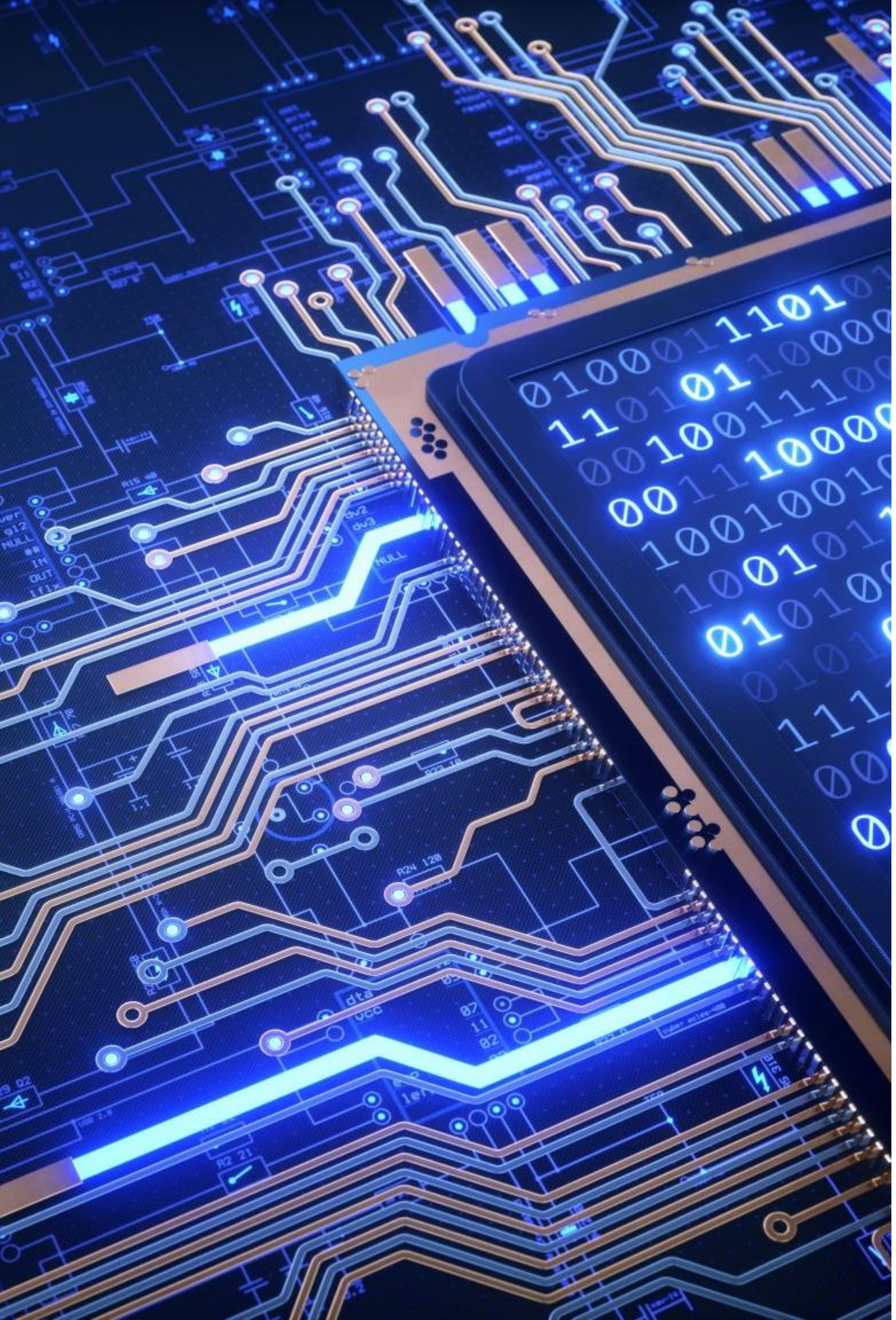
Natural Language Processing

Introduction

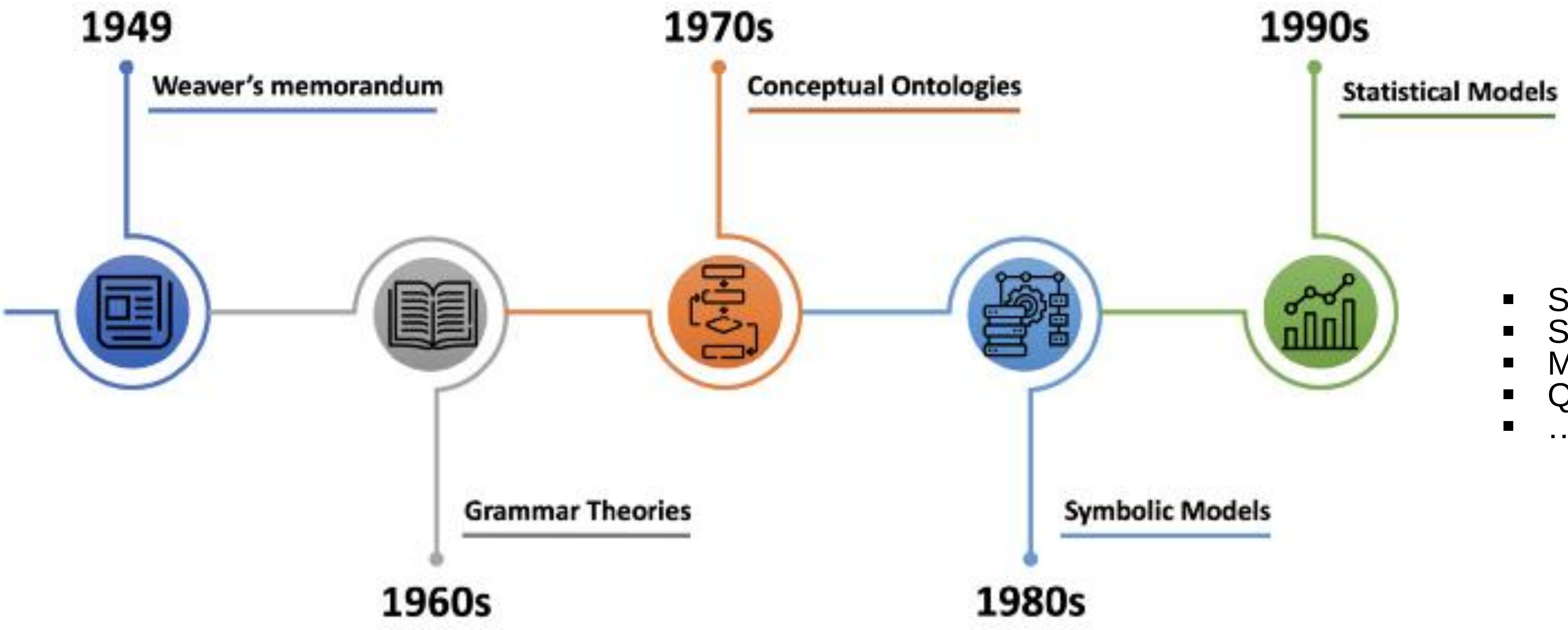
Edwin Puertas, Ph.D(c).
epuerta@utb.edu.co



