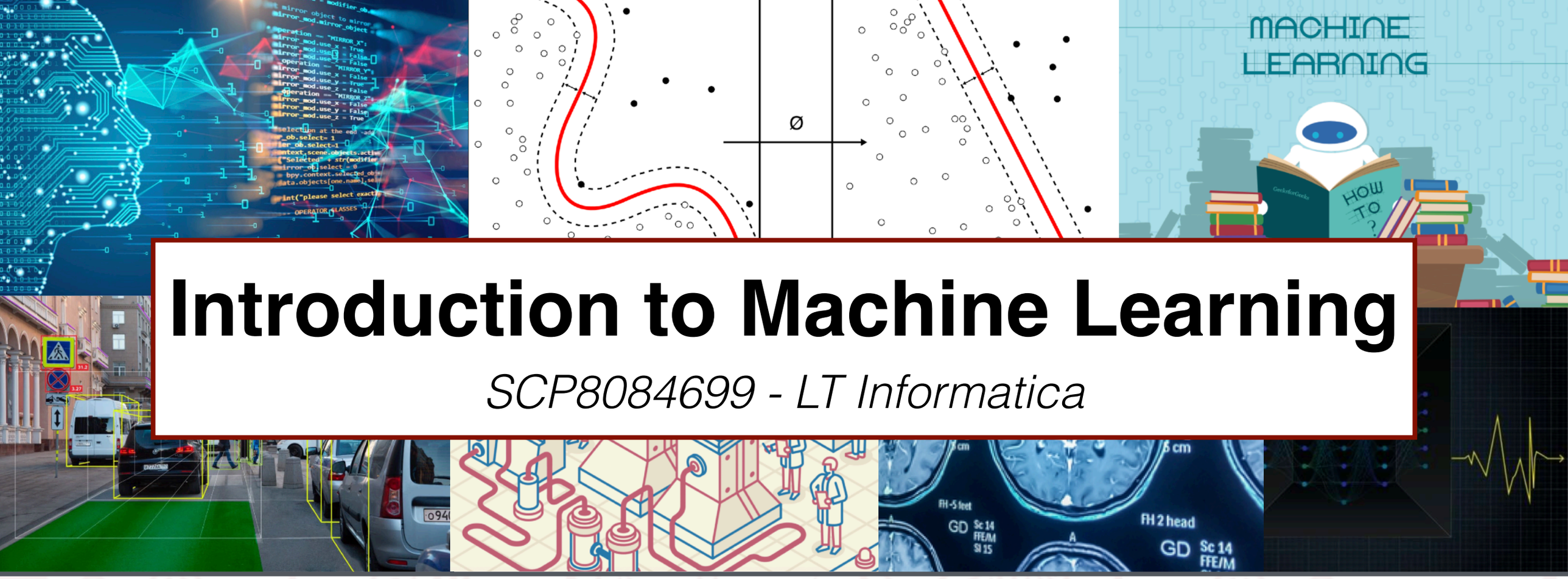


Introduction to Machine Learning

SCP8084699 - LT Informatica

Probabilistic models & NLP

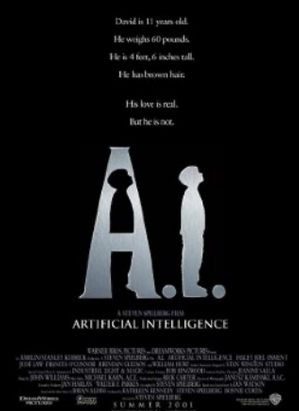
Prof. Lamberto Ballan

University of Padova, A.Y. 2021/22 May 17, 2022

DIPARTIMENTO MATEMATICA
Department of Mathematics “Tullio Levi-Civita”

Recap: AI in science fiction

What's in common?



