Text classification

François Coste

SML, Master SIF

2021-2022

Outline

Outline

Natural language sequences

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light'st flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel.
Thou that art now the world's fresh ornament
And only herald to the gaudy spring,
Within thine own bud buriest thy content
And, tender churl, makest waste in niggarding.
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

Shakespeare's Sonnet 1 (douze vers en trois quatrains et un distique)

DNA sequence

```
tgtaaatataaattgtagcataaaataacaccaagcttgtttttaaacattcactcaaaa
tacattgaacagctactgtgcagtgttgggactgcttagaaacttaagacaaaatactgg
gcgtgatagtgtgtgcctgtagtcccagctactcaagacgcctaggcaagaggatagctt
gaattcaagagtccaagtgcagcctgggcaacatagtgagattcagtctttttttttttt
ttttctgaaatggagtctcactctgtggcccaggctggagggcagtggcacaatcttgcc
tcactgcaacctccacctcctgggttcaaatgattctcctgcctcatcctcctgagtaac
taggattacaggcatgcaccaccacactcagctgatttcatatttttagtagagatgggg
tttcaccatgttggccaggctggtctcgaattcctgacctcaagtgatccacccaccttg
gcctcccaaagtgccgggatcacaggcatgagccaccatgcccagccaagacccagtctt
taaaataatgacataaaataaacacataaacctcaaacaaacaaaacacaaactatgatt
tcattattaattcaggtgtttatacaatgctctctgagaactcagttttaaagattaccc
aaagaatccctgcgacaacccagcctaaaatgatacaaaatcagttttaatataactgaa
gtttttagattctgactctaaatattagattgtgacattttacctttctctgactttatg
ccaaatttacttcaacttctaacatcgatcgttctcatttatgaacaaatatgtatttca
caaatgtggaaatacaagcatgagccaaacaaaaccaaatcctggacttatggagctgat
ggtaaaaaattccccaacataaccactcgattacagttttatgcaatgtgataagagtaa
gggaccactgtagggaggtcagaagttgcctctcagaggtggagactacagctgttgtca
```


Protein sequences

```
>GLPF_BACSU
MTAFWGEVIGTMLLIIFGAGVCAGVNLKKSLSFQSGWIVVVFGWGLGVAMAAYAVGGISGAHLNPALTIALA
FVGDFPWKEVPVYIAAQMIGAIIGAVIIYLHYLPHWKSTDDPAAKLGVFSTGPSIPHTFANVLSEVIGTFVL
VLGILAIGANQFTEGLNPLIVGFLIVAIGISLGGTTGYAINPARDLGPRIAHAFLPIPGKGSSNWKYAWVPV
VGPILGGSFGGVFYNAAFKGHITSSFWIVSVILVVVLLGLYVYTKSHSAKTLSNSKYI
>GLPF_ECOLI
MSQTSTLKGQCIAEFLGTGLLIFFGVGCVAALKVAGASFGQWEISVIWGLGVAMAIYLTAGVSGAHLNPAVT
IALWLFACFDKRKVIPFIVSQVAGAFCAAALVYGLYYNLFFDFEQTHHIVRGSVESVDLAGTFSTYPNPHIN
FVQAFAVEMVITAILMGLILALTDDGNGVPRGPLAPLLIGLLIAVIGASMGPLTGFAMNPARDFGPKVFAWL
AGWGNVAFTGGRDIPYFLVPLFGPIVGAIVGAFAYRKLIGRHLPCDICVVEEKETTTPSEQKASL
>AqpZe
MFRKLAAECFGTFWLVFGGCGSAVLAAGFPELGIGFAGVALAFGLTVLTMAFAVGHISGGHFNPAVTIGLWA
GGRFPAKEVVGYVIAQVVGGIVAAALLYLIASGKTGFDAAASGFASNGYGEHSPGGYSMLSALVVELVLSAG
FLLVIHGATDKFAPAGFAPIAIGLALTLIHLISIPVTNTSVNPARSTAVAIFQGGWALEQLWFFWVVPIVGG
IIGGLIYRTLLEKRD
>AQP1h
MASEFKKKLFWRAVVAEFLATTLFVFISIGSALGFKYPVGNNQTAVQDNVKVSLAFGLSIATLAQSVGHISG
AHLNPAVTLGLLLSCQISIFRALMYIIAQCVGAIVATAILSGITSSLTGNSLGRNDLADGVNSGQGLGIEII
GTLQLVLCVLATTDRRRRDLGGSAPLAIGLSVALGHLLAIDYTGCGINPARSFGSAVITHNFSNHWIFWVGP
FIGGALAVLIYDFILAPRSSDLTDRVKVWTSGQVEEYDLDADDINSRVEMKPK
```