What is NLP?

Natural Language Processing (NLP), or Computational Linguistics, is concerned with theoretical and practical issues in the design and implementation of computer systems for processing human languages

Brief history



Today

- Statistical methods
- Speech recognition
- MT systems
- Question-answering
- …

Aspects of language processing

Word, lexicon: lexical analysis

- Morphology, word segmentation

Syntax

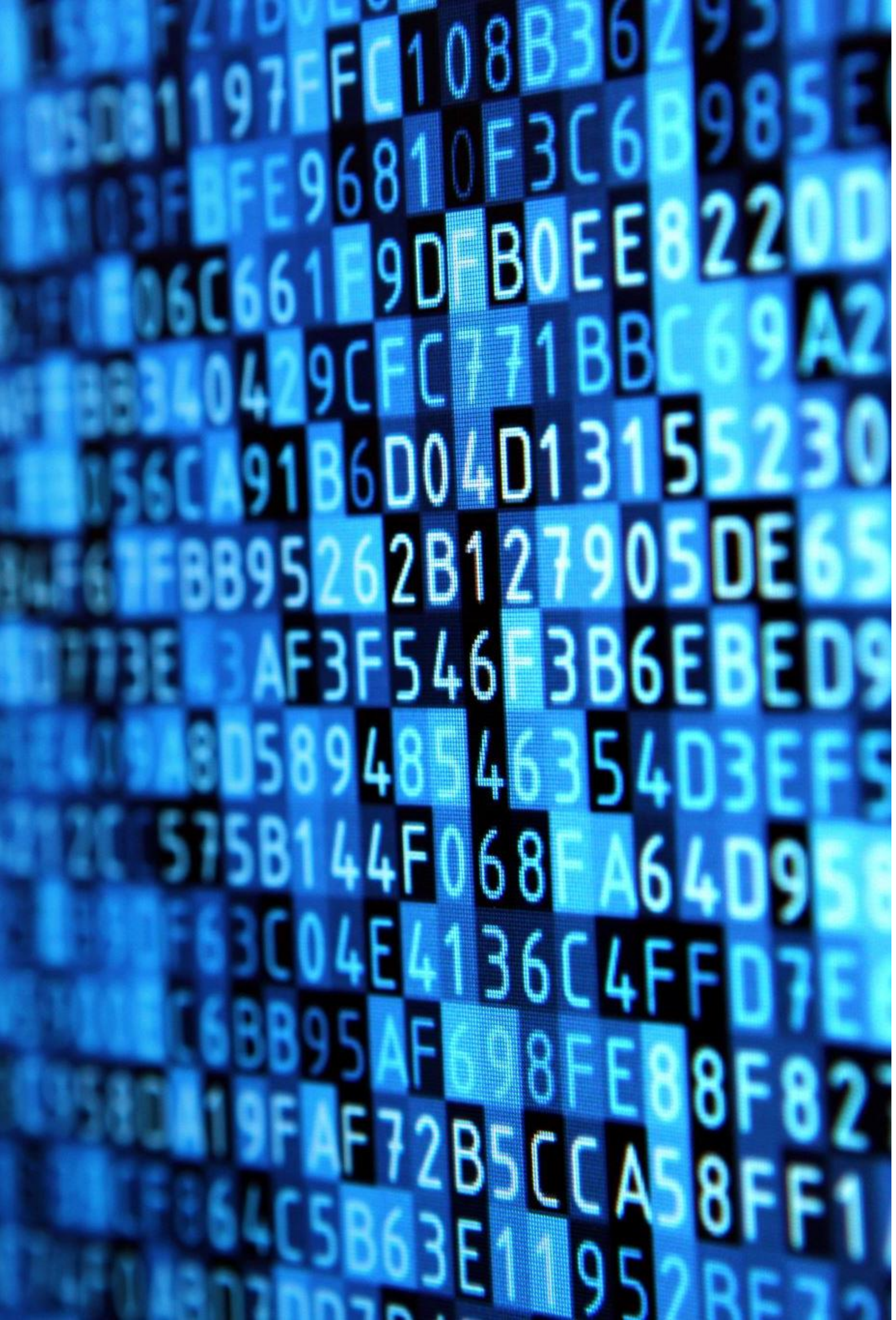
- Sentence structure, phrase, grammar, …

Semantics

- Meaning
- Execute commands

Discourse analysis

- Meaning of a text
- Relationship between sentences (e.g. anaphora)



Applications

- Detect new words
- Language learning
- Machine translation
- NL interface
- Information retrieval
- Language Translator
- Social Media Monitoring
- Chatbots
- Voice Assistants

Classical symbolic methods

- Morphological analyzer
- Parser (syntactic analysis)
- Semantic analysis (transform into a logical form, semantic network, etc.)
- Discourse analysis
- Pragmatic analysis