Recap: Turing test

- The **imitation game** (based on language):
  - The interrogator (C) is unable to see players (A, B) and can communicate with them only through written notes
  - The interrogator tries to determine which player is a computer and which is a human

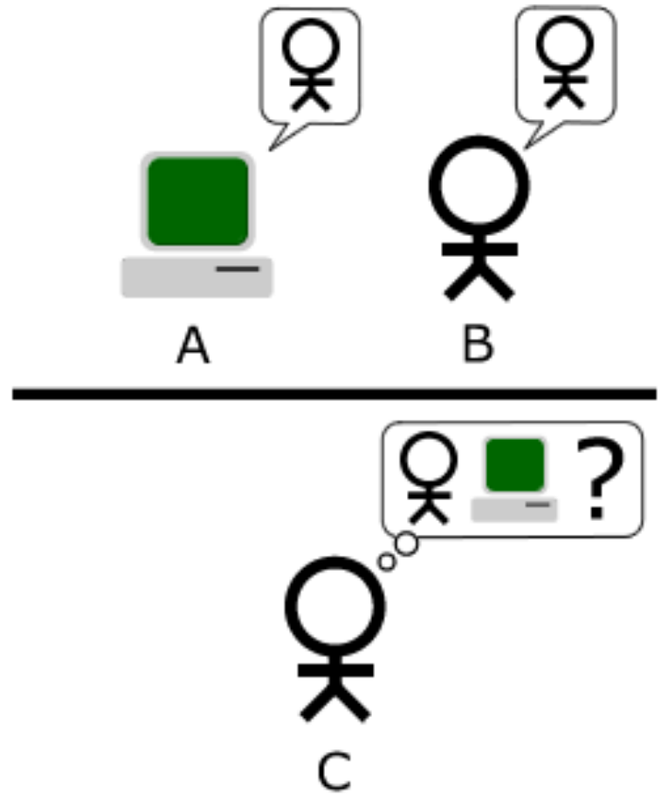
A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.



http://cogprints.org/499/1/turing.html

Recap: ML is fundamental in NLP

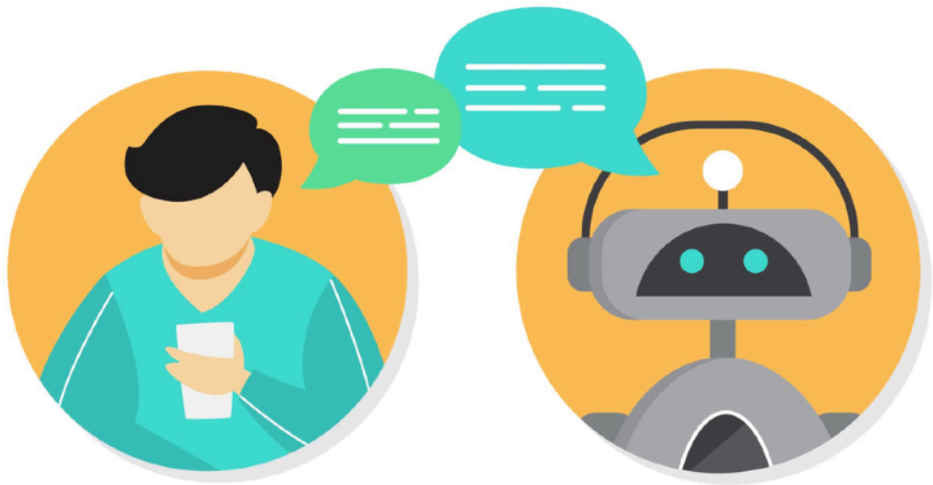
Back in 2000 , People Magazine PUBLISHER highlighted Prince Williams' PERSON style who at the time was a little more fashion-conscious , even making fashion statements at times .

Now-a-days the prince mainly wears navy COLOR suits ITEM (sometimes double-breasted DESIGN) , light blue COLOR button-ups ITEM with classic LOOK pointed DESIGN collars PART , and burgundy COLOR ties ITEM .