E-bank customer transactions (timestamped)

Cus1: (4200,married,tech,24) - <(2,Friday) TM,CD> <(4,Sunday) WM> <(20,Saturday) RD,WM,TM>

Cus2: (4000,married,tech,22) - <(3,Tuesday) TM,CD,WM> <(7,Sunday) WM,CD> <(20,Saturday) RD,WM> <(1,Tuesday) TM,CD>

Cus3: (1500,single,retired,70)] - <(3,Monday) CD,TM,WM> <(10,Monday) CD,TM,WM> <(16,Sunday) WM>

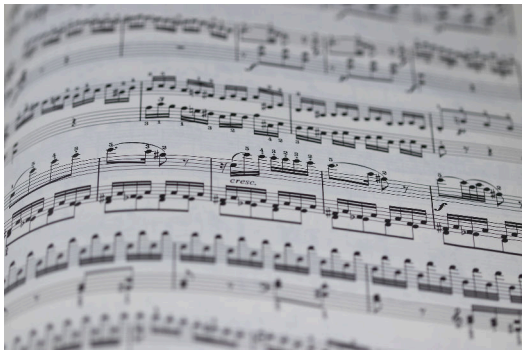
SD: receive money ; TM: transfer ; WM: withdraw money ; CD: create time deposit ; RD: cancel deposit

Source : R. Quiniou

tcdump of server scan Source : http://www.linuxfocus.org/Francais/May2003/article292.shtml

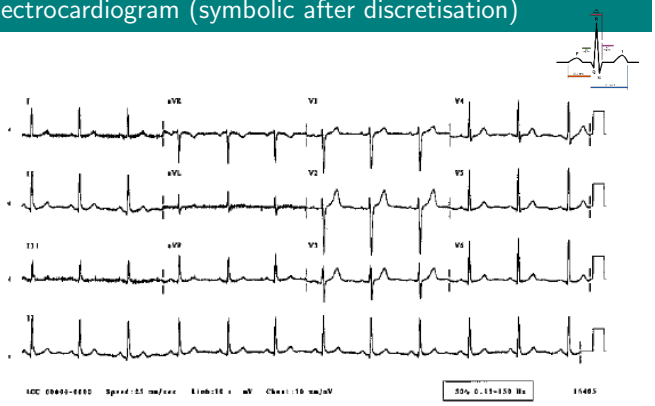
```
10:43:37.594397 Diablo > Diablo: icmp: echo request
                        4500 001c 2ecf 0000 2c01 6210 7f00 0001
                        7f00 0001 0800 8f87 6878 0000
10:43:37.594397 Diablo > Diablo: icmp: echo reply (DF)
                        4500 001c 0000 4000 ff01 7dde 7f00 0001
                        7f00 0001 0000 9787 6878 0000
10:43:37.604397 Diablo.34607 > Diablo.http: . ack 1932747046 win 4096
                        4500 0028 ee0f 0000 3706 97be 7f00 0001
                        7f00 0001 872f 0050 5b20 0003 7333 6126
                        5010 1000 ead5 0000
10:43:37.604397 Diablo.http > Diablo.34607: R 1932747046:1932747046(0) win 0 (DF)
                        4500 0028 0000 4000 ff06 7dcd 7f00 0001
                        7f00 0001 0050 872f 7333 6126 0000 0000
                        5004 0000 5605 0000
10:43:37.904397 Diablo.34587 > Diablo.408: . win 4096
                        4500 0028 e3bb 0000 3706 a212 7f00 0001
                        7f00 0001 871b 0198 0000 0000 0000 0000
                        5000 1000 192f 0000
10:43:37.904397 Diablo.408 > Diablo.34587: R 0:0(0) ack 0 win 0 (DF)
                        4500 0028 0000 4000 ff06 7dcd 7f00 0001
                        7f00 0001 0198 871b 0000 0000 0000 0000
                        5014 0000 291b 0000
```