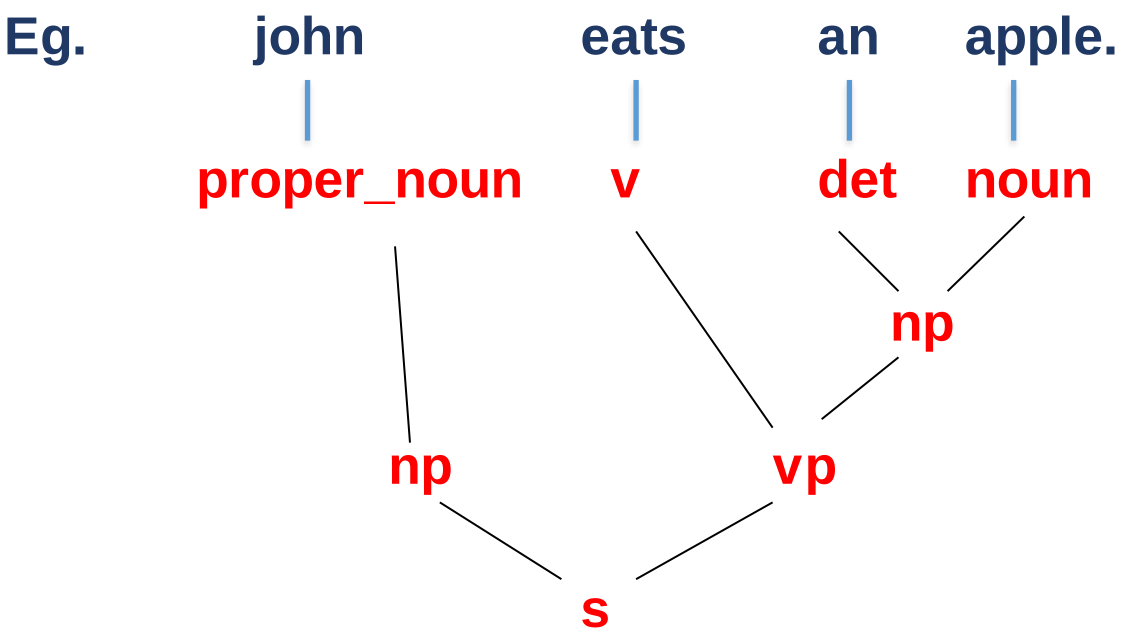
Morphological analysis

- Goal: recognize the word and category
- Using a dictionary: word + category
- Input form (*computed*)
- Morphological rules:
  - Lemma + ed -> Lemma + e (verb in past form)
  - …
- Is Lemma in dict.? If yes, the transformation is possible
- Form -> a set of possible lemmas

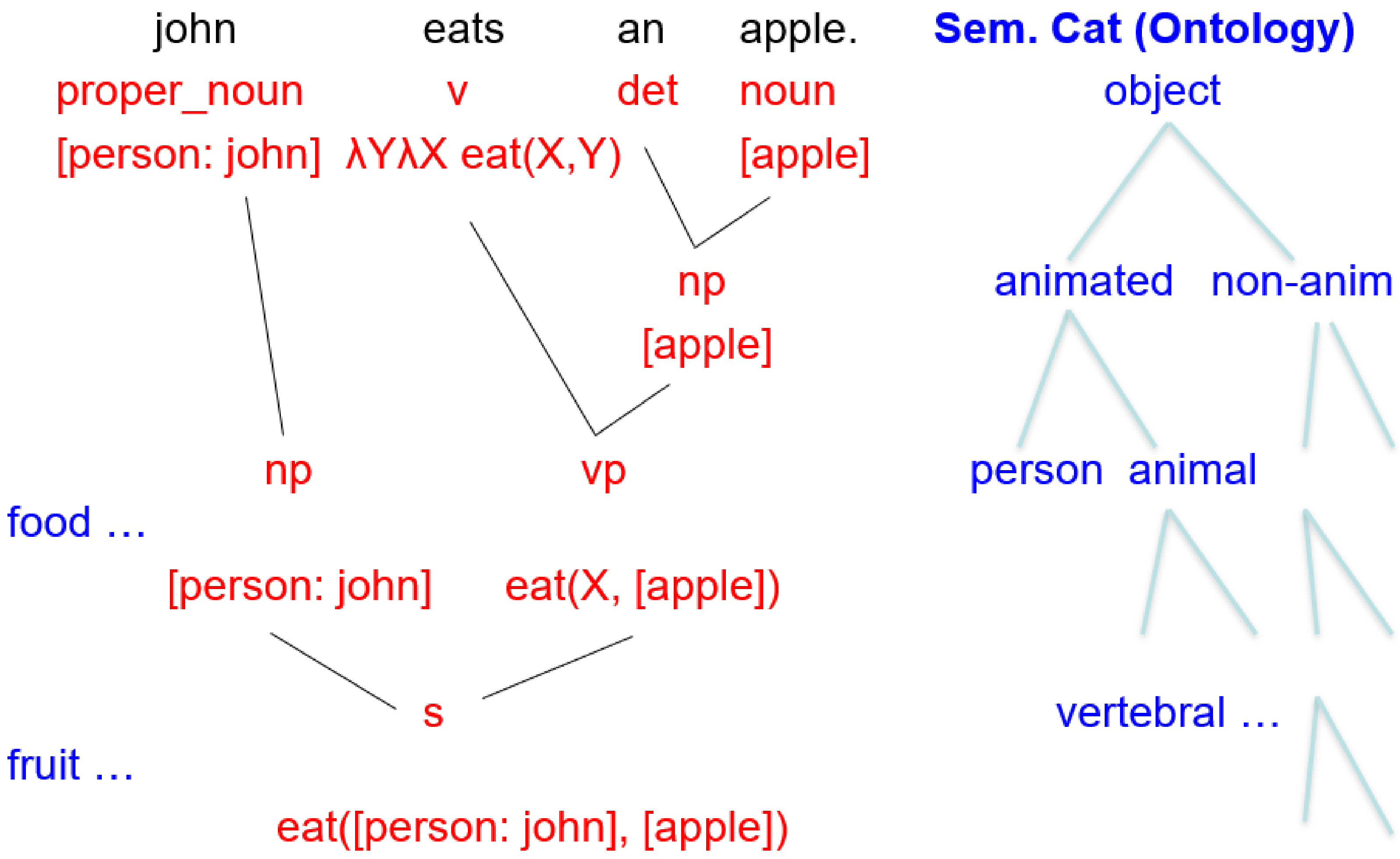
Parsing (in DCG)

```
stament --> noun_phrase , verb_phrase.
noun_phrase --> det, noun.
np --> proper_noun.
verb_phrase --> verb, ng.
verb_phrase --> verb.
```

```
det --> [an].
det --> [the].
noun --> [apple].
noun --> [orange].
proper_noun --> [john].
proper_noun --> [mary].
verb --> [eats].
verb --> [loves].
```


Eg. john eats an apple.