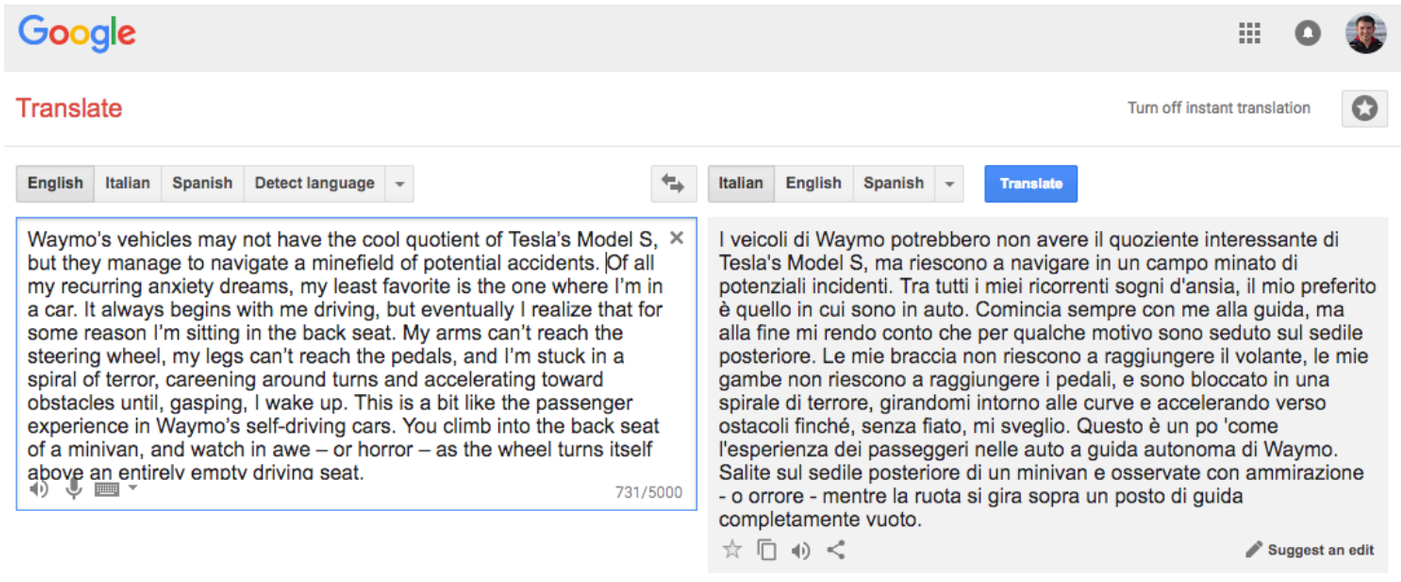
But who knows what the future holds ...

Duchess Kate PERSON did wear an Alexander McQueen BRAND dress ITEM to the wedding OCCASION in the fall of 2017 SEASON .

Named Entity Recognition



Question Answering



Machine Translation

The nature of human language

- What's special about human language?

- A human language is a system constructed to **convey the** speaker/writer's **meaning**
  - Not just an environmental signal, it's a deliberate communication
  - Using an encoding which little kids can quickly learn (amazingly!) and which changes

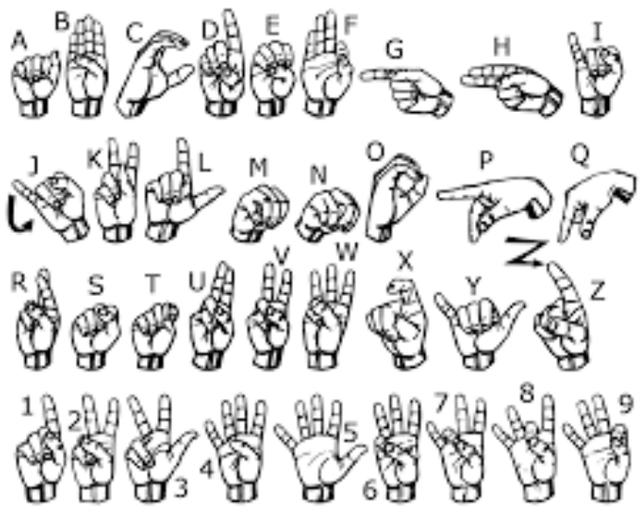
- A human language is mostly a (discrete) **symbolic/ categorical** "signaling" **system**

