parse ranking to find the most likely parse
- Rich output
- Accuracy: ~80%

# Alpino example



# Alpino format

## nltk_data/corpora/alpino/alpino.xml

```
<alpino_ds version="1.2" id="0008">
  <node begin="0" cat="top" end="9" id="0" rel="top">
    <node begin="0" cat="inf" end="8" id="1" rel="--">
      <node begin="2" end="3" id="2" pos="verb" rel="hd" root="doe" word="doen"/>
      <node begin="0" cat="np" end="8" id="3" rel="obj1">
        <node begin="0" end="1" id="4" pos="noun" rel="hd" root="niks" word="Niks"/>
        <node begin="1" cat="ap" end="8" id="5" rel="mod">
          <node begin="1" end="2" id="6" pos="adj" rel="hd" root="anders" word="anders"/>
          <node begin="3" cat="cp" end="8" id="7" rel="obcomp">
            <node begin="3" end="4" id="8" pos="comparative" rel="cmp" root="dan" word="dan"/>
            <node begin="4" cat="inf" end="8" id="9" rel="body">
              <node begin="4" end="5" id="10" pos="adv" rel="mod" root="almaar" word="almaar"/>
              <node begin="5" cat="np" end="7" id="11" rel="obj1">
                <node begin="5" end="6" id="12" pos="adj" rel="mod" root="ruw" word="ruw"/>
                <node begin="6" end="7" id="13" pos="noun" rel="hd" root="materiaal" word="materiaal"/>
              </node>
              <node begin="7" end="8" id="14" pos="verb" rel="hd" root="verzamel" word="verzamelen"/>
            </node>
          </node>
        </node>
      </node>
    </node>
    <node begin="8" end="9" id="15" pos="punct" rel="--" root="." word="."/>
  </node>
  <sentence>Niks anders doen dan almaar ruw materiaal verzamelen .</sentence>
</alpino_ds>
```

# Parsing: good to know

- Stanford & Alpino built for clean newspaper text, not for tweets, messy blogs or forums.

- Parsing is expensive in memory and time

- Challenges:

    - linguistic phenomena such as conjunctions, ellipsis & long distance dependencies

    - problems in tokenisation and PoS-tagging can harm the parser

# If you don’t need full parse trees: Chunking

- Chunking (also called “shallow parsing”) provides a cheap and robust alternative to parsing
- Chunks are like constituents
- Chunkers do not provide full syntax trees, bit lists of constituents up to a certain level in depth (typically 2)
- After chunking a classifier can assign phrase types as well
- [Krelis]NP zwaaide [naar [het meisje]NP]PP [met [de bikini]NP]PP

# CoNLL Chunk annotation

NLTK: nltk_data/corpora/conll2000/train

- Chancellor NNP O
- of IN B-PP
- the DT B-NP
- Exchequer NNP I-NP
- Nigel NNP B-NP
- Lawson NNP I-NP
- 's POS B-NP
- restated VBN I-NP
- commitment NN I-NP
- to TO B-PP
- a DT B-NP
- firm NN I-NP
- monetary JJ I-NP
- policy NN I-NP
- has VBZ B-VP
- helped VBN I-VP
- to TO I-VP
- prevent VB I-VP
- a DT B-NP
- freefall NN I-NP
- in IN B-PP
- sterling NN B-NP
- over IN B-PP
- the DT B-NP
- past JJ I-NP
- week NN I-NP

- CoNLL = Computational Natural Language Learning competition
- Created training and test data for many NLP tasks for various languages.
- Word tokens are listed on a separate line for each document.
- Annotation are added in columns separate by TABs
- IOB annotation style:
  - I = inside
  - O = outside
  - B = beginning

# Dutch treebanks

- https://universaldependencies.org
- https://universaldependencies.org/treebanks/nl-comparison.html
- https://lindat.mff.cuni.cz/repository/xmlui/browse?value=Dutch&type=language

# Words have meanings

- The head of the department lost his head set after hitting his head against the head of the construction while enjoying Head by Prince and heading for his meeting.

- http://live.babelnet.org

- https://en.wikipedia.org/wiki/Head_(disambiguation)

- http://wordnetweb.princeton.edu/perl/webwn?c=7&sub=Change&o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&i=-1&h=00000000000000000000000000000000&s=head

- https://nl.wikipedia.org/wiki/Hoofd

- https://www.vandale.nl/gratis-woordenboek/nederlands/betekenis/hoofd#.XPDv5aeB0Wo

- http://cornetto.clarin.inl.nl/simple_search.xql?word=hoofd&pos=&p_length=20&offset=0&search_type=simple

# Head (disambiguation)

From Wikipedia, the free encyclopedia

The **head** (**Human head**) is the part of an animal or human that usually includes the brain, eyes, ears,

**Head** may also refer to:

## Arts, entertainment, and media [ edit ]

### Music [ edit ]

### Albums [ edit ]

- *Heads* (Bob James album), 1977
- *Head* (The Jesus Lizard album), 1990
- *Head* (the Monkees album), a 1968 soundtrack of the movie
- *Heads* (Osibisa album), 1972

### Songs [ edit ]

- *Head* (The Cooper Temple Clause song), track from *Make This Your Own*
- "Head" (Julian Cope song), 1991
- "Head" (Prince song)
- "Head", a song by Mark Lanegan from *Bubblegum*
- "Head", a song by Static-X from *Beneath... Between... Beyond...*
- "Head", a song by Todd Sheaffer from *The Black Bear Sessions* and *Elko*
- "Head", a song by Lotion from *full Isaac*
- "Heads", a song by Hawkwind from *The Xenon Codex*

### Other music [ edit ]

- Head (band), an English rock band
- The Head (band), an indie rock band from Atlanta, Georgia
- Head (music), a main theme in jazz
- Drumhead, a membrane on a drum
- Headstock, a part of an instrument

### Film and television [ edit ]

- *Head* (film), a 1968 film starring The Monkees
- *Heads* (film), a 1994 TV movie
- *The Head* (film), a 1959 German horror film directed by Victor Trivas
- *The Head*, a 1994–1996 American animated television series
- "Head" (*Blackadder*), a 1986 episode of *Blackadder*
- Head (*American Horror Story*), a 2013 episode of the anthology television series

## WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: head Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"
Display options for word: word#sense number

**Noun**

- S: (n) **head#1**, caput#2 (the upper part of the human body or the front part of the body in animals; contains the face and brains) *"he stuck his head out the window"*
- S: (n) **head#2** (a single domestic animal) *"200 head of cattle"*
- S: (n) mind#1, **head#3**, brain#3, psyche#1, nous#2 (that which is responsible for one's thoughts, feelings, and conscious brain functions; the seat of the faculty of reason) *"his mind wandered"; "I couldn't get his words out of my head"*
- S: (n) **head#4**, chief#1, top dog#1 (a person who is in charge) *"the head of the whole operation"*
- S: (n) **head#5** (the front of a military formation or procession) *"the head of the column advanced boldly"; "they were at the head of the attack"*
- S: (n) **head#6** (the pressure exerted by a fluid) *"a head of steam"*
- S: (n) **head#7** (the top of something) *"the head of the stairs"; "the head of the page"; "the head of the list"*
- S: (n) fountainhead#2, headspring#1, **head#8** (the source of water from which a stream arises) *"they tracked him back toward the head of the stream"*
- S: (n) **head#9**, head word#2 ((grammar) the word in a grammatical constituent that plays the same grammatical role as the whole constituent)
- S: (n) **head#10** (the tip of an abscess (where the pus accumulates))
- S: (n) **head#11** (the length or height based on the size of a human or animal head) *"he is two heads taller than his little sister"; "his horse won by a head"*
- S: (n) capitulum#1, **head#12** (a dense cluster of flowers or foliage) *"a head of cauliflower"; "a head of lettuce"*
- S: (n) principal#2, school principal#1, head teacher#1, **head#13** (the educator who has executive authority for a school) *"she sent unruly pupils to see the principal"*
- S: (n) **head#14** (an individual person) *"tickets are $5 per head"*
- S: (n) **head#15** (a user of (usually soft) drugs) *"the office was full of secret heads"*
- S: (n) promontory#1, headland#1, **head#16**, foreland#1 (a natural elevation (especially a rocky one that juts out into the sea))
- S: (n) **head#17** (a rounded compact mass) *"the head of a comet"*
- S: (n) **head#18** (the foam or froth that accumulates at the top when you pour an effervescent liquid into a container) *"the beer had a large head of foam"*
- S: (n) forefront#1, **head#19** (the part in the front or nearest the viewer) *"he was in the forefront"; "he was at the head of the column"*
- S: (n) pass#9, **head#20**, straits#2 (a difficult juncture) *"a pretty pass"; "matters came to a head yesterday"*
- S: (n) headway#2, **head#21** (forward movement) *"the ship made little headway against the gale"*
- S: (n) point#20, **head#22** (a V-shaped mark at one end of an arrow pointer) *"the point of the arrow was due north"*
- S: (n) question#2, **head#23** (the subject matter at issue) *"the question of disease merits serious discussion"; "under the head of minor Roman poets"*
- S: (n) heading#1, header#1, **head#24** (a line of text serving to indicate what the passage below it is about) *"the heading seemed to have little to do with*