Music



source: Image by HeungSoon from Pixabay

Electrocardiogram (symbolic after discretisation)

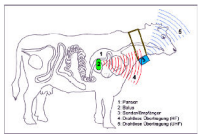


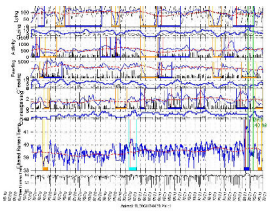
Source : E. Vidal

Multi sequences (symbolic after discretisation)

Calves monitoring from sensor data

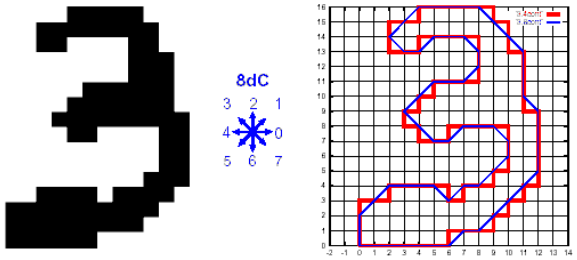
- Bolus sensor in calf's rumen
  - Raw temperature
  - Filtered temperature: without drinking effects
  - Drink counter
  - Drink amount
- Accelerometer
  - Step count
  - Lying count
  - Lying duration
  - Feeding count
  - Feeding duration





Source : R. Quiniou

Picture contour (sequential after recoding)



8dC: 00007776667666655554544444321100071011234454331100123445431 1

Source : E. Vidal

Semi-structured sequences

Document structure

```
<book>
  <part>
    <chapter>
      <sect1/>
      <sect1>
        <orderedlist numeration="arabic">
          <listitem/>
          <f:fragbody/>
        </orderedlist>
      </sect1>
    </chapter>
  </part>
</book>
```

Source : C. de la Higuera

Semi-structured sequences

XML

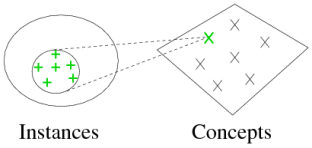
```
<?xml version="1.0"?>
<sample>
  <title>Sample XML File</title>
  <description>
    This is a sample XML document that you can use to test
    the sample1.exe test program.
  </description>
  <usage>
    sample1 sample.xml
  </usage>
</sample>
```

Source : C. de la Higuera

Outline

An inductive learning problem

Learning a concept from examples



Important choices (representation biases) :

- Instance space
- Hypothesis space

Description of sequences

- Attribute-value vector representations

  ⇝ use classical machine learning algorithms: Naive Bayes, decision trees,...

  x_i: n *attributes ("vectorization" of each sequence)*

  $$x_i = \{A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n\}$$

Outline

Outline

Example

Goal

Learn concept "well-to-do people" (: income > mean income)

- Objects to classify $o_i \in O$ — O : *French people*
- Description of o_i: $x_i \in X$ — $X = \{, \neg\}$
- Class of o_i: $c_i \in C$ — $C = \{, \neg\}$
- "Representative" sample S

40% of people are , 80% of people have a smartphone, else only 45% of ¬ people have a smartphone.

Class		¬
P(c)	0.4	O.6
P(|c)	0.8	0.45

Classification rules

Class		¬
P(c)	0.4	O.6
P(|c)	0.8	0.45

Majority:

$$h_{\text{maj}} = \text{argmax}_{c \in C}\, P(c)$$

$\forall x, h_{maj}(x) =$ ¬

Maximum Likelihood:

$$h_{\text{ML}} = \text{argmax}_{c \in C}\, P(x|c)$$

returns class in which description has the highest probability

$h_{ML}() =$

$h_{ML}(\neg) =$ ¬

Classification task in working domains $(Telecom, Health, Education)$ with $P(|\text{Telecom}) = 1 \Rightarrow h_{\text{ML}}() = \text{Telecom}$

Classification rules

Class		¬
P(c)	0.4	O.6
P(|c)	0.8	0.45

Bayes:

$$h_{\text{Bayes}} = \text{argmax}_{c \in C}\, P(c|x) = \text{argmax}_{c \in C}\, P(x|c) \times P(c)$$

returns most probable class for description (*maximum a posteriori*)

(Bayes Formula: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$ and $P(x)$ is independent from c)

$P(|) \times P() = 0.8 \times O.4 = \mathbf{0.32}$

$P(|\neg) \times P(\neg) = 0.45 \times 0.6 = 0.27$ $\quad h_{ML}() =$

$P(\neg|) \times P() = 0.2 \times 0.4 = 0.08$

$P(\neg|\neg) \times P(\neg) = 0.55 \times 0.6 = \mathbf{0.33}$ $\quad h_{ML}(\neg) =$ ¬

Optimal... if we know $P(x|c)$ and $P(c)$!

