U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

LIACC

MIEIC
L.EIC0029 – Artificial Intelligence

Introduction to Natural Language Processing

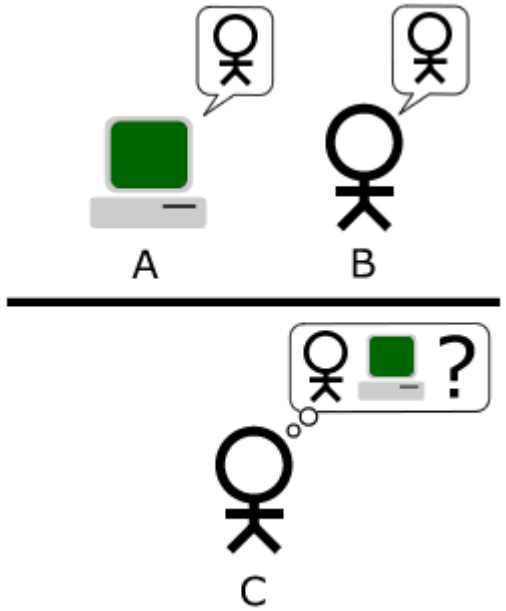
Henrique Lopes Cardoso

hlc@fe.up.pt

The Turing Test

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

[Alan Turing, 1950]



Capabilities:

- natural language processing
- knowledge representation
- automated reasoning
- machine learning

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

LIACC

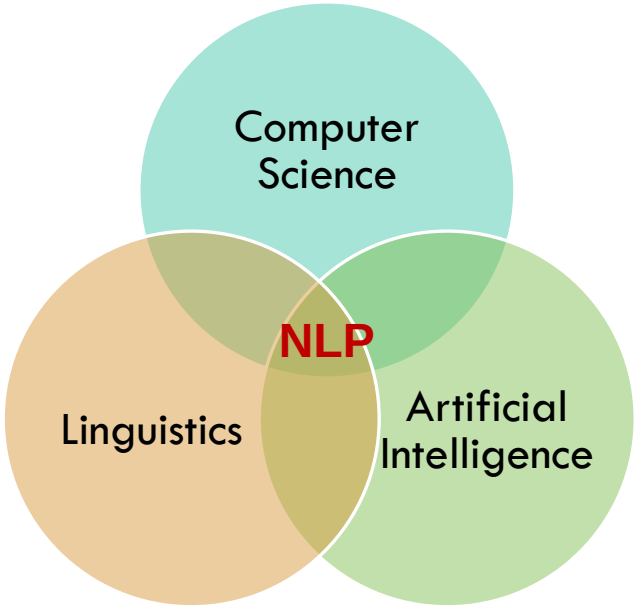
Natural Language Processing (NLP)

definitions, tasks and applications

Natural Language Processing

Natural language processing (**NLP**) is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language corpora.

[Wikipedia]



NLP Tasks

- Most NLP tasks aim at making it easier for machines to process natural language

- Tokenization
  - Split a sentence into tokens (words)

```
That U.S.A. poster-print costs $12.40...
['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']
```

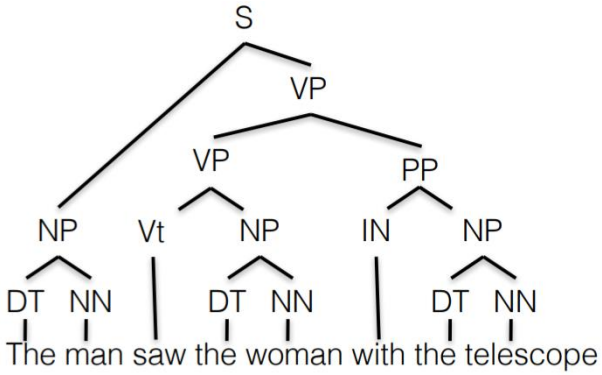
- Sentence breaking
  - Split a text into sentences

```
Hello. Are you Mr. Smith? I've finished my M.Sc. on Informatics!
['Hello.',
 'Are you Mr. Smith?',
 'I've finished my M.Sc. on Informatics!']
```