# Hoofd

**Hoofd** kan verwijzen naar:

- hoofd (anatomie) (bij dieren ook *kop*), het bovenste deel van het lichaam
- het hoogste of voorste deel (vgl. Aan het hoofd staan.)
- manager of leidinggevende
- hoofdje, een bloeiwijze
- (per) hoofd van de bevolking, zie per capita
- Krib (rivier), synoniem voor een korte stenen dam in een rivier
- sluishoofd
- hoofd (taalkunde), een term uit de ontleding
- Hoofd (Hoorn), een straat in Hoorn



## Betekenis 'hoofd'

Je hebt gezocht op het woord: hoofd.

**hoofd** *(het; o;* meervoud: *hoofden)*

1 bovenste deel van het menselijk lichaam: *aan iets het hoofd bieden* zich ertegen verzetten; *iemands hoofd eisen* zijn aftreden eisen; *een hard hoofd in iets hebben* een zaak somber inzien; *heel wat aan zijn hoofd hebben* de zorg voor veel dingen hebben; *er hangt ons iets boven het hoofd* er dreigt gevaar; *iem., iets over het hoofd zien* (per ongeluk) niet zien; *uit het hoofd leren* van buiten; *iem. voor het hoofd stoten* kwetsend behandelen; *zich het hoofd breken over iets* erover tobben; *het groeit me boven het hoofd* ik kan het niet meer overzien, het wordt me te veel; *het hoofd koel houden* rustig blijven, niet in paniek raken; *zijn hoofd stoten (a)* het door stoten bezeren; *(b)* afgaan, gezichtsverlies lijden; *het hoofd boven water houden (a)* niet onder de omstandigheden bezwijken; *(b)* zich financieel redden; *uit hoofde van* wegens

2 verstand: *niet goed bij het hoofd zijn* min of meer gek

3 eerste, met leiding belaste, voornaamste persoon

4 persoon: *zoveel hoofden, zoveel zinnen* zoveel mensen, zoveel zienswijzen; *zestig euro per hoofd*

5 het bovenste, voorste gedeelte van iets: *het hoofd van een brief*; *aan het hoofd staan* de leiding hebben

# Sentences have meanings
# *who did what to whom, when where how*

- The man **opened** the door with a key
  - the man = NP, subject, agent
  - the door= NP, direct object, patient
  - with a key = PP, adjunct, instrument
- The key **opened** the door
  - ????
- grammatical constituent + syntactic function + semantic role
- diathesis alternation: semantic roles can be realised through different syntactic structures using the same main verb
- For text mining you need to be able to handle different types of linguistic packaging of the same content (!)

Events and Semantic roles

- **Agent:** performs with control (can stop doing it)
- **Patient:** undergoes the action and is changed by it
- **Instrument:** what the agent uses to perform the action
- **Others:** recipient, thema, source, path, goal …

# From syntax to semantics

**The boy ran from the shop across the street to his mummy**

**The boy fell**

**Harvey bought her flowers**

**She got flowers from Harvey**

**Flowers were given to her by Harvey**

**She broke Harveys eye socket**

**The hammer broke his eye socket**

**His eye socket broke**

**What colors correspond with what semantic relations?**

**Agent, patient, theme, beneficiary/recipient, instrument,**

**Source, path, goal,**

# From syntax to semantics

**The boy ran from the shop across the street to his mummy**

**The boy fell**

**Harvey bought her flowers**

**She got flowers from Harvey**

**Flowers were given to her by Harvey**

**She broke Harveys eye socket**

**The hammer broke his eye socket**

**His eye socket broke**

**What colors correspond with what semantic relations?**

**Agent, patient, theme, beneficiary/recipient, instrument,**

**Source, path, goal,**

# Pragmatics

- In real live language is stretched to serve a purpose in a context (*people always try to make sense*)
    - The three pizzas still need to pay. (**Metonymy**)
    - The salty peanut fell in love with the cashew nut. (**N400** effect in the brain)
    - Can you close the window, please? (Form: question, Meaning: Request)
    - It is a bit cold here, isn’t it? (Form: a declarative sentence, Meaning: request)
- Further readings
    - Nieuwland, Mante S., and Jos JA Van Berkum. "When peanuts fall in love: N400 evidence for the power of discourse." *Journal of cognitive neuroscience* 18, no. 7 (2006): 1098-1111.
    - Grice, H. Paul, Peter Cole, and Jerry L. Morgan. "Logic and conversation." *1975* (1975): 41-58. => ***maxims of conversation***
    - Searle, John R.. ***Speech acts***: An essay in the philosophy of language. Vol. 626. Cambridge university press, 1969.

# Language as Data

- Complex, rich and still incomplete (abstract)
- Ambiguity is pervasive
- Variation is abundant: genres, media
- Data is skewed (Zipfian distributions of meaning and form)
- Data sets for training and testing are small, suffer from semantic overfitting, hardly show variation, poorly sampled over time
- NLP technology suffer from bias, semantic and form overfitting, task overfitting

# Ambiguity

- *Structural ambiguity:*
  - *“Fruit flies like a banana”*

travels through the air
in a similar fashion to

- *Lexical ambiguity: words have more than one meaning (polysemy)*

- Foot Heads Arms Body.
- Hospitals Sued by 7 Foot Doctors.
- British Left Waffles on Falkland Islands.
- Stolen Painting Found by Tree.

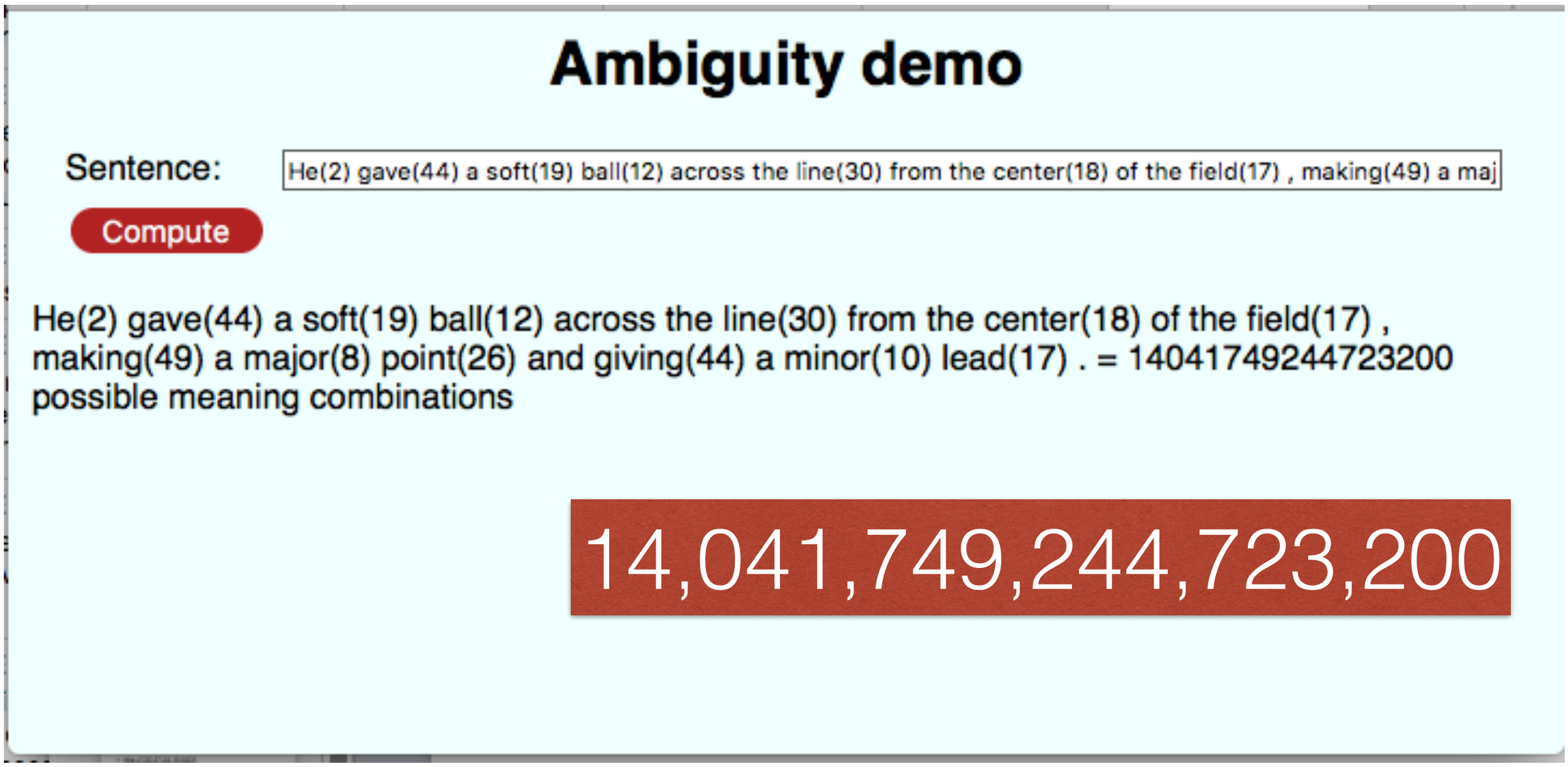
Examples taken from: Maynard, Bontcheva, Augenstein Natural language processing for the semantic web