Naive Bayes

Approximation of $P(d|c)$ and $P(c)$

Let $D = A_1 \times A_2 \times \ldots \times A_m$ (m attributes)

$$h_{\text{Bayes}} = \underset{c \in C}{\text{argmax}}\, P(\langle a_1, a_2, \ldots, a_m \rangle | c) \times P(c)$$

Naive Bayes conditional independence assumption:

$$P(\langle a_1, a_2, \ldots, a_m \rangle | c) = \prod_{i \in [1,m]} P(a_i|c)$$

Naive Bayes

$$h_{\text{NB}} = \text{argmax}_{c \in C} \prod_{i \in [1,m]} P(a_i|c) \times P(c)$$

"False" assumption but very efficient!

Outline

Example
(adapted from François Denis)

Who wrote this fable: La Fontaine or Esope?

- Corpus: 100 first words of the fables of La Fontaine and Esope
- Attribute-value description
  A_i: i-th word of the fable
  La cigale ayant chanté tout l'été...
  → A_1 =*la*,A_2=*cigale*, ...
- One can use Part-of-Speech preprocessing
  La pianiste donne le la à la cantatrice.
  → A_1 =*article(la, fem, sing)*, A_2=*noun(pianiste, fem, sing)*, ...

Conditional Independence assumption ?

- Probability of a sentence is determined only by the probabilities of observing the words at their position
  $$P(\textit{La cigale ayant} \ldots | \textit{Esope}) = P(A_1 = \textit{la} | Esope) \times P(A_2 = \textit{cigale} | Esope) \times \ldots \text{ !!!}$$
- Naive Bayes works very well!
  as long as conditional independence assumption does not penalize one class more than others, i.e. if:

$$\frac{P(x|c_1)}{P(x|c_2)} \sim \prod_{i=1}^{m} \frac{P(a_i|c_1)}{P(a_i|c_2)}$$

Bag of words

Assumption: probability of a word is independent of the position

$$P(\textit{La cigale ayant} \ldots | \textit{Esope}) = P(\textit{la} | Esope) \times P(\textit{cigale} | Esope) \times \ldots$$

→ *Bag of words* representation

La pianiste donne le la à la cantatrice
→ {*à, cantatrice, donne, la, la, la, le, pianiste*}

La France bat la Croatie en finale
→ {*bat, Croatie, en, finale, France, la, la*}
La Croatie bat la France en finale
→ {*bat, Croatie, en, finale, France, la, la*}

Naive Bayes on Bag of Words

Classification of a sentence $s = w_1 \ldots w_l$ of length l

$$h_{\text{NB BOW}} = \underset{c \in C}{\operatorname{argmax}} \prod_{i=1}^{l} \hat{P}(w_i|c)\hat{P}(c)$$

where $\hat{P}(w_i|c)$ and $\hat{P}(c)$ are estimator of $P(w_i|c)$ and $P(c)$
A simple estimation:

$$\hat{P}(c) = \frac{|c|}{|S|}$$

where $|S|$: number of sentences in training sample S
and $|c|$: number of sentence of class c in S

$$\hat{P}(w|c) = \frac{TF(w,c)}{\Sigma_{w'} TF(w',c)}$$

where TF (*Term Frequency*): occurrence count in S of w of class c

Smoothing

What if none of the training sentences of class c_j contain word w_i ?

$$\hat{P}(w_i|c_j) = 0, \text{ and } \ldots$$

$$\prod_{i=1}^{l} \hat{P}(w_i|c)\hat{P}(c) = 0$$

Solution add pseudo count of 1 for each unseen word (Laplace estimator):

$$\hat{P}(w|c) = \frac{TF(w,c) + 1}{\Sigma_{w'} TF(w',c) + |words|}$$

More generally:

$$\hat{P}(w|c) = \frac{TF(w,c) + mp}{\Sigma_{w'} TF(w',c) + m}$$

where: p prior estimate for $\hat{P}(w|c)$
and m weight given to prior (i.e. number of "virtual" examples)