Adapted from slide by R. Socher

Introduction to NLP

- The **symbols** of a language can be encoded as a signal for communication in several ways:
  - Sound
  - Gesture
  - Writing / images

- A **symbol is invariant** across different encodings



Introduction to NLP

- What is Natural Language Processing (NLP)?
  - Goal: a computer agent which is able to understand natural language - **knowledge acquisition** - and **communicate** with humans
  - This would allow to perform useful tasks such as:
    - Language translation
    - Question answering
    - Text summarisation, …
- Fully understanding and representing the meaning of language (or even defining it) is a very difficult goal
  - Human language is often ambiguous!

NLP levels

Text

Speech

OCR / Tokenization

Phonetic / Phonological Analysis

Morphological Analysis

Syntactic Analysis

Semantic Interpretation

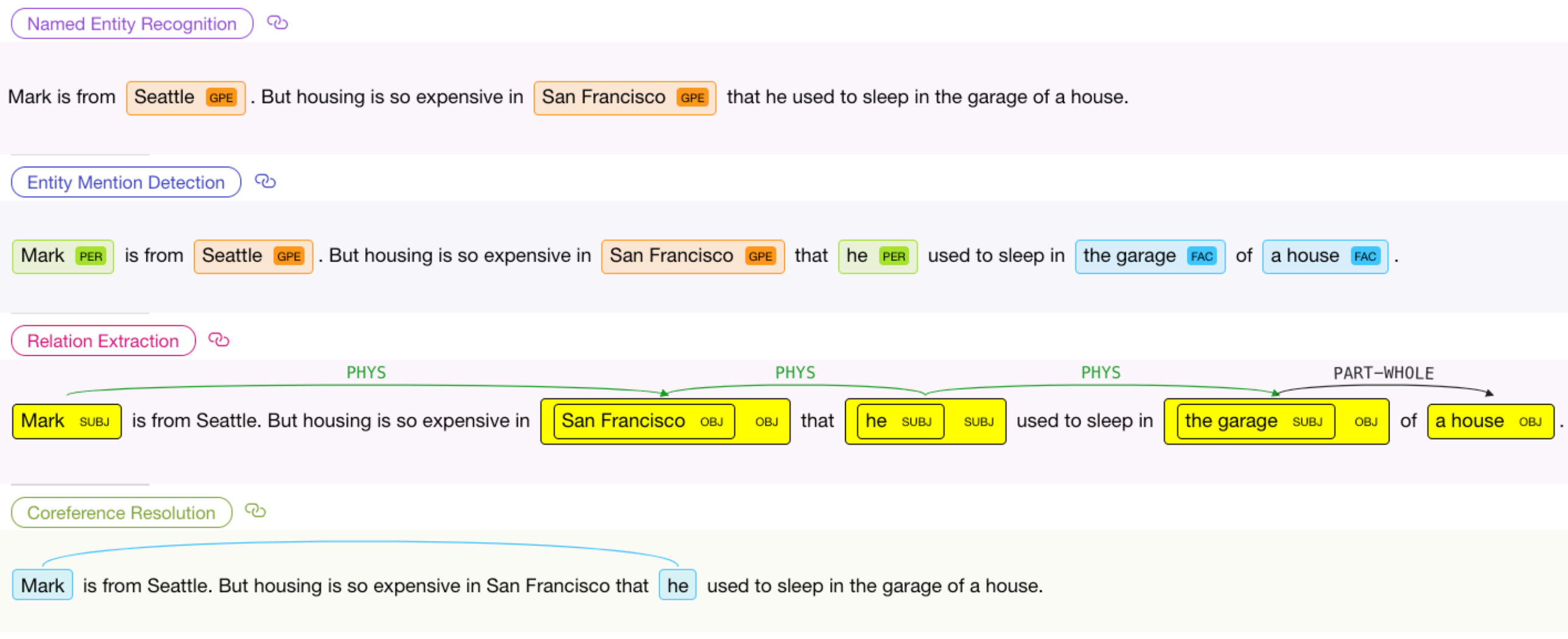
Discourse Processing

NLP applications

- A sample of NLP applications, from simplex to complex:
  - Spell checking, keyword search, finding synonyms
  - Extracting information from documents (or websites)
    - e.g. product price, date, location, people/company names
  - Text classification, reading level of school texts, sentiment analysis of longer documents
  - Machine translation
  - Conversational (spoken dialog) agents, e.g. chatbots
  - Question answering

NLP/NLU fundamental tasks