# Ambiguity is pervasive and we do not perceive it!!!

- 121 most frequent English nouns have on average 7.8 meanings each and account for about 20% of word occurrences in real text (in the Princeton WordNet (Miller 1990), according to Ng and Lee (1996)).

- “He gave a soft ball across the line from the center of the field, making a major point and giving a minor lead.”

# The meaning puzzle



# Variation

How many words for a thing?

# Variation in expressions

## Different words and expressions

- BNC Holdings Inc named Ms G Torretta as its new chairman.
- Nicholas Andrews was succeeded by Gina Torretta as chairman of BNC Holdings Inc.
- Ms. Gina Torretta took the helm at BNC Holdings Inc.

## Spread over multiple sentences involving coreference

- After a long boardroom struggle, Mr Andrews stepped down as chairman of BNC Holdings Inc. He was succeeded by Ms Torretta.

Examples taken from: Maynard, Bontcheva, Augenstein Natural language processing for the semantic web

# 5000 words for person

mortal; posturer; controller; **withholder**; suppressor; subduer; fugitive; divider; subdivider; outcaste; bereaved_person; yielder; nude; streaker; **unperson**; baby_buster; neighbor; loose_cannon; ladino; communicator; announcer; town_crier; caller; caller; muezzin; hisser; gossipmonger; yenta; cat; telltale; **scandalmonger**; allegoriser; presenter; promisee; quoter; transmitter; spammer; answerer; hedger; assenter; interviewee; **don’tknow**; testee; passer; popularizer; avower; laudator; clapper; waffler; wirer; conferrer; confessor; reporter; newswoman; television_reporter; anchorman; avower; postulator; author; gagwriter; poet; sonneteer; poetess; homer; elegist; frost; key; gilbert; gray; pound; poet_laureate; odist; spender; poet_laureate; bard; biographer; autobiographer; hagiographer; novelist; folk_writer; folk_poet; **cyberpunk**; west; wood; authoress; abstractor; pamphleteer; speechwriter; drafter; paragrapher; space_writer; tragedian; snow; day; playwright; rice; kid; cooper; buck; scriptwriter; film_writer; wordmonger; framer; literary_hack; rand; coauthor; scenarist; litterateur; **word-painter**; wordsmith; alliterator; sand; e._e._cummings; heller; grass; spark; jour nalist; photojournalist; reed; stone; sob_sister; sports_writer; newspaperwoman; war_correspondent; foreign_correspondent; broadcast_journalist; gazetteer; columnist; newspaper_columnist; newspaper_critic; gossip_columnist; agony_aunt; scribbler; rhymer; lyrist; rice; ghostwriter; librettist; compiler; encyclopaedist; lexicologist; etymologist; synonymist; neologist; polemist; commentator; contributor; twaddler; alarmist; stirrer; letter_writer; pen_pal; broadcaster; telecaster; announcer; sportscaster; tv_announcer; newscaster; news_reader; radio_announcer;

# 8643 words for persoon

- naarling, beroerling, ellendeling, etterbak, etterbuil, fielt, fluim, gemenerik, hond, hondenlul, kankerlijer, kelerelijder, kelerelijer, klerelijer, kloot, kloothommel, klootspiraal, klootzak, kwal, lamgat, lammeling, lamstraal, lamza lazersteen, lazerstraal, loeder, lul, lulhannes, lulletje, miesgasser, mispunt, onverlaat, paardelul, paardenlul, patjakker, pleurislijder, ploert, plurk, pokkenlijer, pokkenvent, pooier, rasploert, reptiel, rotzak, schoelje, schoft, serpent, smeerlap, stinker, teringlijder, tyfuslijer, vuilak, zakkenwasser, zwijn, zak, hondelul, etter, lelijkerd, smiecht, pokkenlijder, sekreet, stinkerd, individu
- huichelaar, Januskop, draaikont, farizeeër, hypocriet, januskop, jezuïet, smoelentrekker, valsaard, valserik, veinzaard, veinzer
- onruststoker, aanstoker, aanzetter, agitator, herrieschopper, onrustzaaier, oproerkraaier, opruier, paniekzaaier, provocateur, raddraaier, roervink, stemmingmaker, stokebrand, stoker, woelgeest
- boef, booswicht, galgeaas, galgebrok, galgenaas, gannef, kwaaddoener, satan, slechterik, snoodaard, spitsboef, zwijnjak, schurk
- krankzinnige, fanatiekeling, geesteszieke, gek, gestoorde, waanzinnige, fanate
- dwaas, achterlijke, gek, halvezool, idioot, imbeciel kwibus, lijp, lijpo, mafkees, mafketel, mafkikker, malloot, nar piechem, zot, druif, debiel

# 4,000 words for move

go; swim; buoy; drive; island_hop; whistle; ski; slalom; hot-dog; wedel; water_ski; schuss; pass_over; breeze; err; return; revisit; retrace; cut_back; resurrect; return; home; head_home; double_back; bounce; boomerang; fly; come; retrograde; walk; constitutionalize; speed; bang; **swash**; tread; step_on; beetle; circulate; drift; swim; bucket_along; shoot; rip; barge; dash; plunge; hurtle; sit; canter; override; prance; ride_herd; ride_horseback; prance; post; trot; gallop; canter; gallop; outride; lance; scramble; plough; sift; **zigzag**; billow; pursue; haunt; tail; tree; hound; ferret; run_down; quest; stalk; roll; troll; bowl; travel_purposefully; wend; whisk; cruise; stooge; steamer; go_forward; head; make; trace; limp; wander; roar; ease; circulate; float; ride; shack; draw; caravan; career; raft; swap; thrash; retreat; cocoon; automobile; step; backpedal; blow; stream; tide; waft; crawl; formicate; slice_into; run; precede; lead; draw_away; travel_along; ascend; heel; turn; angle; push; travel; ride; fly; cruise; ship; sail; wind; snake; pan; repair; taxi; precess; cast; **jazz_around**; maunder; travel; itinerate; go_up; rise_up; resurface; well; intumesce; emerge; uprise; ferry; transfer; betake_oneself; march_on; string; edge; forge; penetrate; rachet_up; sneak_up; plough_on; draw_in; slide_by; fell; impinge; overtake; clear; hop; get_by; tram; prance; derail; go_through; get_across; stride; take; ford; tramp; jaywalk; crisscross; bridge; walk; hop; course; cut; muscle; lock; negociate; pass_through; make; jostle; bushwhack; claw; pass_over; cut; crash; transit; blunder; cycle; cycle_on; squeak_by; break_through; run; overstep; go_around; drive; pull; cut_in; wing; fly_on; soar; rack; buzz; hover; poise; flight; go_down; go_down; drip; correct; subside; dismount; pitch; go_down; founder; subside; submerge; dive; belly-flop; jackknife; flop; rope_down; cascade; drop; flump_down; decline; dip;