Semantic analysis



john eats an apple.

proper_noun v det noun

[person: john] λYλX eat(X,Y) [apple]

np

[apple]

np vp

[person: john] eat(X, [apple])

s

eat([person: john], [apple])

food …

fruit …

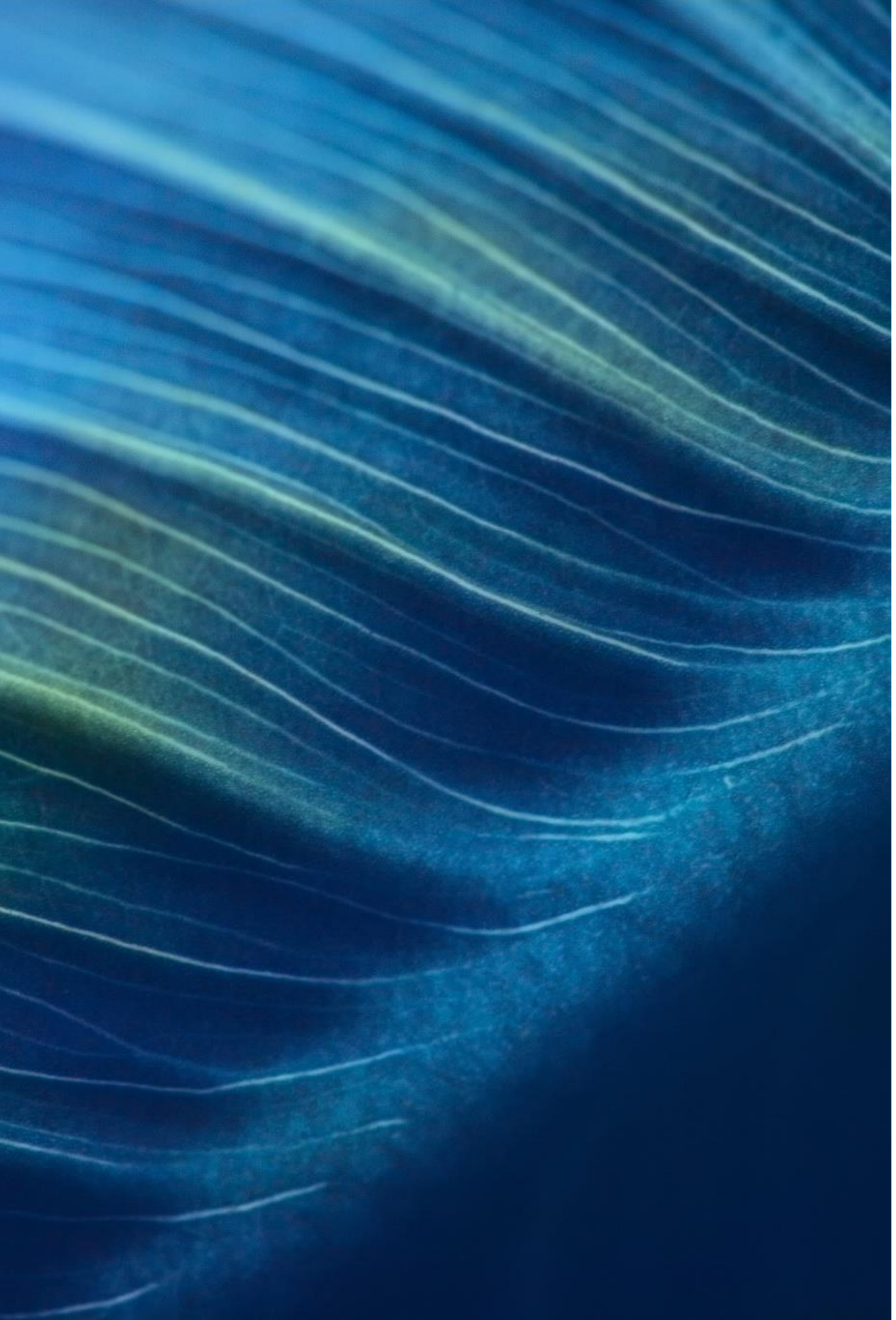
Sem. Cat (Ontology)

object

animated non-anim

person animal

vertebral …



Parsing & semantic analysis

Rules: syntactic rules or semantic rules

- What component can be combined with what component?
- What is the result of the combination?

Categories

- Syntactic categories: Verb, Noun, …
- Semantic categories: Person, Fruit, Apple, …

Analyses

- Recognize the category of an element
- See how different elements can be combined into a sentence
- Problem: The choice is often not unique

Write a semantic analysis grammar

S(pred(obj)) -> NP(obj) VP(pred)

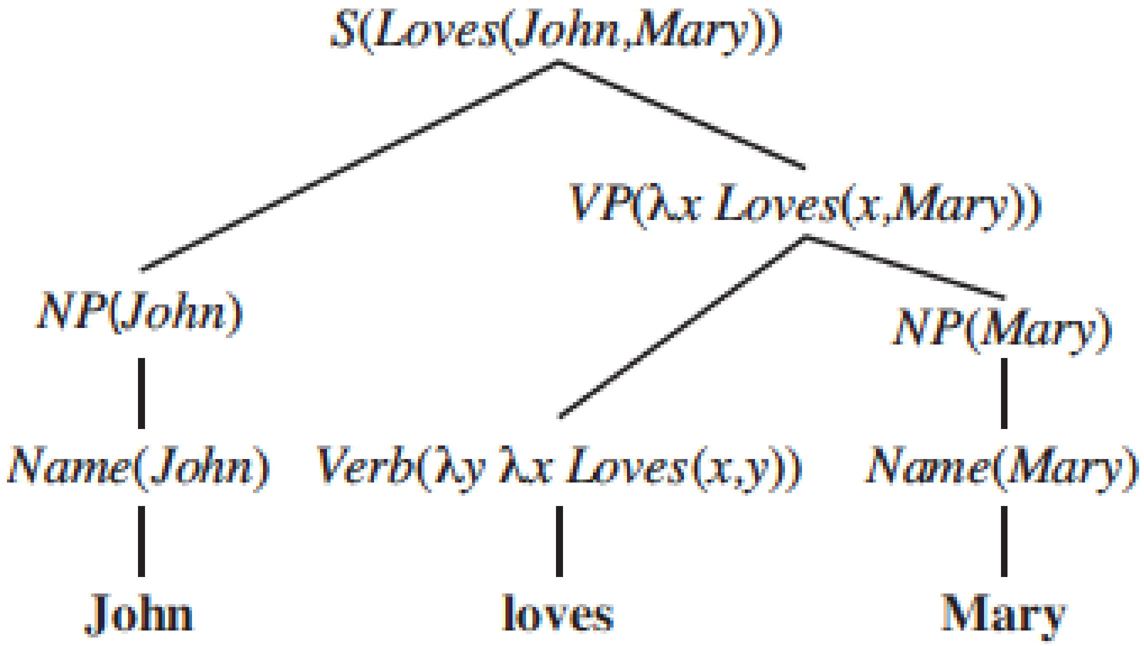
VP(pred(obj)) -> Verb(pred) NP(obj)