- Let's take a quick look at 4 fundamental tasks in natural language processing/understanding:



Reference: https://goo.gl/1NpkAz

Knowledge acquisition

- We will (mostly) look at NLP in terms of **information-seeking** tasks:
  - Text classification
  - Information Retrieval
  - Information Extraction

- A common factor in addressing these tasks is the use of **language models**, i.e. models that predict the probability distribution of language expressions

Language models

- *Definition:* A (statistical) **language model** is a probability distribution over sequences of words

- A **formal language** (e.g. Python) consists of strings specified by a set of rules called **grammar**
  - The grammar defines which are the legal programs
  - Example:

    `print(2 + 2)` *This is a legal program in Python*

    `)2 +2) print` *This is not*

  - A formal language has also rules that define the **semantic** of a program, e.g. `4` is the meaning of `(2 + 2)`

Language models

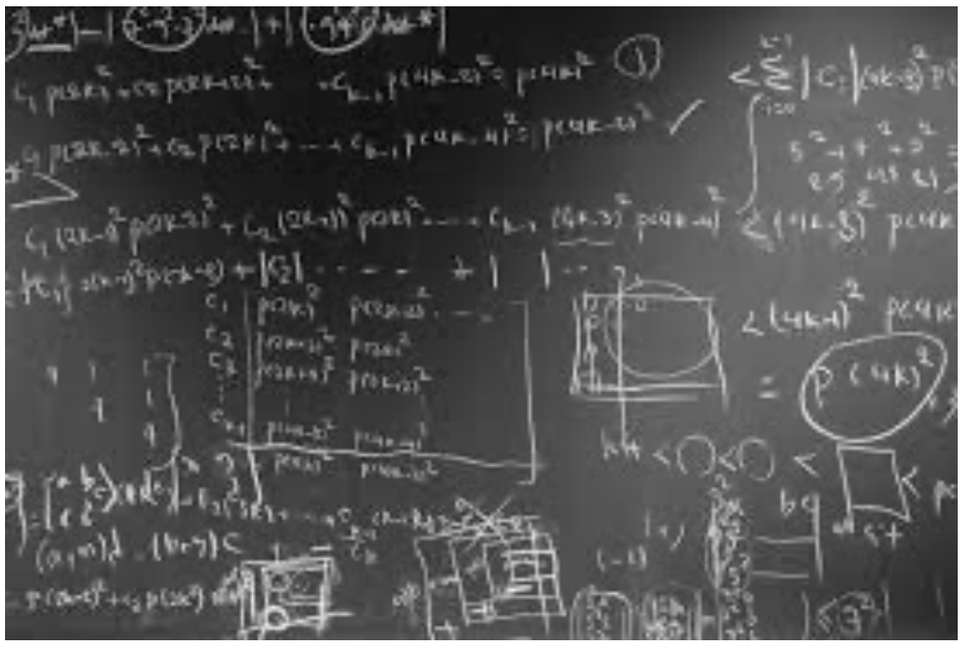
- **Natural languages** can not be characterized as a definitive set of sentences
  - "Not to be invited is sad." *everyone agrees is sentence of English*
  - "To be not invited is sad." *people disagree on its grammaticality*

- Natural languages are **very large** and **ambiguous**
  - "Enraged cow injures farmer with axe"

- Rather than asking if a string of *words* is or is not a member of the set defining the language, we ask for $P(S=words)$