# 2,037 words for noise

grinding; racket; report; squeak; clap; clack; snore; chatter; chattering; brouhaha; hubbub; uproar; katzenjammer; clatter; shrieking; scream; screech; screaming; shriek; screeching; blaring; blare; din; cacophony; clamor; grumble; rumble; grumbling; rumbling; squawk; plump; crackling; crackle; crepitation; decrepitation; snap; explosion; squish; rhonchus; hum; humming; swoosh; whoosh; clangoring; clang; clank; clash; clangor; crash; clangour; whisper; whispering; rustling; rustle; chug; sizzle; plonk; howl; squeal; plop; scrape; scratching; scratch; scraping; chatter; chattering; ding-dong

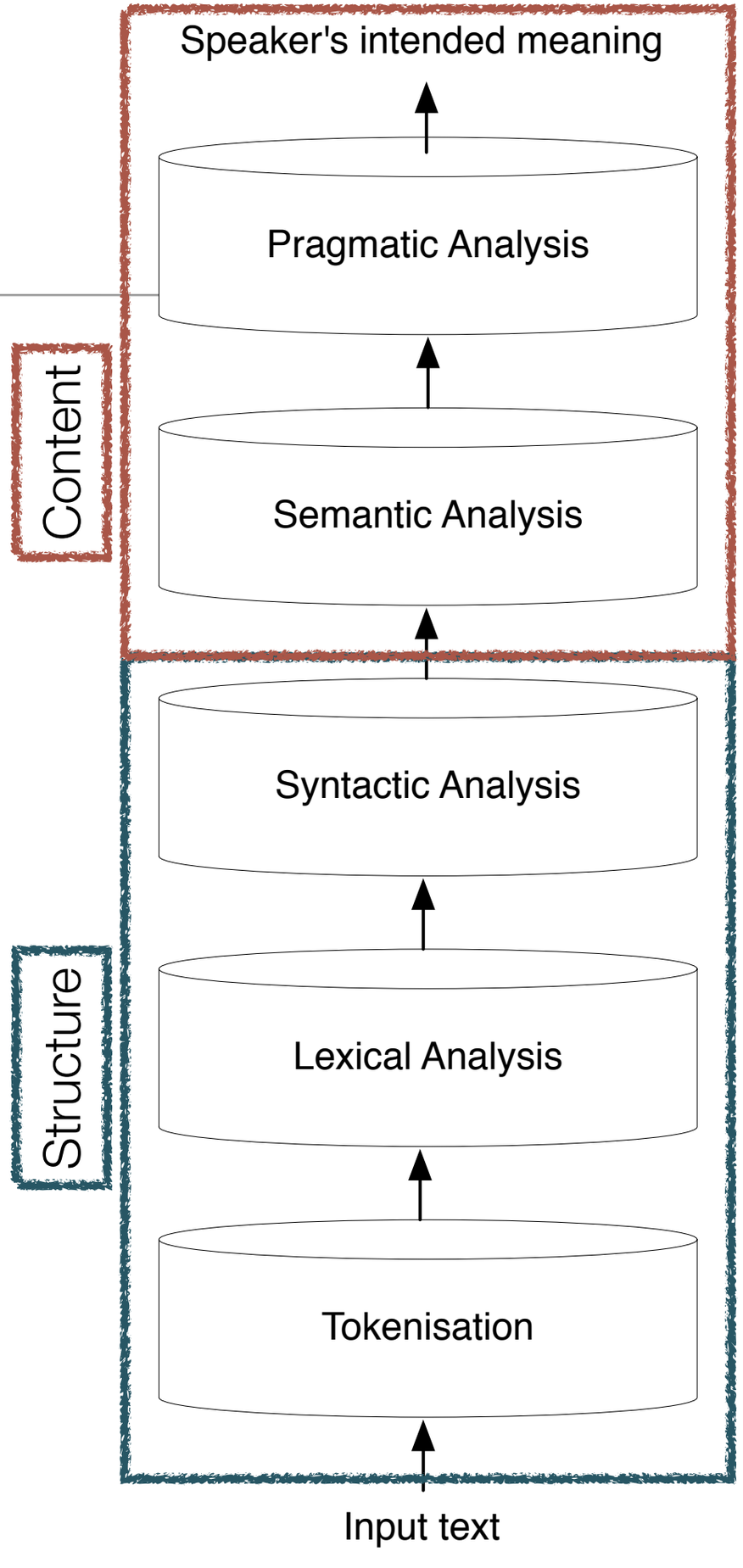
# Part II: NLP Pipelines

The art of digging text

Patterns and rules
Machine learning
Pipelines

# Natural Language Processing

- Complex problem is broken down into a number of smaller problems
- Simple, structural problems solved first and higher-level semantic tasks are solved later, using the output of earlier modules as input:
  - pipeline architecture with dependencies across modules
- For each problem different techniques:
  - knowledge-base & rules (linguistic knowledge)
  - machine learning (supervised and unsupervised), data driven



# We always need to do preprocessing

- First problem, what is a word, what is a sentence?

- **Tokenization**

  - nitty-gritty, data-base, (semi-)irony, $523,45, 21st century, 9-11, Encoding issues (Latin, UTF-8/16, diacritiçćčs, “‘’`` quotes…), don’t, men’s, end-of-sentence hy-p-h-ens

- **Sentence splitting**

  - Dr., Mrs., Bol.com, 7.5, etc. etc, white spaces, TABs, new lines, HTML markup <body><h1></h1><p><li></body>

# All text needs preprocessing

- **Example***:*

  *Documents filed to the San Jose federal court in California on November 23 list six Samsung products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Apple claims infringe its patents.*

# All text needs preprocessing

- **Example***:*

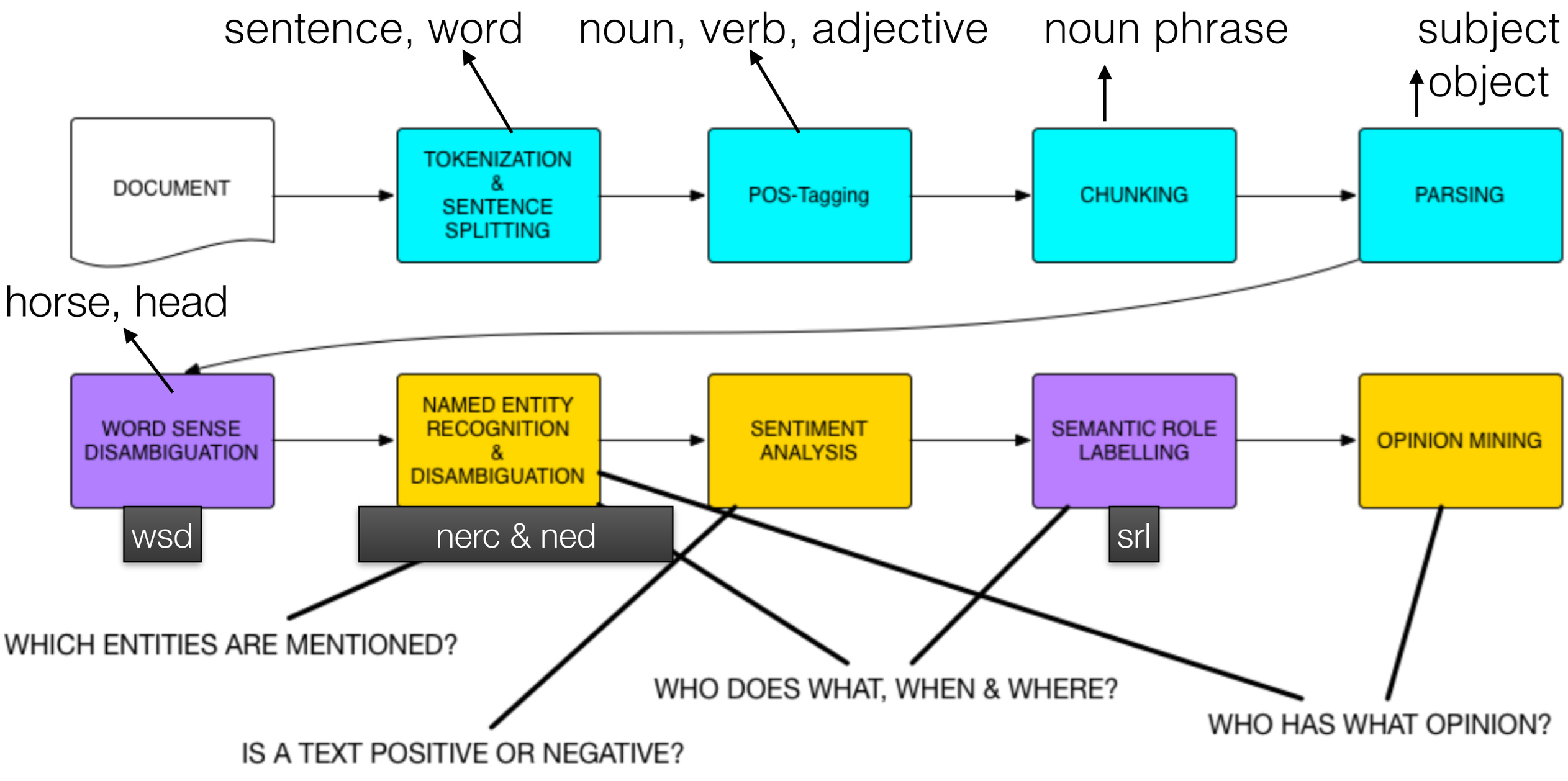
  *Documents filed to the San Jose federal court in California on November 23 list six Samsung products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Apple claims infringe its patents.*