- Part-of-Speech (POS) tagging
  - Determine the role category for each word in a sentence

NLP Tasks

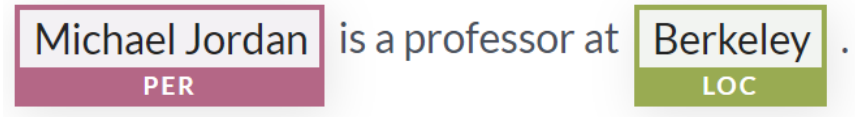
- Syntax parsing
  - Determine the parse tree (grammatical analysis) of a sentence



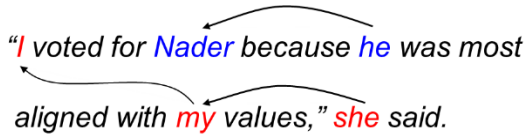
- Word sense disambiguation
  - Select the meaning of words in a context

A mouse is a mammal.
My mouse is broken.

- Named-entity recognition (NER)
  - Determine which items in a text map to entities (people, institutions, places, dates, …)



- Co-reference resolution
  - Determine which words (“mentions”) refer to the same objects (“entities”)



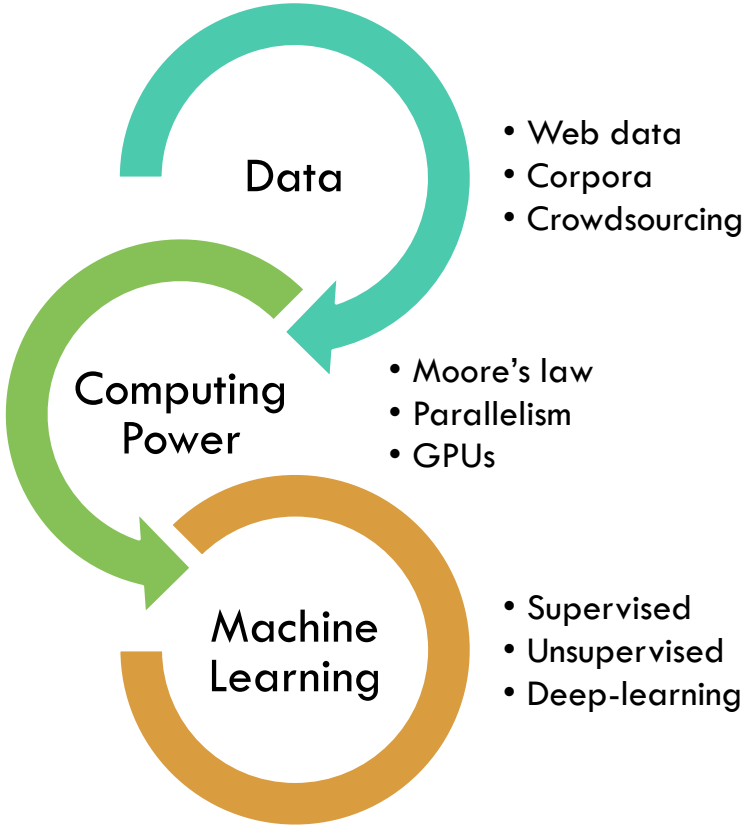
- …

Language Resources

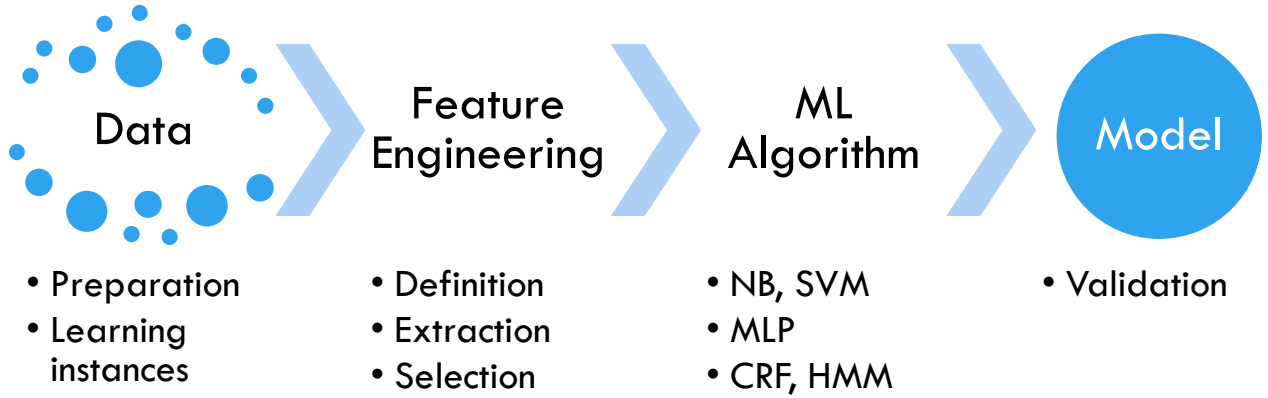
- Lexical databases: WordNet, CONTO.PT, WordNet.pt, …
  - Synsets, word-sense pairs
  - Semantic relations: hypernym/hyponym, meronym/holonym, troponym, entailment, …
- TreeBanks: PDTB, CSTNews, …
  - Text corpora annotated with discourse or semantic sentence structures
- Knowledge graphs: Google, DBpedia, …
  - Entity-predicate relations
- Lexicons: SentiWordNet, SocialSent, SentiLex, VADER, …
  - Words connoted with specific classes (+/-, objectivity, …)
- Word embeddings: word2vec, GloVe, fastText, …
  - Distributed representations of words
- Language Models: ELMo, BERT, GPT, …
- Annotated datasets for several NLP tasks
  - Often released under “shared-tasks”, such as those at SemEval or CLEF
- …

Statistical NLP

- Data-driven statistical techniques have largely overtaken knowledge (grammar) based methods