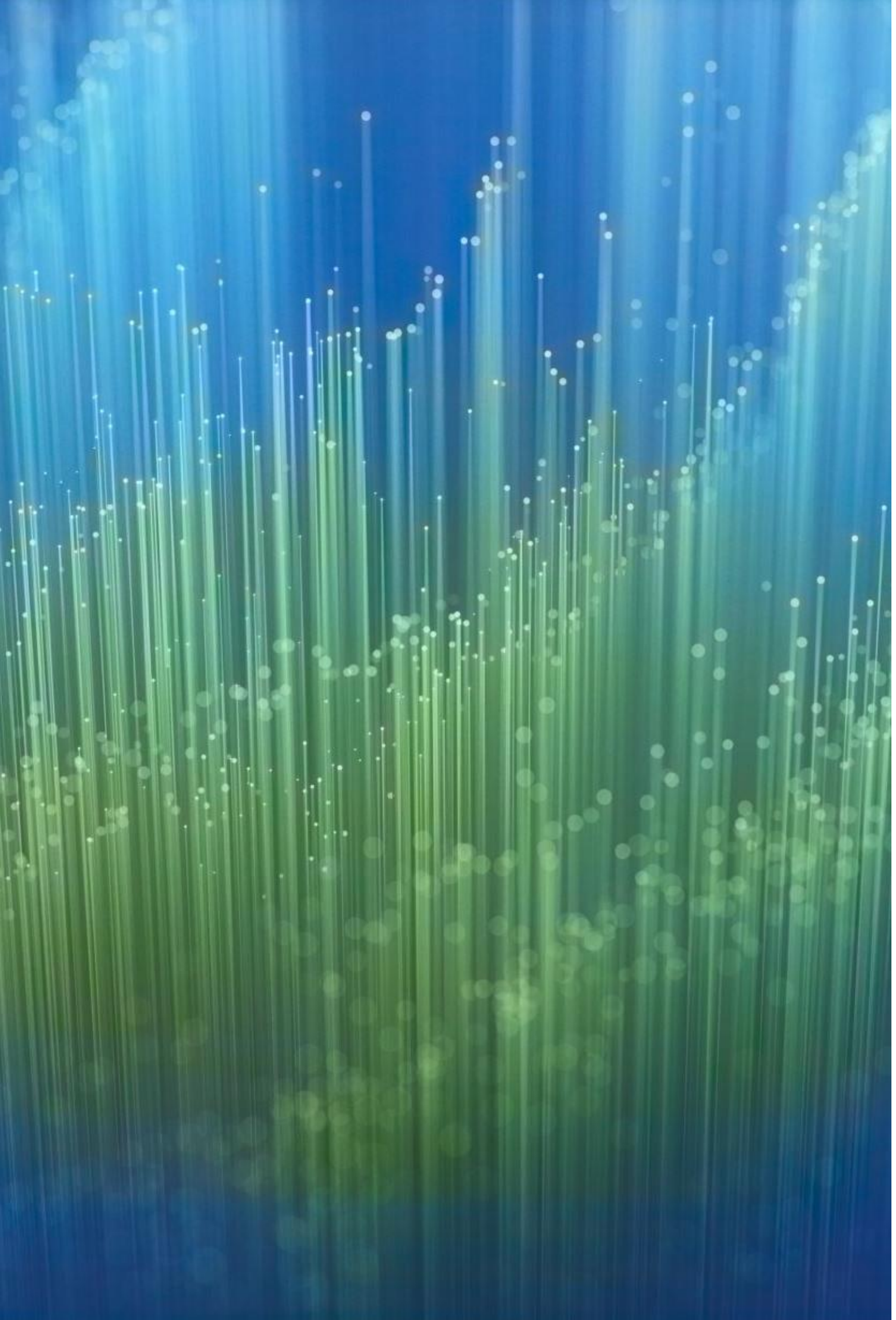
NP(obj) -> Name(obj)

Name(John) -> **John**

Name(Mary) -> **Mary**

Verb(λyλx Loves(x,y)) -> **loves**





Discourse analysis

Anaphora

He hits the car with a stone. It bounces back.

Understanding a text

Who/when/where/what … are involved in an event?