  - *Documents —> documents —> document —> noun or verb?, subject-of-list, object-of-filed*

  - *the "Jelly Bean" and "Ice Cream Sandwich" operating systems, (9 tokens)*

  - *the “ Jelly Bean “ and “ Ice Cream Sandwich “ operating systems , (14 tokens)*

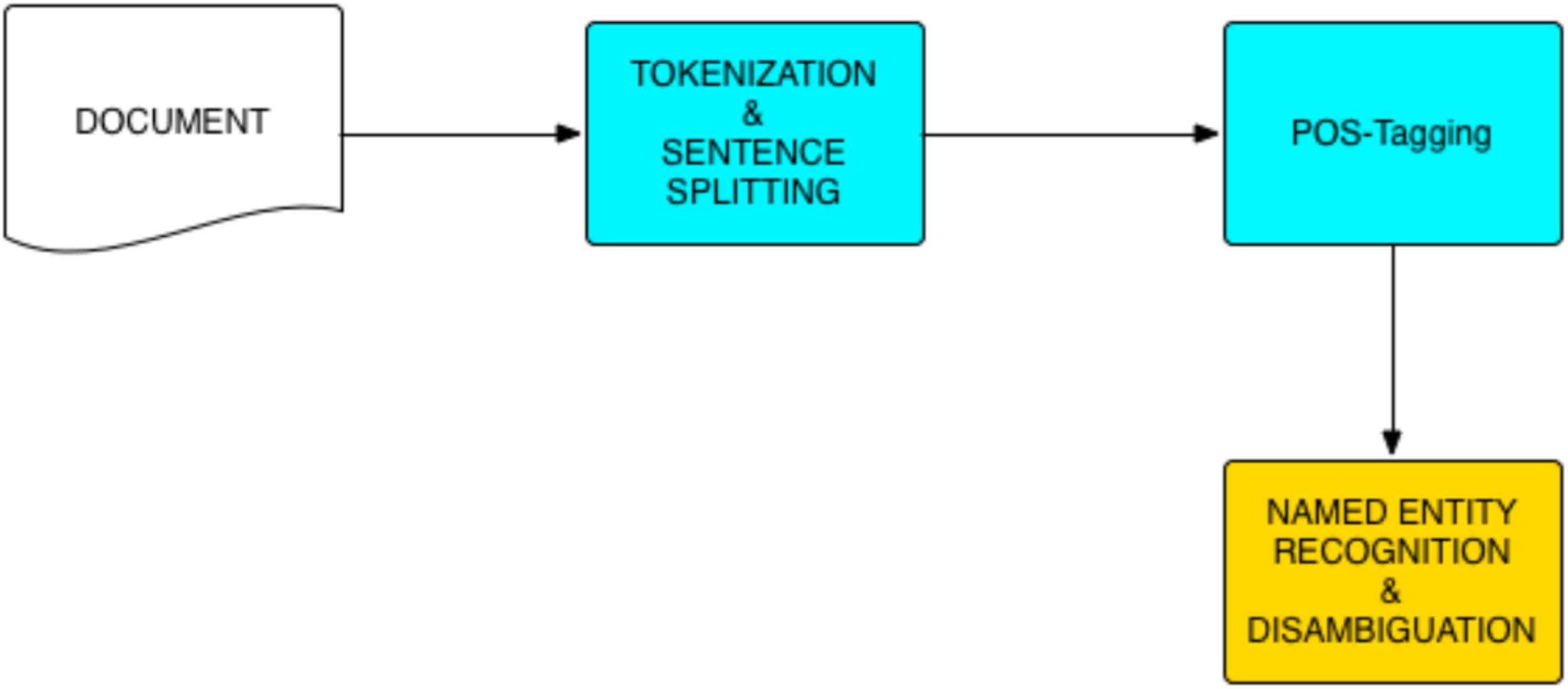
# Example of an NLP pipeline



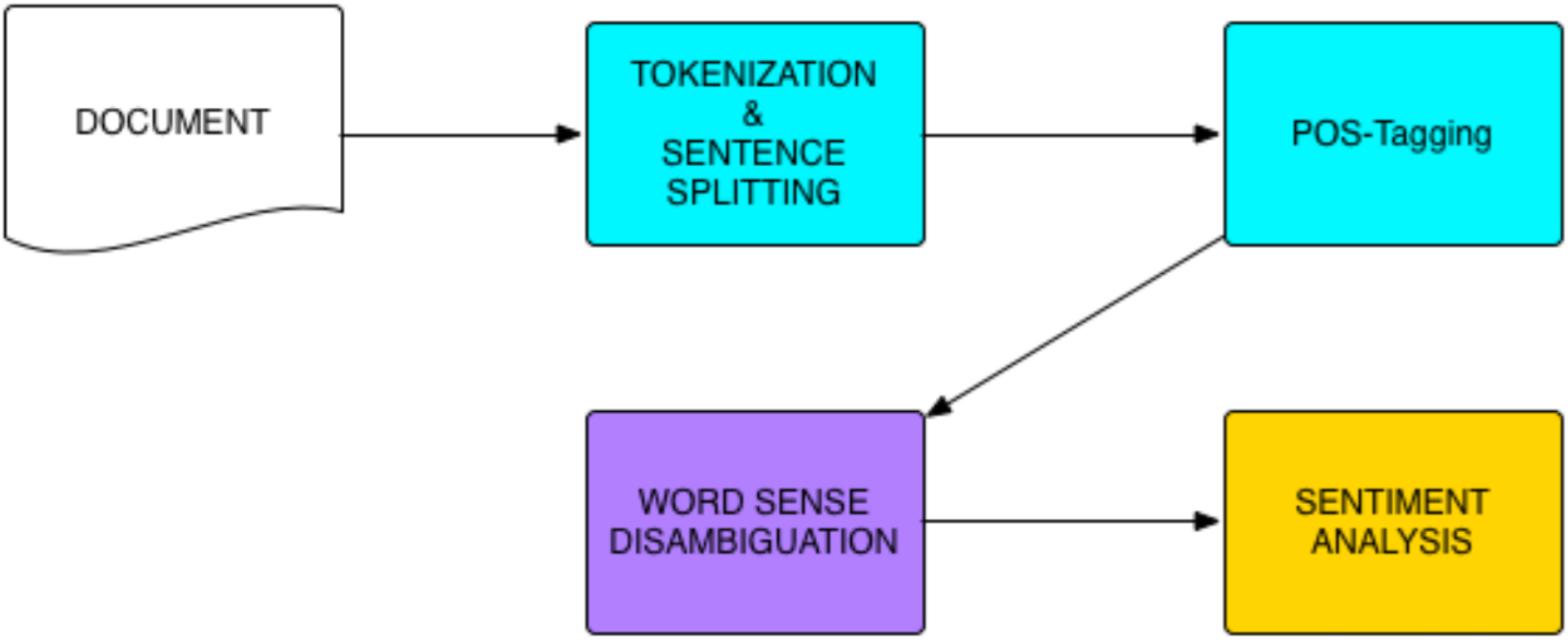
# Communication between modules

- Modules in a pipeline often reuse information from a previous step
- Modules must therefore be able to communicate with each other
- Frameworks that support integration of several steps
- Representation formats that are compatible or easily adaptable