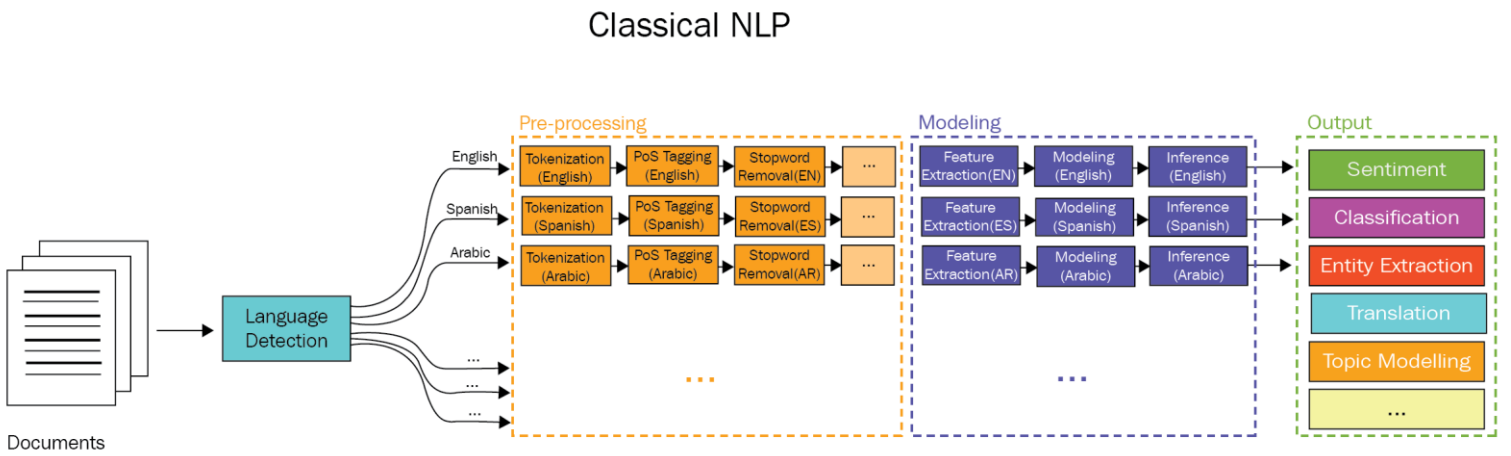


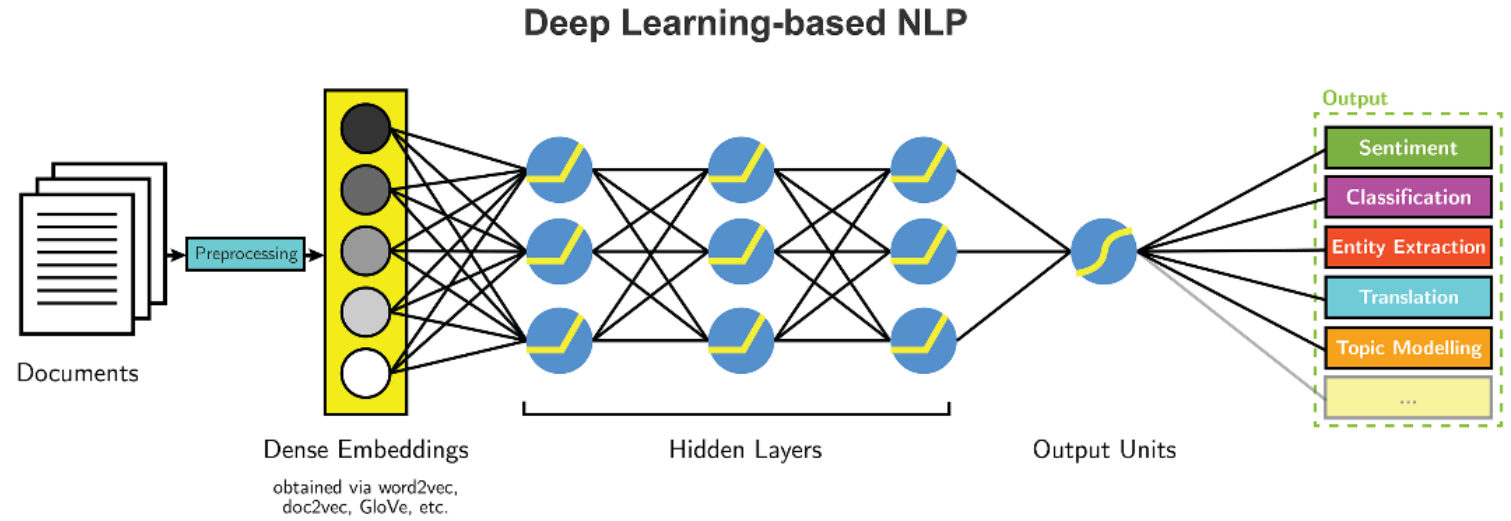
Machine Learning in NLP



- Common linguistic features used in NLP
  - *Lexical*: BoW, TF-IDF, n-grams, word stems, …
  - *Syntactic*: part-of-speech (POS) tagging, parsing, …
  - *Grammatical*: verb tenses, number, gender, …
  - *Semantic*: word similarities, relations, embeddings, …
  - *Structural*: paragraphs, sentence length, document sections, distance metrics, …

Classical vs Deep Learning NLP

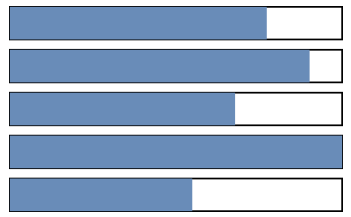




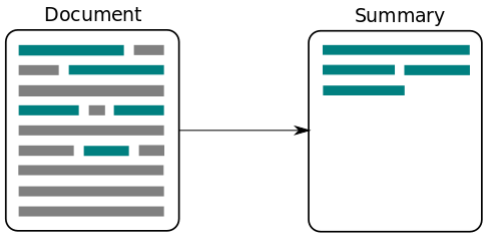
Some NLP Applications

- Sentiment Analysis and Opinion Mining
  - Determine polarity about specific topics
  - Identify trends of public opinion in social media
  - Analyze product reviews