Reminder: probabilities, chain rule

- Recall the definition of conditional probabilities
- More variables
- The chain rule in general
- Bayesian inference



More on blackboard…

The *n*-gram model

- A written text is basically composed of characters (letters, punctuation ad spaces)

- ***n*-gram character model**: it is the language model defined as the probability distribution over sequences of characters:

$P(c_{1:N})$:= *the probability of a sequence of N characters,* c_1 *through* c_N

  - A sequence of written symbols of length *n* is an *n*-gram
    - Special cases: unigram (*1*-gram), bigram (*2*-gram), trigram, …
    - E.g.: in a Web collection $P(\text{“the”})=0.027$ and $P(\text{“zgq”})=0.000000002$

More in general we can have n-gram models over sequences of words (or other units), not just over characters

The *n*-gram model

- *Definition:* an ***n-gram*** is a Markov chain of order n-1
- In a Markov chain the probability of a character c_i depends only on the probability of the immediately preceding chars

  - E.g. in a trigram model (Markov chain of order 2) we have:

$$P(c_i \mid c_{1:i-1}) = P(c_i \mid c_{i-2:i-1})$$

  - We can define $P(c_{1:N})$ under the trigram model by factoring with the chain rule and then using the Markov assumption:

$$P(c_{1:N}) = \prod_{i=1}^{N} P(c_i \mid c_{1:i-1}) = \prod_{i=1}^{N} P(c_i \mid c_{i-2:i-1})$$

The *n*-gram model

- What can we do with *n*-gram character models?
- Language Identification
  - e.g. “Hello world” *vs* “Come stai?”
  - A simple approach for language is identification is based on building a trigram model of each candidate language:

$$P(c_i \mid c_{i-2:i-1}, \ell)$$

  - For each language the model is built by counting trigrams in a corpus of that specific language (~100K chars needed)
  - This gives us the probability: $\mathbf{P}(\mathit{Text} \mid \mathit{Language})$

The *n*-gram model

- What can we do with *n*-gram character models?
- Language Identification
  - …
  - We want to select the most probable language given the text, so:

$$\ell^* = \operatorname*{argmax}_\ell P(\ell \mid c_{1:N}) = \operatorname*{argmax}_\ell P(\ell) P(c_{1:N} \mid \ell)$$

$$= \operatorname*{argmax}_\ell P(\ell) \prod_{i=1}^{N} P(c_i \mid c_{i-2:i-1}, \ell)$$

The *n*-gram model

- That's the trigram model from a corpus, but what about the prior probability $P(\ell)$?
  - We can estimates these values (e.g. check a random web page)
  - The choice of these priors is not critical because the trigram model selects a language that is order of magnitude higher than the others
- A few other popular applications:
  - Spelling correction
  - Text (genre) classification
  - Named-entity recognition

Model evaluation

- Intuition: pick the model which assign a higher probability to the (next) word that actually occurs!
  - The Shannon game: how well can we predict the next word?

"I always order pizza with cheese and ____"

"The 33rd President of the US was ____"

"I saw a ____"

mushrooms (0.1)
pepperoni (0.1)
anchovies (0.05)
pineapple (0.03)
…
fried rice (0.0001)
…
and (1e-100)

  - Unigrams are terrible at this game
  - So the best language model gives (on average) the best $P(\text{Sentence})$

The *n*-gram (word) model

- The same approach applies equally to word models
  - A key difference is that the vocabulary is significantly larger
    - Usually there are ~100 characters in most languages
    - Word models have tens of thousands (or millions) of symbols
    - We also need to deal with *out of vocabulary* words (a common trick is to define a “special" <UNK> word)
    - Let’s see an example to “visualise” what word models do:

Unigram: logical are as are confusion a may right tries agent goal the was . . .
Bigram: systems are very similar computational approach would be represented . . .
Trigram: planning and scheduling are integrated the success of naive bayes model is . . .