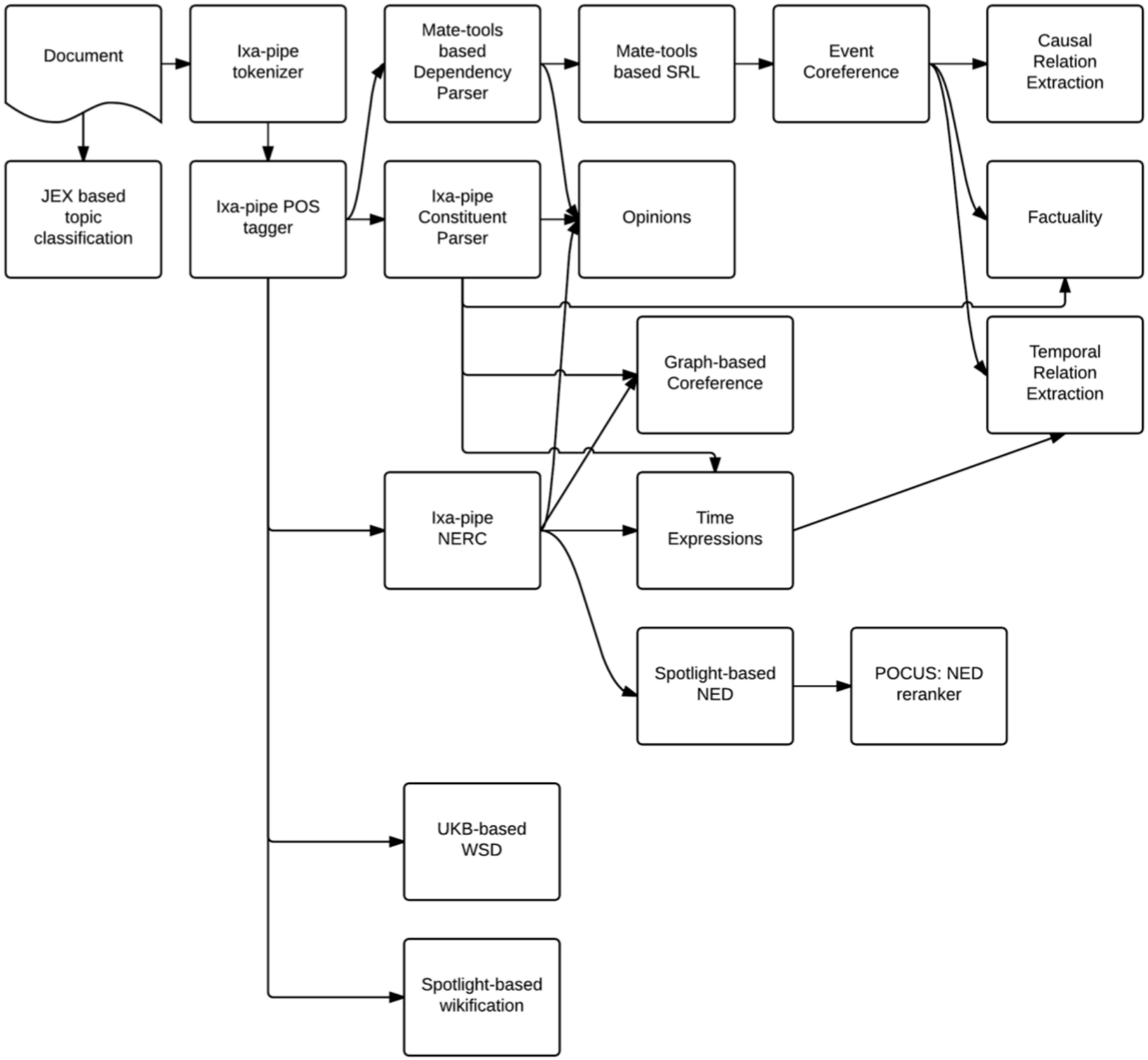
# Named Entity Recognition Pipeline Example