Attributes:
- zoom
- affordability
- size and weight
- flash
- ease of use



- Text Summarization
  - Build a summary out of a long text



Some NLP Applications

- Machine Translation
  - Based on multilingual textual corpora
  - Text translation and multilingual real-time conversations

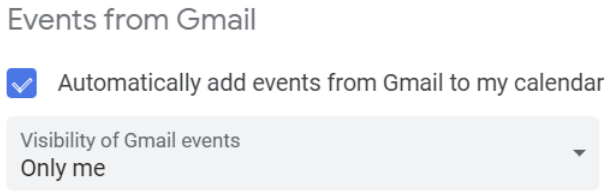
DeepL

- Speech-to-Text/Text-to-Speech
  - Convert spoken language to written text and vice versa
  - Voice control, domotics, readers, …

amazon alexa

Some NLP Applications

- Information Extraction
  - Extract relevant entities from text
  - Event identification, "add to calendar" features



Subject: extra NLP class
Date: September 25, 2020
To: Henrique Lopes Cardoso

Dear Henrique, we're having an extra NLP class tomorrow, from 10:00-11:30, via Zoom.
-HLC

Event: extra NLP class
Date: Sept-26-2020
Start: 10:00am
End: 11:30am
Where: Zoom

- Question Answering
  - Automatically answer questions posed in natural language
  - IBM Watson won *Jeopardy!* on 2011

Some NLP Applications

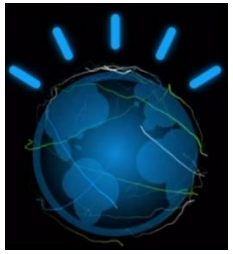
- Fact Checking and Fake News Detection
  - Given a claim, collect evidence to check if it is true
  - Given a news article, check whether it is accurate

FEVER

- Argument Mining and Debate Portals
  - Extract arguments that expose a certain position
  - Aggregate pros and cons for a debatable topic
  - Debate on a given topic

PRO DEBATEPEDIA CON

BRITANNICA PROCON.ORG | RELIABLE. NONPARTISAN. EMPOWERING.



Some NLP Applications

- Natural Language Generation
  - Generate a narrative from data
  - Generate source code from natural language descriptions
  - Language models

- Conversational AI (Chatbots)
  - Conversational interfaces
  - Human-like voice assistants

Basic Text Processing

regular expressions, tokenization, normalization, lemmatization, stemming

Regular Expressions

- A regular expression is a sequence of characters that define a search pattern
  - Makes use of meta-characters, such as {}[]()^$.|*+?\
    - [A-Z] uppercase letter
    - [a-z] lowercase letter
    - [0-9] digit
    - ^ negation
    - | disjunction
    - ? optional
    - * zero or more
    - + one or more
    - . Any
    - …

Example: find all instances of the word "**the**" in a text

- the misses capitalized letters
- [tT]he returns "other" or "theology"
- [^a-zA-Z][tT]he[^a-zA-Z]

The role of Regular Expressions

- Sophisticated sequences of regular expressions are often a first model for many text processing tasks
  - E.g. detecting named entities based on Capitalization

- For harder tasks or increased performance, we use machine learning classifiers
  - But regular expressions are still used for pre-processing, or as features in the classifiers
  - Can be very useful in capturing generalizations