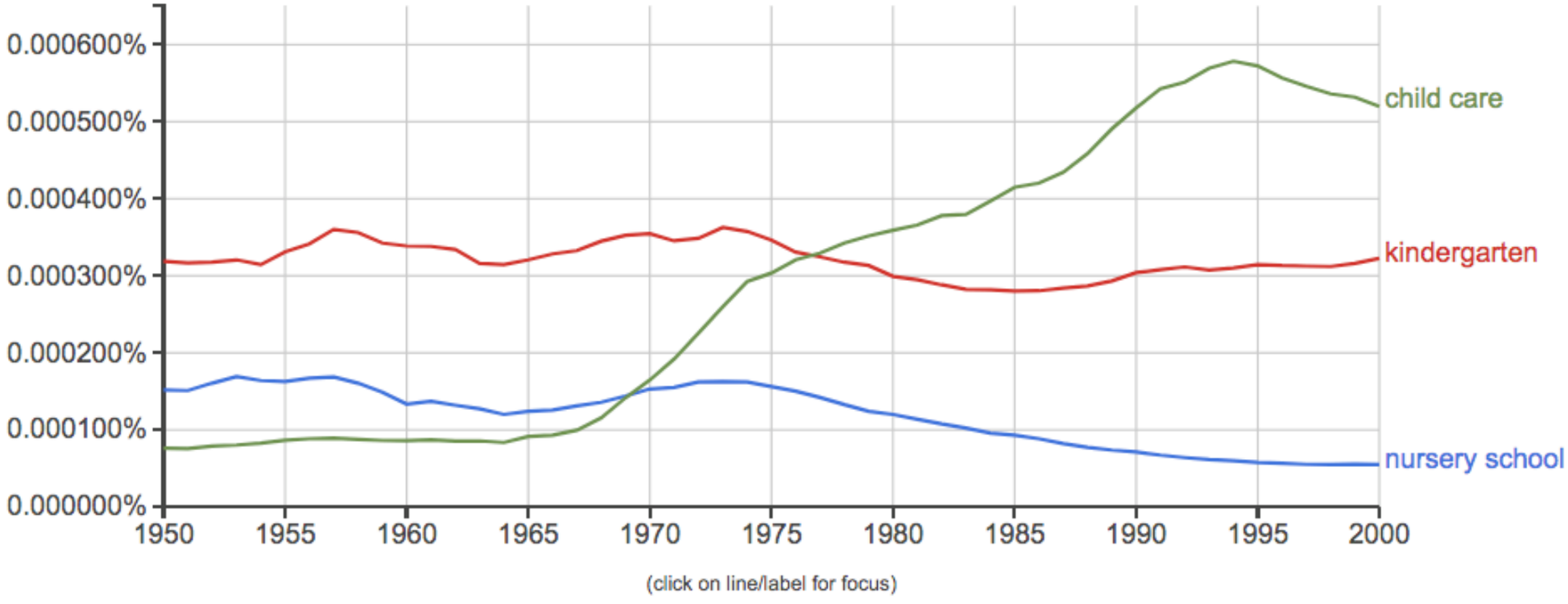
A practical example

- Google *n*-gram viewer: https://books.google.com/ngrams

Google Books Ngram Viewer

What does the Ngram Viewer do?