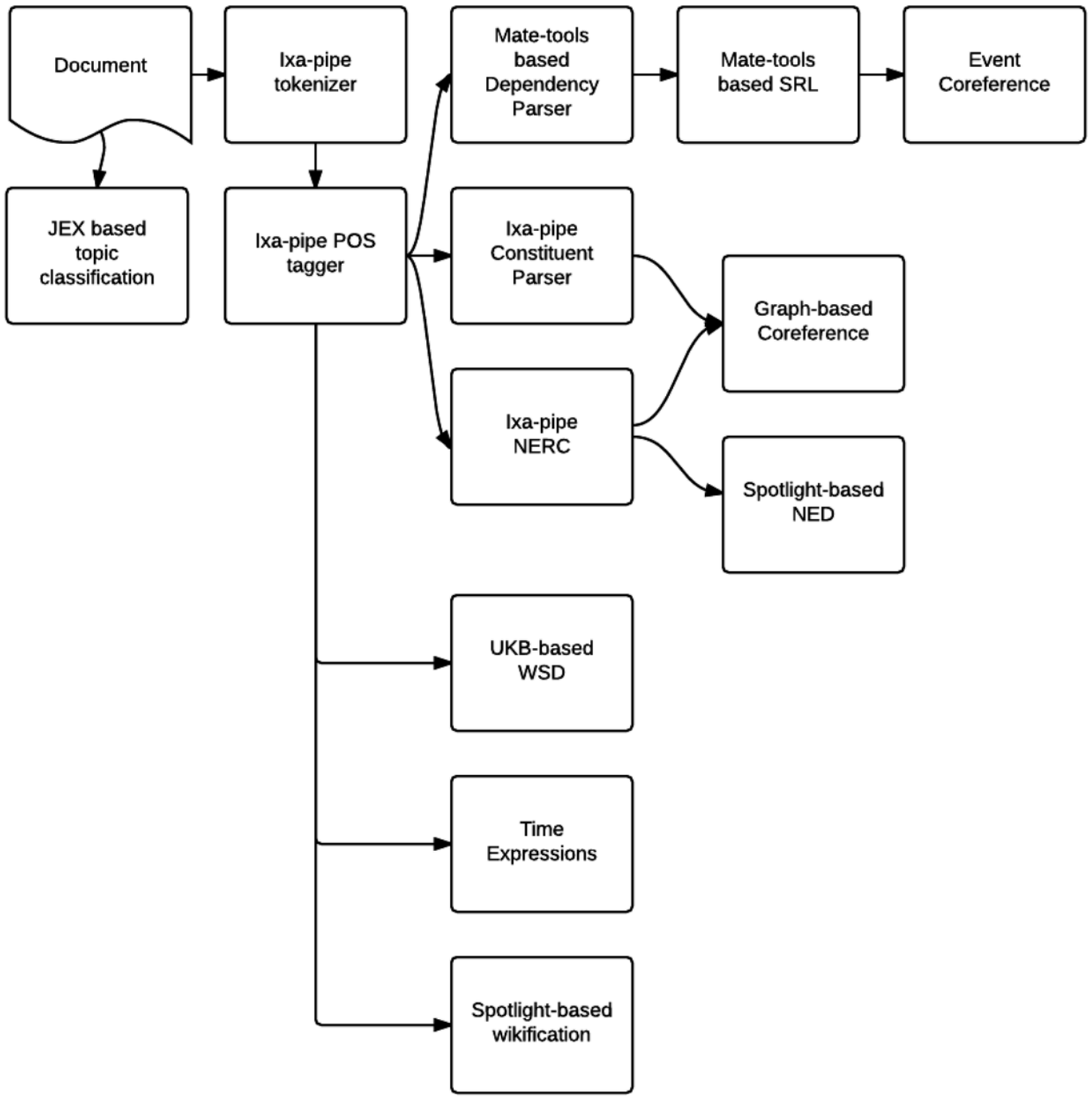
# Sentiment Analysis Pipeline Example



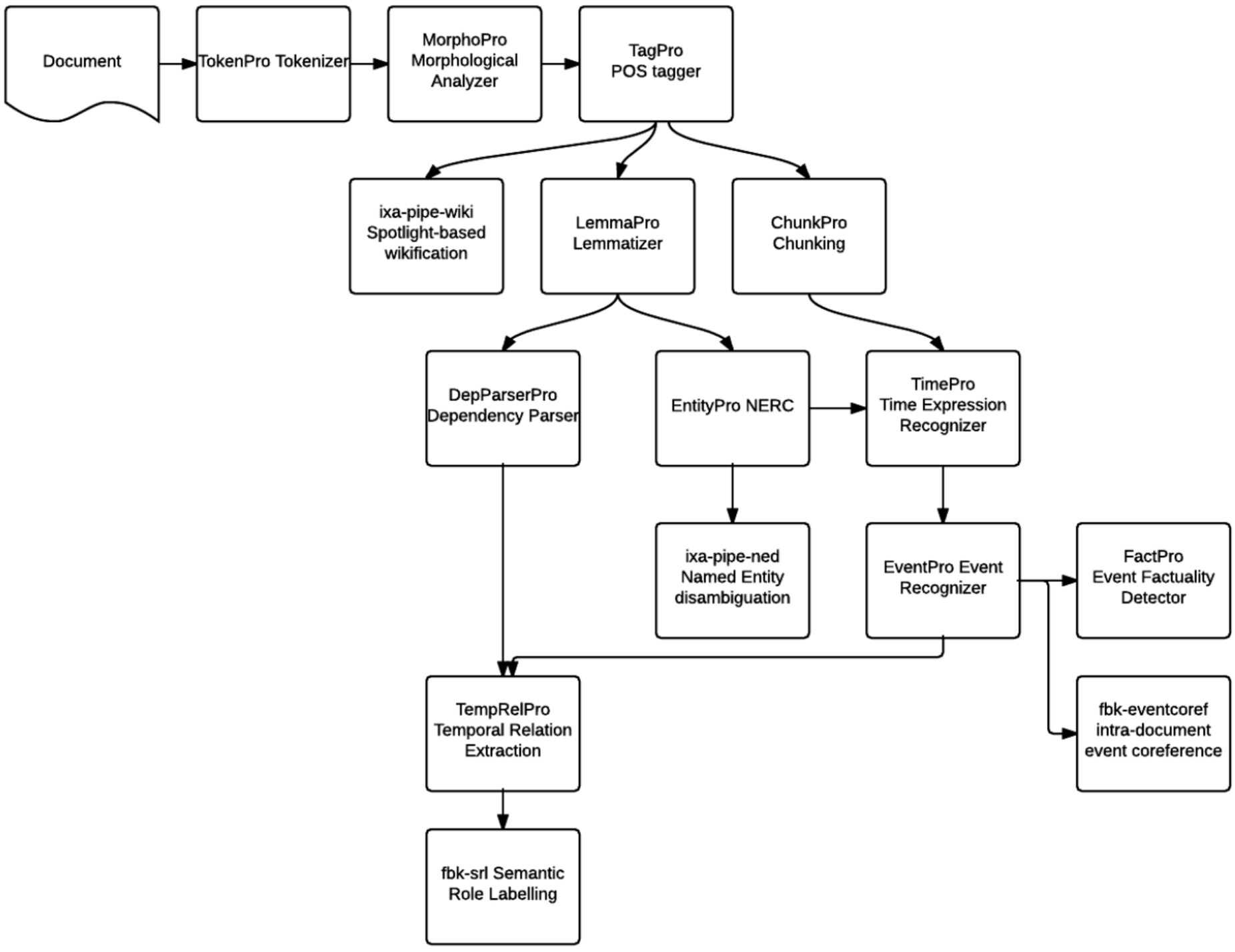
# NewsReader: English pipeline



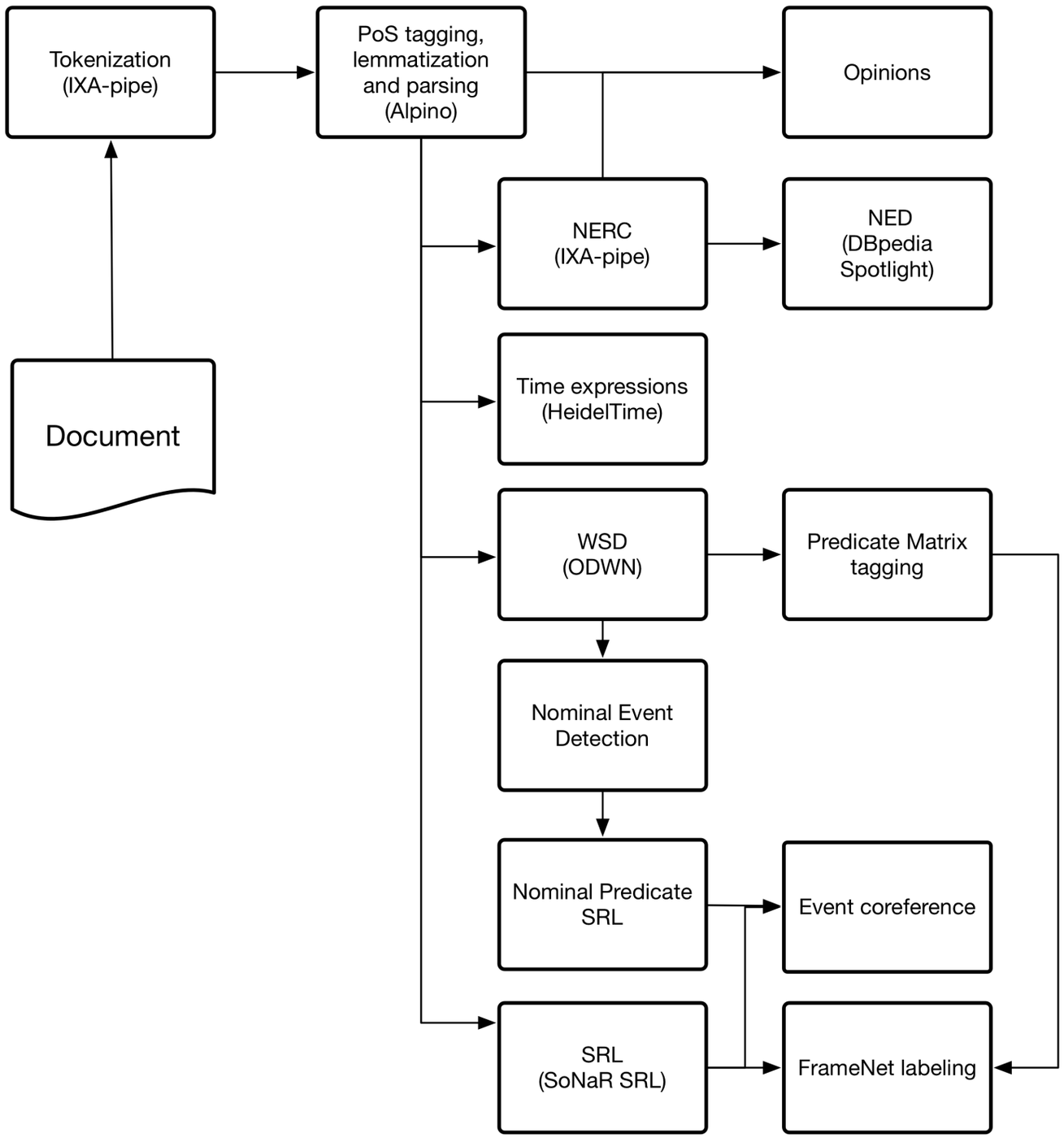
# Spanish pipeline



# Italian pipeline

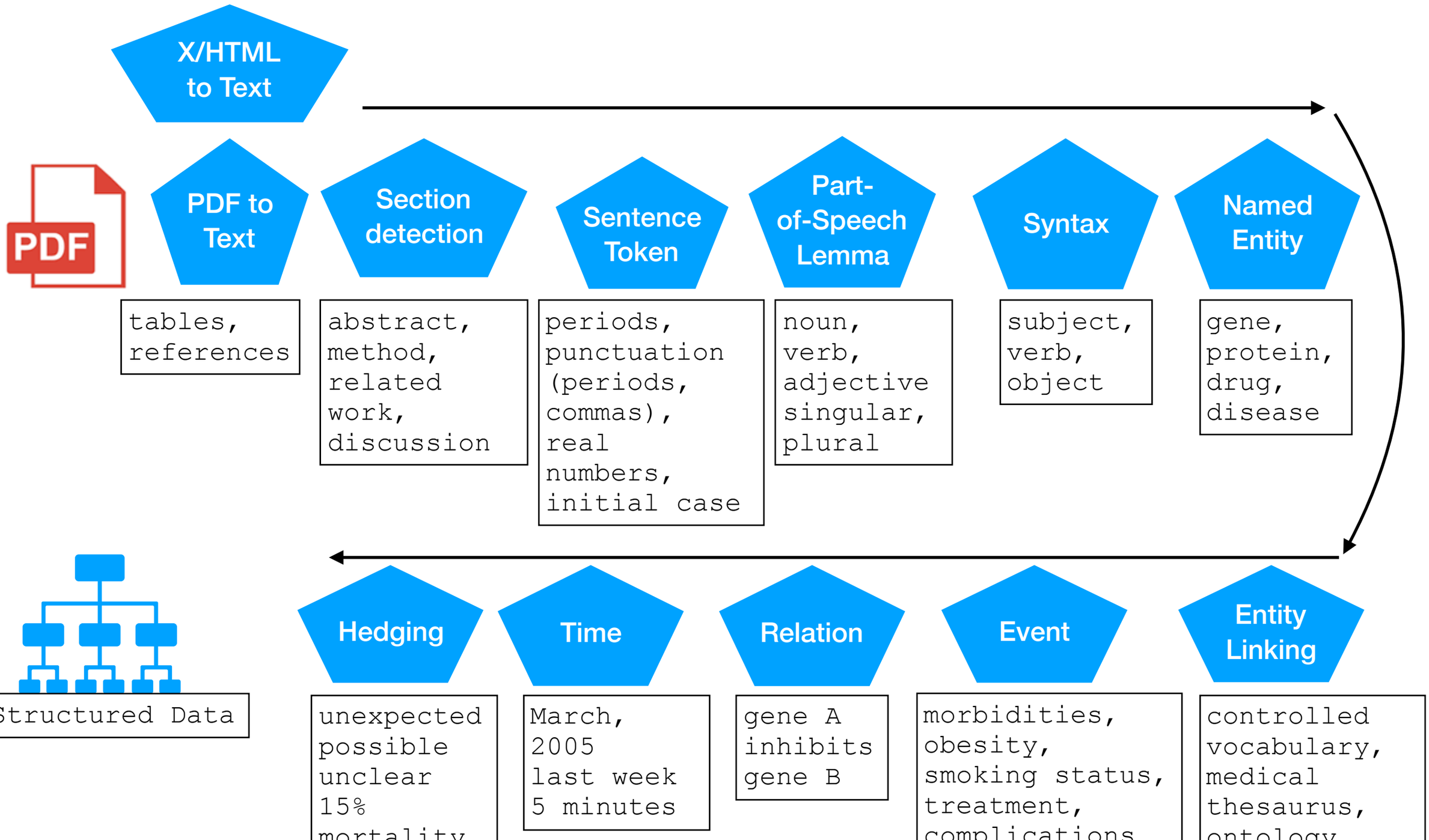


# Dutch pipeline



# Medical NLP Pipeline

scientific articles & clinical documents: admission notes, discharge summaries, radiology reports, pathology reports, etc.



# Some issues

- Dependencies across modules result in error propagation
- Ambiguities (multiple values with confidence score) are not exploited by next levels
- Conflicts: different modules state information that is not compatible
- Complex and difficult to maintain, e.g. input and output needs to be interoperable across modules

# NLP Documentation

**Computational linguistics**

- Juravsky and Martin, Speech and Language Processing, 3rd edition:
  - 2017 edition: JuravskyMartin_ed3book-2017.pdf
  - Current edition https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
- Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit: www.nltk.org/book/

**Programming**

- Anaconda Python distribution: https://www.anaconda.com/distribution/
- Python documentation: https://docs.python.org/3/.
- Code editors :Atom (https://atom.io/) and PyCharm (https://www.jetbrains.com/pycharm/) .
- References:
  - Python cookbook: chimera.labs.oreilly.com/books/1230000000393/index.html