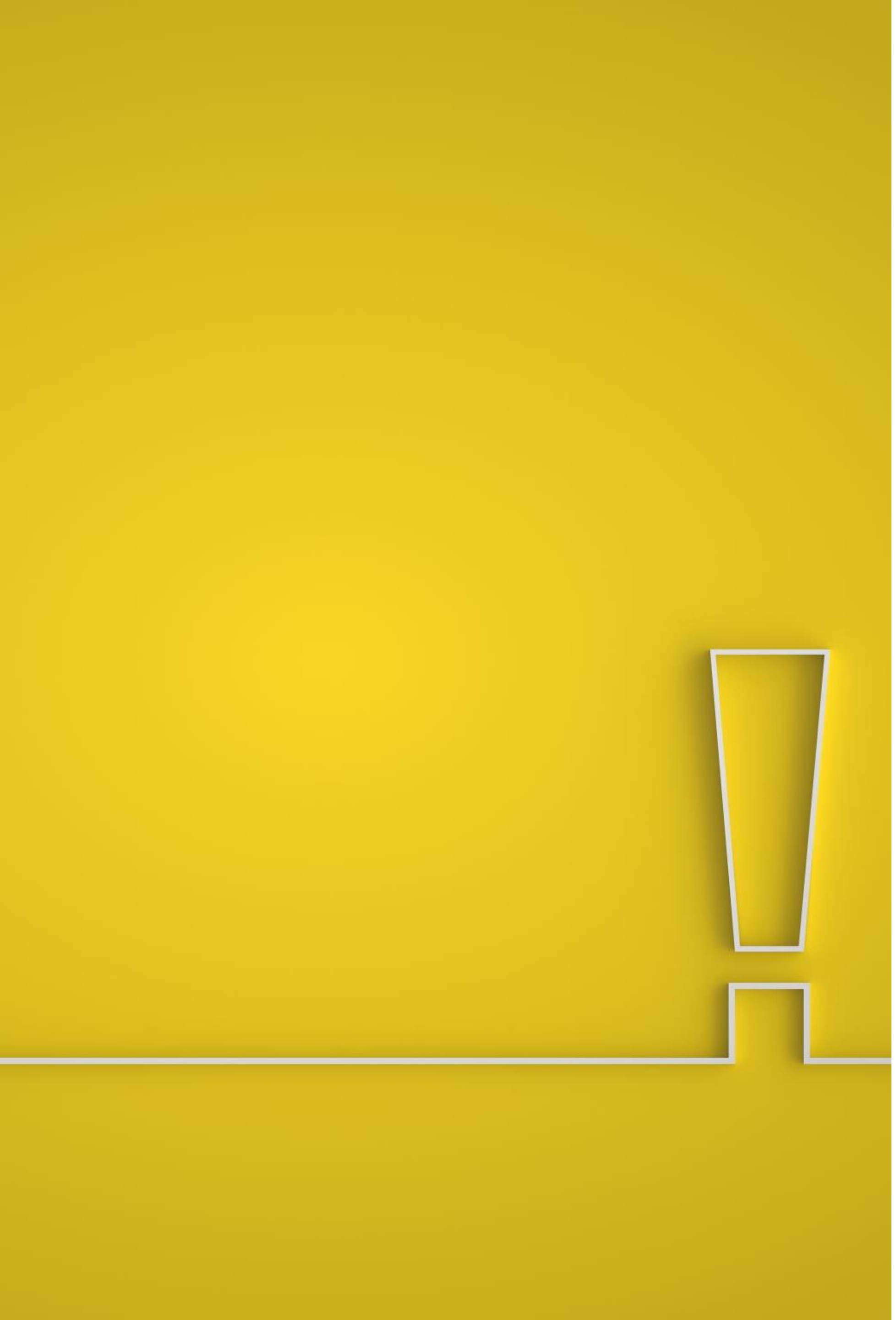
How to connect the semantic representations of different sentences?

What is the cause of an event and what is the consequence of an action?

…

Pragmatic analysis

- Practical usage of language: what a sentence means in practice
  - Do you have time?
  - How do you do?
  - It is too cold to go outside!
  - …