Word Tokenization

- Initial approach: look for spaces, punctuation and other special characters
- What about:
  - Ph.D., AT&T, can't, we're, state-of-the-art, guarda-chuva
  - $45.55, 123,456.78, 123.456,78
  - 07/04/2020, April 4, 2020
  - http://www.fe.up.pt, hlc@fe.up.pt, #iart
  - New York, Vila Nova de Gaia
- Certain languages do not have space splitting!
  - Chinese, Japanese, some words in German, …

```
import nltk
from nltk import word_tokenize

text = 'That U.S.A. poster-print costs $12.40...'
tokens = word_tokenize(text)
print(len(tokens))
print(tokens)
```

```
7
['That', 'U.S.A.', 'poster-print', 'costs', '$', '12.40', '...']
```

```
word_tokenize("I don't think we're flying today.")
```

```
['I', 'do', "n't", 'think', 'we', "'re", 'flying', 'today', '.']
```

Word Normalization

- Putting words/tokens in a standard format
  - Reduces the vocabulary size
  - Helps Machine Learning models to generalize

- Case folding
  - Putting every word in lower case
  - Not always helpful, and thus not always performed
    - Sentiment analysis: uppercase might denote anger, …
    - Named-entity recognition: US/us, Mike Pence/mike pence, …

Lemmatization

- Determining the root of the word: many words have the same root!
  - "am", "are", "is" → "be"
  - "He is reading detective stories" → "He be read detective story"

- Morphological parsing: words are built from morphemes
  - Stem: the central morpheme of a word, supplying the main meaning
  - Affix: adding additional meaning
  - *cats* = *cat* (stem) + *s* (affix)
  - *iremos* = *ir* (stem) + 1st plural + future tense (morphological features)

Stemming

- Lemmatization algorithms can be complex
- Stemming: a simpler and cruder method that simply cuts off word final affixes
  - Subject to over- and under-generalization

The European Commission has funded a numerical study to analyze the purchase of a pipe organ with no noise for Europe's organization. Numerous donations have followed the analysis after a noisy debate.

⇩

the european commiss ha fund a numer studi to analyz the purchas of a pipe organ with no nois for europ 's organ . numer donat have follow the analysi after a noisi debat .

- The Porter stemmer [Porter, 1980]
  - Set of rules run in series
    - ATIONAL → ATE (e.g., relational → relate)
    - ING → ε if stem contains vowel (e.g., motoring → motor)
    - SSES → SS (e.g., grasses → grass)

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

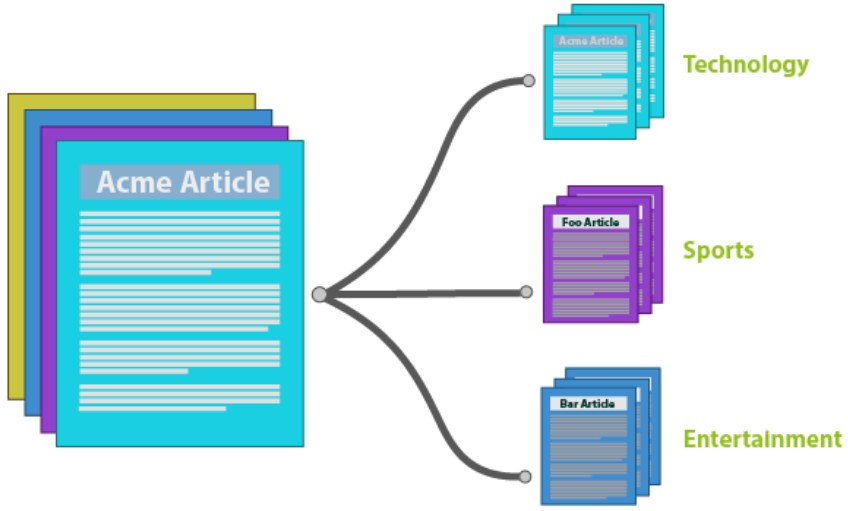
LIACC

Text Classification

bag-of-words, Naïve Bayes, features, generative and discriminative classifiers

Text Classification Tasks

- Given a text, classify it according to a number of classes
  - Spam detection in emails: spam/not spam
  - Sentiment analysis in product reviews: positive/negative, -/0/+, --/-/0/+/++
  - Assign subject categories, topics, or genres
  - Authorship identification from a closed list
  - Age/gender identification
  - Language detection
  - …