  - Think Python: greenteapress.com/wp/think-python-2e/

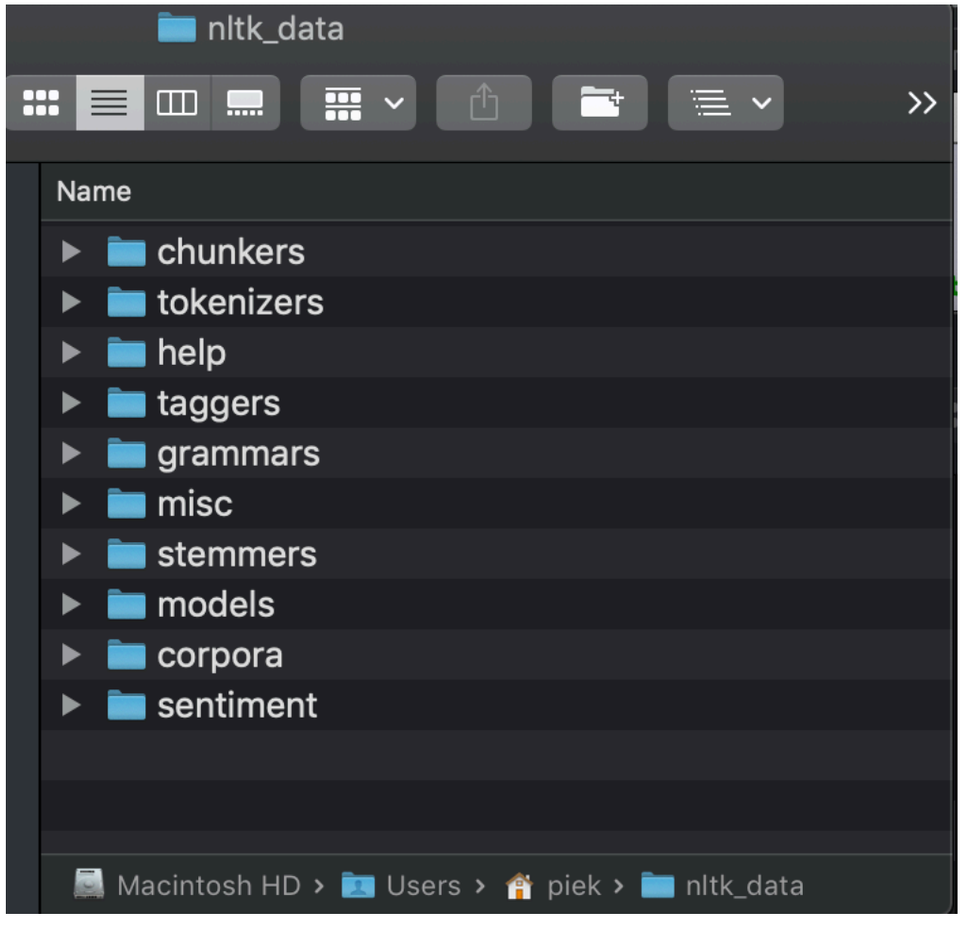
**Mailing lists**

Subscribe to get information about conferences, jobs, etc.

- linguistlist.org/
- https://mailman.uib.no/listinfo/corpora
- https://lists.uni-duesseldorf.de/mailman/listinfo/semantik
- www.siggen.org/mailing.html

# Linguistic processors

- NLTK, SpaCy package many linguistic processors for many languages
    - Sentence segmentation
    - Tokenisation
    - Lemmatisation
    - Part-of-Speech tagging
    - Chunkers, constituency parsers, dependency parsers
- For many tasks these processors are called before performing text mining
- Specific processing is needed for micro-blogs:
    - http://www.cs.cmu.edu/~ark/TweetNLP/



# Some NLP for Dutch

- STEVIN: https://ivdnt.org/taalmaterialen and CLARIAH: https://www.clariah.nl/en/
- Morpho-syntactic processing:
  - Frog: https://languagemachines.github.io/frog/
  - Alpino: https://www.let.rug.nl/vannoord/alp/Alpino/
  - SpaCy's dutch language model: https://spacy.io/models/nl
- Semantic processing of text:
  - VU-reading-machine
    - Docker image https://cloud.docker.com/u/vucltl/repository/docker/vucltl/vu-rm-pip3
    - Source code: https://github.com/cltl/vu-rm-pip3
- Resources:
  - E-Lex: https://ivdnt.org/downloads/taalmaterialen/tstc-e-lex
  - Dutch WordNet (http://wordpress.let.vupr.nl/odwn/ ) and FrameNet (http://dutchframenet.nl)
  - ANS, Algemene Nederlandse Spraakkunst: http://www.let.ru.nl/ans/