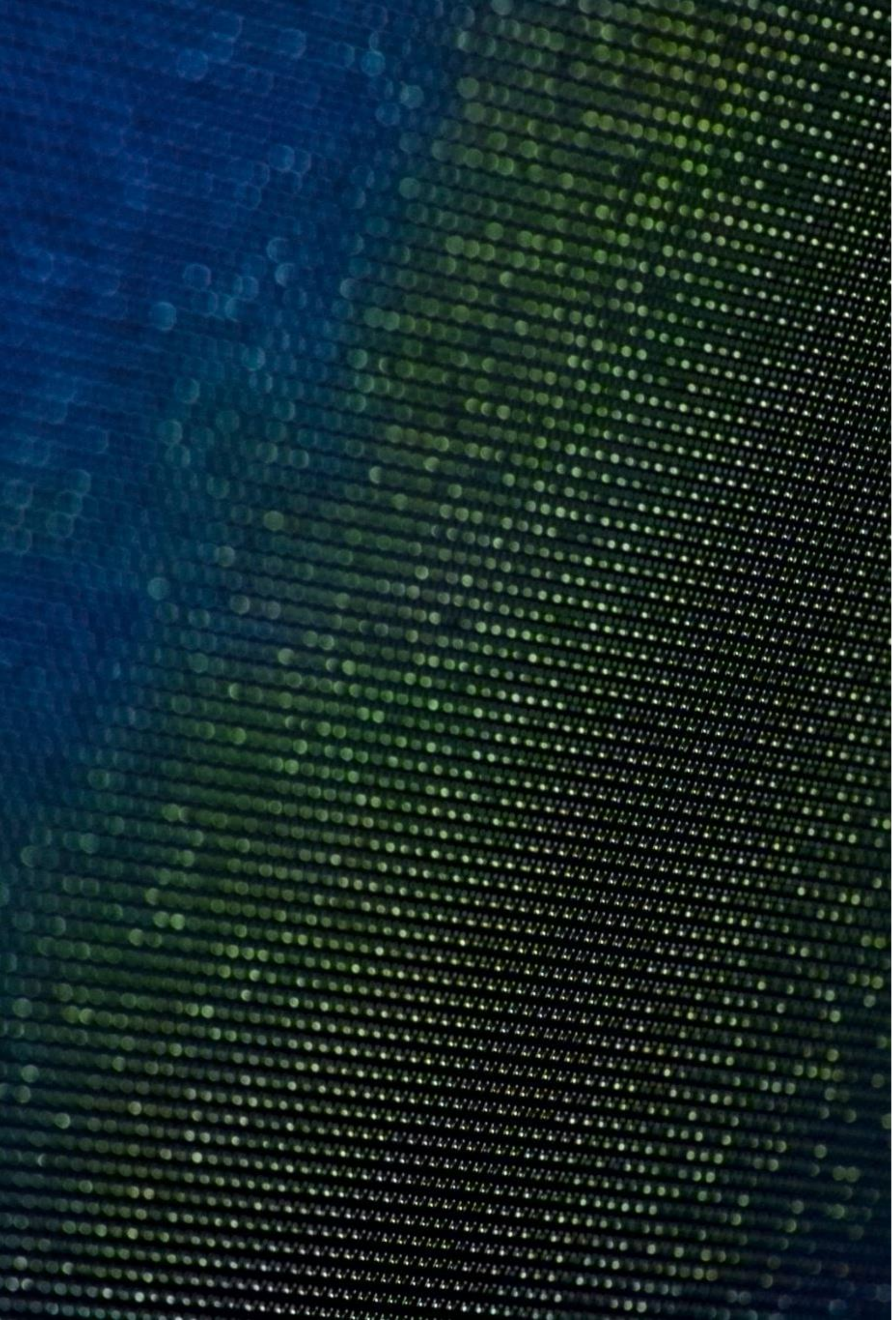


Problems

- Ambiguity
  - Lexical/morphological: change (V,N), training (V,N), even (ADJ, ADV) …
  - Syntactic: Helicopter powered by human flies
  - Semantic: He saw a man on the hill with a telescope.
  - Discourse: anaphora, …
- Classical solution
  - Using a later analysis to solve ambiguity of an earlier step
  - Eg. He gives him the change.
  - (change as verb does not work for parsing)
  - He changes the place.
  - (change as noun does not work for parsing)
  - However: He saw a man on the hill with a telescope.
    - Correct multiple parsings
    - Correct semantic interpretations -> semantic ambiguity
    - Use contextual information to disambiguate (does a sentence in the text mention that “He” holds a telescope?)

Rules vs. statistics

- Rules and categories do not fit a sentence equally
  - Some are more likely in a language than others
  - E.g.
    - hardcopy: noun or verb?
      - P(N | hardcopy) >> P(V | hardcopy)
    - the training …
      - P(N | training, Det) > P(V | training, Det)
- Idea: use statistics to help



Statistical analysis to help solve ambiguity

Choose the most likely solution

- solution* = $\text{argmax}_{\text{solution}}$ P(solution | word, context)
- e.g. $\text{argmax}_{\text{cat}}$ P(cat | word, context)
  - $\text{argmax}_{\text{sem}}$ P(sem | word, context)
- Context varies largely (precedent work, following word, category of the precedent word, …)

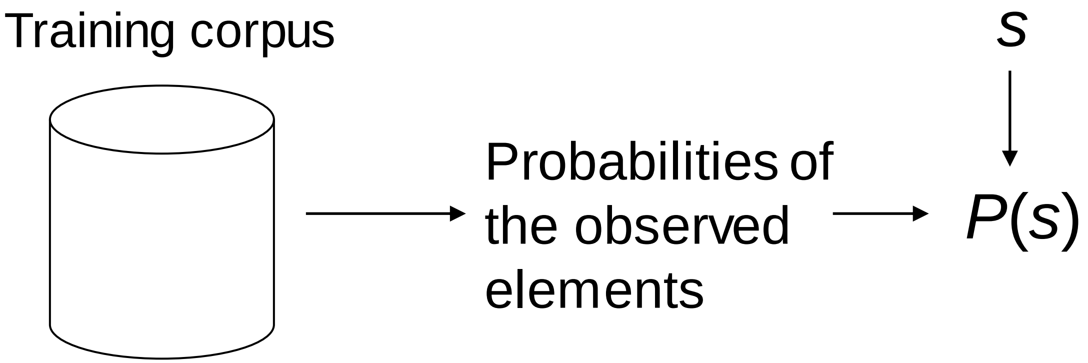
How to obtain P(solution | word, context)?

- Training corpus

Statistical language modeling

Goal: create a statistical model so that one can calculate the probability of a sequence of tokens $s = w_1, w_2, \ldots, w_n$ in a language.

General approach:



Prob. of a sequence of words

$$P(s) = P(w_1, w_2, \ldots w_n)$$

$$= P(w_1)P(w_2 \mid w_1) \ldots P(w_n \mid w_{1,n-1})$$

$$= \prod_{i=1}^{n} P(w_i \mid h_i)$$

Elements to be estimated: $P(w_i \mid h_i) = \dfrac{P(h_i w_i)}{P(h_i)}$

- If h_i is too long, one cannot observe (h_i, w_i) in the training corpus, and (h_i, w_i) is hard generalize

- Solution: limit the length of h_i

N-grams

- Limit h_i to n-1 preceding words

Most used cases

- Uni-gram: $P(s) = \prod_{i=1}^{n} P(w_i)$
- Bi-gram: $P(s) = \prod_{i=1}^{n} P(w_i \mid w_{i-1})$
- Tri-gram: $P(s) = \prod_{i=1}^{n} P(w_i \mid w_{i-2} w_{i-1})$

A simple example

(corpus = 10 000 words, 10 000 bi-grams)

w_i	$P(w_i)$	w_{i-1}	$w_{i-1}w_i$	$P(w_i\|w_{i-1})$
I (10)	10/10 000 = 0.001	# (1000)	(# I) (8)	8/1000 = 0.008
		that (10)	(that I) (2)	0.2
talk (8)	0.0008	I (10)	(I talk) (2)	0.2
		we (10)	(we talk) (1)	0.1
		…		
talks (8)	0.0008	he (5)	(he talks) (2)	0.4
		she (5)	(she talks) (2)	0.4
		…		
she (5)	0.0005	says (4)	(she says) (2)	0.5
		laughs (2)	(she laughs) (1)	0.5
		listens (2)	(she listens) (2)	1.0

Uni-gram: P(I, talk) = P(I) * P(talk) = 0.001*0.0008
P(I, talks) = P(I) * P(talks) = 0.001*0.0008

Bi-gram: P(I, talk) = P(I | #) * P(talk | I) = 0.008*0.2
P(I, talks) = P(I | #) * P(talks | I) = 0.008*0

Estimation

- History: short long
- modeling: coarse refined
- Estimation: easy difficult
- Maximum likelihood estimation MLE

$$P(w_i) = \frac{\#(w_i)}{|C_{uni}|} \quad P(h_i w_i) = \frac{\#(h_i w_i)}{|C_{n-gram}|}$$

- If (hi mi) is not observed in training corpus, P(wi|hi)=0
- P(they, talk)=P(they|#) P(talk|they) = 0
- never observed (they talk) in training data
- smoothing

Examples of utilization

Predict the next word

- $\text{argmax}_w\ P(w \mid \text{previous words})$

Used in input (predict the next letter/word on cellphone)