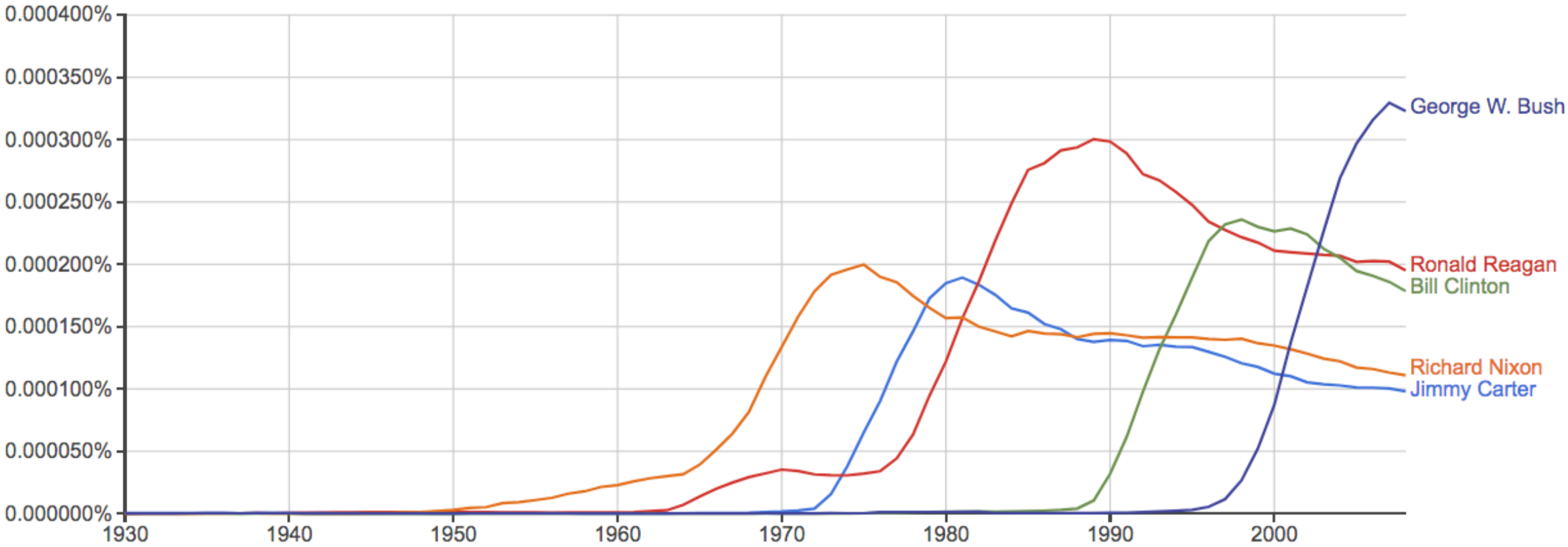
When you enter phrases into the Google Books Ngram Viewer, it displays a graph showing how those phrases have occurred in a corpus of books (e.g., "B
years. Let's look at a sample graph:



This shows trends in three ngrams from 1950 to 2000: "nursery school" (a *2-gram* or *bigram*), "kindergarten" (a *1-gram* or *unigram*), and "child care" (anoth
contained in our sample of books written in English and published in the United States, what percentage of them are "nursery school" or "child care"? Of all

A practical example

- USA presidents example: https://goo.gl/u2TK3o



n-grams and chain rule

- Goal: compute the probability of a sentence (words sequence): $P(\mathrm{W}) = P(w_1, w_2, w_3, \ldots, w_n)$

- A **language model** is a model that computes either of these: $P(\mathrm{W})$ or $P(w_n \mid w_1, w_2, w_3, \ldots, w_{n-1})$

- The chain rule applied to compute joint probability of words: $P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$

  - An example:

$$P(\text{“its water is so transparent that”}) =$$

$$= P(\text{“its”})\, P(\text{“water} \mid \text{its”})\, P(\text{“is} \mid \text{its water”})\, P(\text{“so} \mid \text{its water is”})$$
$$P(\text{“transparent} \mid \text{its water is so”})\, P(\text{“that} \mid \text{its water is so transparent”})$$

n-grams and chain rule

- How to estimate these probabilities? Could we just count?

$$P(\text{“the | its water is so transparent that”}) = \frac{Count(\text{“its water is so transparent that the”})}{Count(\text{“its water is so transparent that”})}$$

  - No! Too many sentences… we’ll never see enough data to estimate these probabilities

Contact

- **Office:** Torre Archimede 6CD, room 622
- **Office hours** (ricevimento)**:** Monday 11:00-13:00

✉ lamberto.ballan@unipd.it

🏠 http://www.lambertoballan.net

🏠 http://vimp.math.unipd.it

@ twitter.com/lambertoballan