- More formally:
  - Input: a document d and a fixed set of classes $C = \{c_1, c_2, \ldots, c_m\}$
  - Output: predicted class $c \in C$ for document d

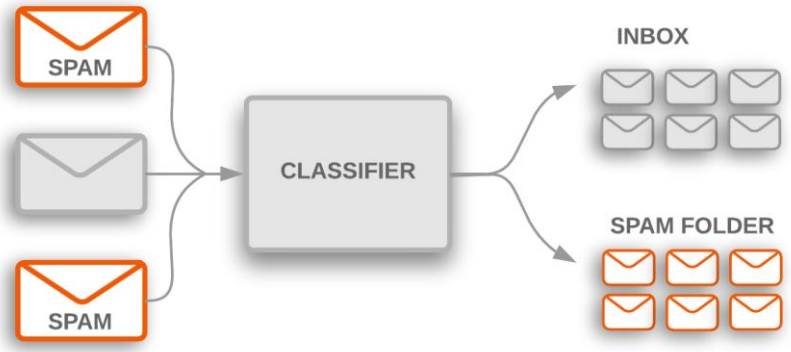
Hand-coded Rules

- Rules based on combinations of words or other features
  - Spam detection: black-list of addresses and keyword detection
  - Sentiment analysis: ratio of word polarities appearing in a sentiment lexicon

- Accuracy can be high…
  - If rules are carefully refined by expert
- …but building and maintaining these rules is expensive

Supervised Machine Learning

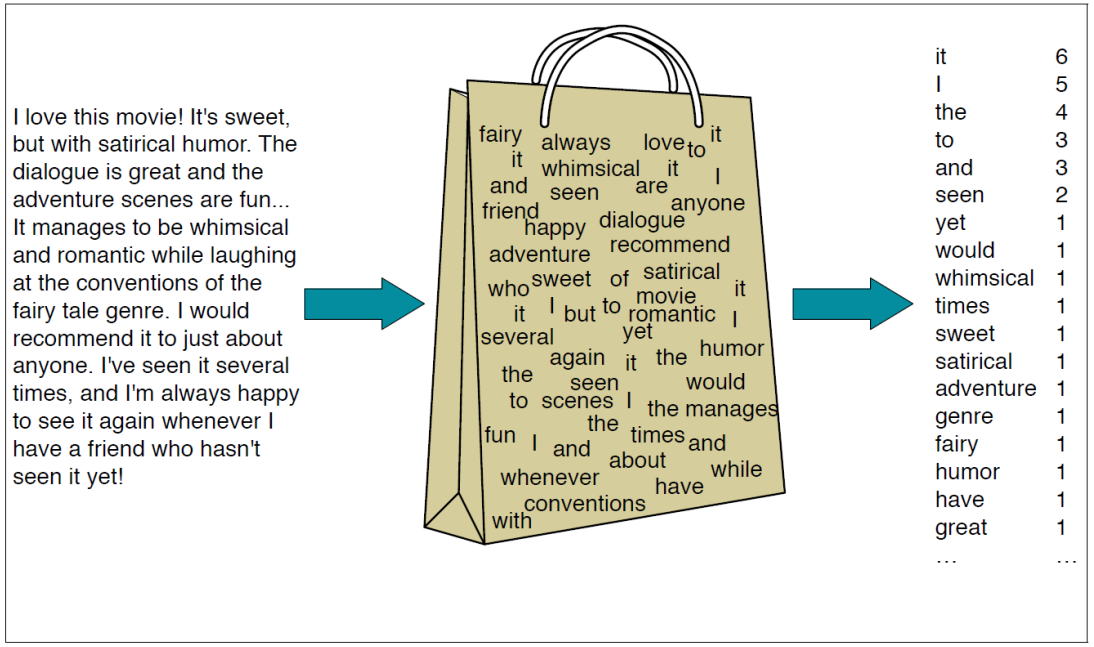
- Making use of annotated datasets through Machine Learning algorithms
- Building a model
  - Input:
    - a fixed set of classes $C = \{c_1, c_2, \ldots, c_m\}$
    - a training set of m hand-labeled documents $\{(d_1,c_1), (d_2,c_2), \ldots, (d_n,c_n)\}$, where $d_i \in D$ and $c_i \in C$
  - Output: a classifier $\gamma : D \rightarrow C$
    - a mapping from documents to classes (or class probabilities)
- Classifying a document
  - Input
    - a document d
    - a classifier $\gamma : D \rightarrow C$
  - Output: predicted class $c \in C$ for document d