Naive Bayes on Bag of Words

An ongoing surprise and disapointment is that structurally simple representation produced without linguistic or domain knowledge have been as effective as many others [Lewis 98]

Outline

Notebook:

http://people.rennes.inria.fr/Francois.Coste/sml-files/MLforTextClassification-skeleton.ipynb

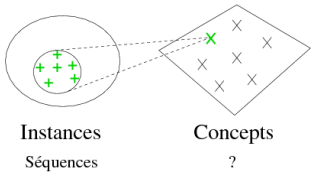
Description of sequences

- Attribute-value vector representations
  ⇝ use classical machine learning algorithms: Naive Bayes, decision trees,...
  x_i: n *attributes ("vectorization" of each sequence)*
  $$x_i = \{A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n\}$$
- Keep sequence information
  x_i: *one sequence of symbols*
  $$x_i = s_1 s_2 \ldots$$

Outline

An inductive learning problem

Learning a concept from sequences



Important choices:

- Instance space
- Hypothesis space

Concepts on sequences

Representation of a set of sequences:

- Enumeration (for small sets of sequences!)
- "Mathematical" characterizations (sequences properties)
  Set descriptions, logical description, pattern...
- Grammars (generative)
  Regular (automata), context-free, context-sensitive, unrestricted.
  Categorial, rewriting systems ...
- Stochastic models (generative + weight of sequence wrt concept)
  n-grams, PA (probabilistic automata), HMM (hidden Markov models),
  SCFG (stochastic context-free grammars)...
- SVM based on string kernels (sequence classification)
- Recurrent neural networks (sequence classification and generation)
  RNN, RAAM, LSTMs ...

White box vs black box?
Expressivity?

White box: explicit models

"A grammar of English"

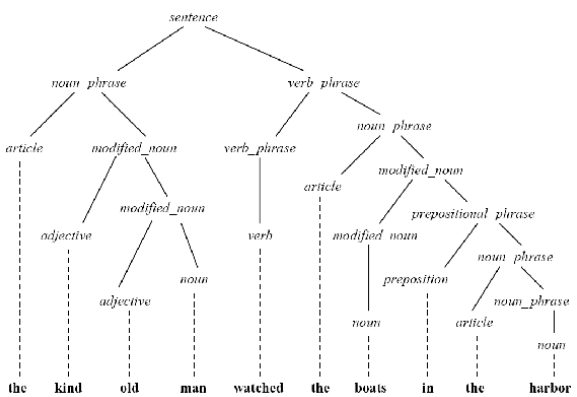
source: D.B Searls

Sentence → Noun-Phrase Verb-Phrase
Noun-Phrase → Article Modified-Noun
Verb-Phrase → Verb Noun-Phrase
Modified-Noun → Noun | Adjective Modified-Noun
Noun → **linguist** | **biologist**
Verb → **sees** | **believes**
Adjective → **young** | **famous**
Article → **a** | **the**

Generated sentences

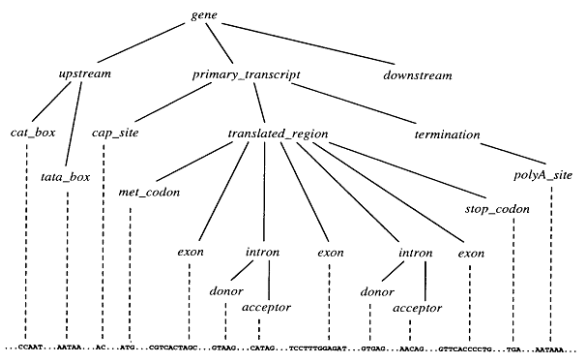
The linguist sees a famous young biologist
The linguist sees a famous young young young young biologist