Use in machine aided human translation

- Source sentence
- Already translated part
- Predict the next translation word or phrase
- $\text{argmax}_w\ P(w \mid \text{previous trans. words, source sent.})$

Quality of a statistical language model

- Test a trained model on a test collection
  - Try to predict each word
  - The more precisely a model can predict the words, the better is the model
- Perplexity (the lower, the better)
  - Given $P(w_i)$ and a test text of length N

$$Perplexity = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(w_i)}$$

  - Harmonic mean of probability
  - At each word, how many choices does the model propose?
    - Perplexity=32 ~ 32 words could fit this position

State of the art

- Sufficient training data
  - The longer is n (n-gram), the lower is perplexity
- Limited data
  - When n is too large, perplexity decreases
  - Data sparseness (sparsity)
- In many NLP researches, one uses 5-grams or 6-grams
- Google books n-gram (up to 5-grams)https://books.google.com/ngrams

More than predicting words

Speech recognition

- Training corpus = signals + words
- probabilities: P(signal|word), P(word2|word1)
- Utilization: signals sequence of words

Statistical tagging

- Training corpus = words + tags (n, v)
- Probabilities: P(word|tag), P(tag2|tag1)
- Utilization: sentence sequence of tags

Example of utilization

Speech recognition (simplified)

$$\begin{aligned} &\text{argmax}_{w1,\ \ldots,\ wn}\ P(w_1,\ \ldots,\ w_n|s_1,\ \ldots,\ s_n) \\ &= \text{argmax}_{w1,\ \ldots,\ wn}\ P(s_1,\ \ldots,\ s_n|w_1,\ \ldots,\ w_n) * P(w_1,\ \ldots,\ w_n) \\ &= \text{argmax}_{w1,\ \ldots,\ wn}\ \Pi_I\, P(s_i|w_1,\ \ldots,\ w_n)*P(w_i|w_{i-1}) \\ &= \text{argmax}_{w1,\ \ldots,\ wn}\ \Pi_I\, P(s_i|w_i)*P(w_i|w_{i-1}) \end{aligned}$$

Argmax - Viterbi search

- probabilities:
  - P(signal|word),
    - P(*** | ice-cream)=P(*** | I scream)=0.8;
  - P(word2 | word1)
    - P(ice-cream | eat) > P(I scream | eat)
- Input speech signals s_1, s_2, ..., s_n
  - I eat ice-cream. > I eat I scream.

Example of utilization

Statistical tagging

- Training corpus = word + tag (e.g. Penn Tree Bank)
- For $w_1, \ldots, w_n$:
  - $\text{argmax}_{\text{tag1, …, tagn}}\ \Pi_I\, P(w_i|\text{tag}_i)*P(\text{tag}_i|\text{tag}_{i-1})$
- probabilities:
  - P(word|tag)
    - P(change|noun)=0.01, P(change|verb)=0.015;
  - P(tag2|tag1)
    - P(noun|det) >> P(verb|det)
- Input words: $w_1, \ldots, w_n$
  - I give him the change.
    - pronoun verb pronoun det noun >
    - pronoun verb pronoun det verb

Some improvements of the model

- Class model
  - Instead of estimating P(w2|w1), estimate P(w2|Class1)
  - P(me|take) v.s. P(me|Verb)
  - More general model
  - Less data sparseness problem
- Skip model
  - Instead of $P(w_i|w_{i-1})$, allow $P(w_i|w_{i-k})$
  - Allow to consider longer dependence

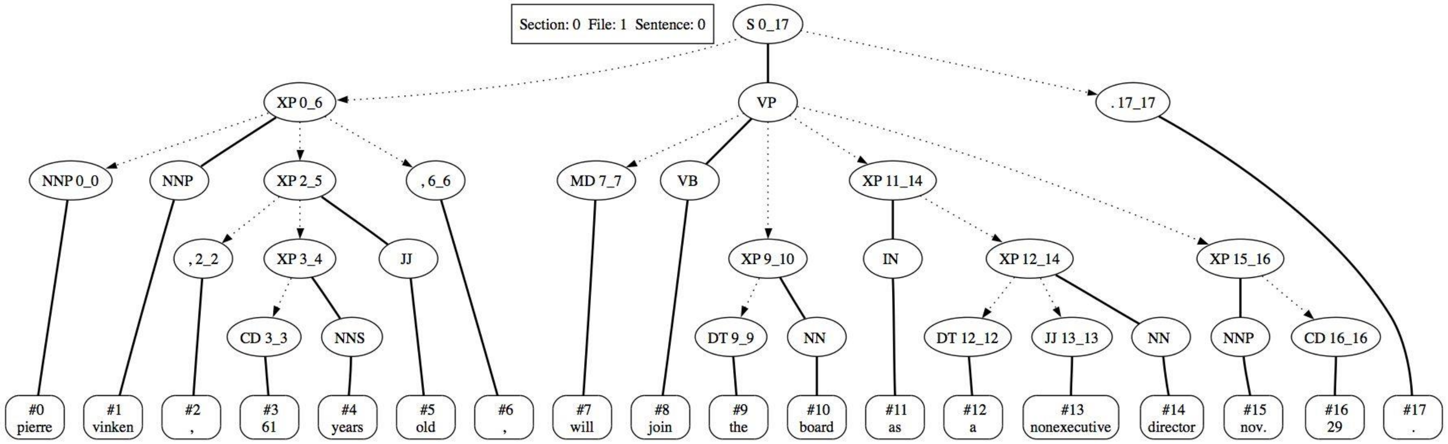
State of the art on POS-tagging

- POS = Part of speech (syntactic category)
- Statistical methods
- Training based on annotated corpus (text with tags annotated manually)
  - Penn Treebank: a set of texts with manual annotations http://www.cis.upenn.edu/~treebank/

Penn Treebank

One can learn:

- $P(w_i)$
- $P(Tag \mid w_i)$, $P(w_i \mid Tag)$
- $P(Tag_2 \mid Tag_1)$, $P(Tag_3 \mid Tag_1, Tag_2)$
- …



50 AÑOS UTB

Universidad Tecnológica de Bolívar

CARTAGENA DE INDIAS

Programa de Ingeniería de Sistemas y Computación