Derivation tree



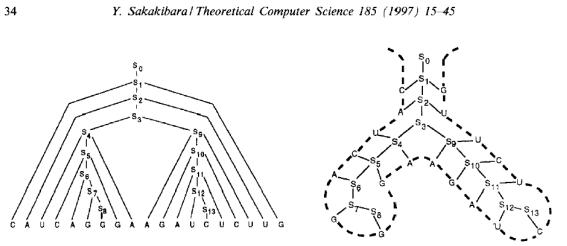
Source : D. Searls

Derivation tree



Source : D. Searls

Derivation tree



34 Y. Sakakibara / Theoretical Computer Science 185 (1997) 15–45

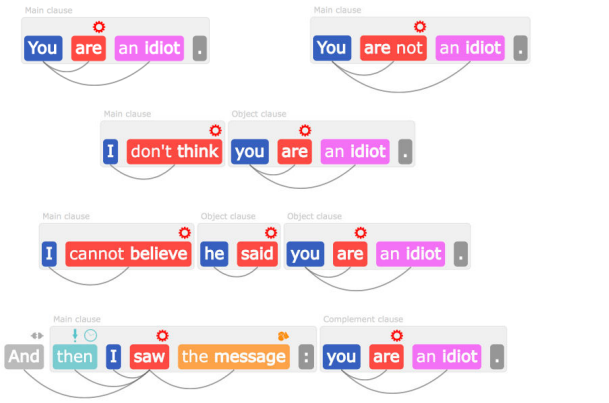
Fig. 3. A derivation tree (left) generated by a simple CFG for RNA molecules and the physical secondary structure (right) of the RNA sequence which is a reflection of the derivation tree.

Grammars for online violence detection

Courtesy of Gniewosz Leliwa
Director of AI Research & Co-founder of Samurai Labs
http://www.samurailabs.com

GRAMMAR AS A KEY FOR ACHIEVING HIGH PRECISION

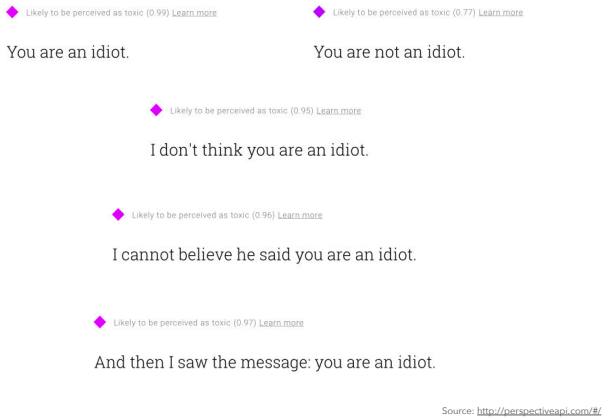
samurai



All of the graphics are examples of a Samurai Labs' Syntactic Parser output

GRAMMAR AS A KEY FOR ACHIEVING HIGH PRECISION

samurai



Source: http://perspectiveapi.com/#/

Expressivity level?

n-gram stochastic language models

See *stochastic language models* [E.G. Schukat-Talamazzini 1995]

- Explicit short distance sequential correlation (symbols are not *i.i.d.*): probability of n^{th} symbol given $n-1$ preceding symbols
- Markov approximation of order $k = n-1$

$$P(w_1 \dots w_m) = \prod_{i=1}^{m} P(w_i \mid w_{i-k} \dots w_{i-1})$$

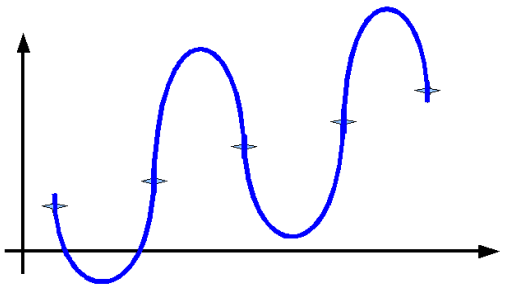
- Maximum likelihood estimation + smoothing (interpolating, discounting, backing-off, ...)
- Choice of n for best bias-versus-variance trade-off (enough data should be available to estimate counts)
- Trigram models have been extremely successfull in natural language processing (speech, language idenfication, translation, ...)

Can we learn more expressive models?

Generalisation is difficult

A finite set of examples
→ an infinite number of compatible concepts
Which one is the solution?

Remember:



Generalisation is difficult

Especially with expressive models!

A finite set of examples
→ an infinite number of compatible concepts
Which one is the solution?

- Occam's razor principle
  choose simplest compatible solution
  - MDL : $h_{MDL} = \text{argmin}_{h \in H} L(h) + L(X|h)$
    minimize length of coding h and its exceptions
- Use and introduce *a priori* knowledge
  (Learning biases)
  - Representation bias
  - Preference bias

Next lecture: Grammatical Inference