Representing a Document with a Bag of Words

- Machine Learning methods require that the data is represented as a set of features
- We thus need a way of going from a document *d* to a vector of features *X*

- The bag-of-words model
  - an unordered set of words, keeping only their frequency in the document
  - assume position does not matter



Naïve Bayes

- Naïve Bayes (NB) makes a simplifying (naïve) assumption about how the features interact
- Bayes rule:

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

- Most likely class:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)} = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

Naïve Bayes Classifier

- Representing a document with features:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}} \qquad \hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, ..., f_n|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- Assuming conditional independence:

$$P(f_1, f_2, ..., f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot ... \cdot P(f_n|c)$$

Naïve Bayes Classifier

- Naïve Bayes classifier:

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{f \in F} P(f|c)$$

- Applying NB to the text:

$$\text{positions} \leftarrow \text{all word positions in test document}$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c)$$

Naïve Bayes: Computing Probabilities

- Class priors: $\hat{P}(c) = \frac{N_c}{N_{doc}}$

- Word probabilities per class:

$$\hat{P}(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

- Handling non-occurring words in a class
  - Add-one (Laplace) smoothing:

$$\hat{P}(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} (count(w, c) + 1)} = \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

Naïve Bayes Example

- A sentiment analysis (or polarity) task:

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

- Prior distributions:

$$P(-) = \frac{3}{5} \qquad P(+) = \frac{2}{5}$$

- Word probabilities per class:

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

  - “with” doesn’t occur in training set: ignore it

- Class probabilities:

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

  - Chosen class: **negative (-)**

Naïve Bayes is Not So Naïve

- Very fast, low storage requirements
- Robust to irrelevant features: they tend to cancel each other without affecting results
- Very good in domains with many equally important features
  - Decision Trees suffer from fragmentation in such cases – especially if little data
- Optimal if the assumed independence assumptions hold

- A good dependable baseline for text classification

Dealing with Negation

- *I really like this movie* (positive)
- *I didn't like this movie* (negative)

- Prepending `NOT_` to words affected by negation tokens (n't, not, no, never, …)
  - `I did n't like this movie , but I`
  - `I did n't NOT_like NOT_this NOT_movie , but I`

- Using bigrams instead of single words
  - Sequences of two words: instead of "not" and "recommend", "not recommend"

Making use of Lexicons

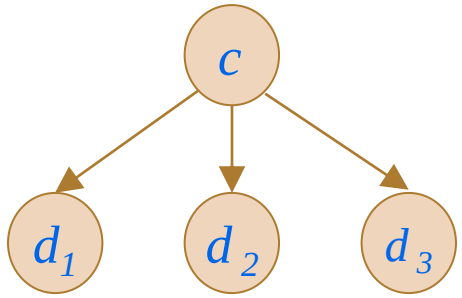
- Lexicons provide external knowledge that can be very useful for the task!
- Sentiment lexicons
  - Lists of words that are pre-annotated with positive or negative polarity
  - Example: VADER sentiment lexicon
    - 7520 "words", positive or negative biased, including intensity
    - + : magnificently (3.4), beautiful (2.9), admirable (2.6), confident (2.2), :-) (1.3), defensive (0.1)
    - – : amortize (-0.1), bias (-0.4), :-((-1.5), harsh (-1.9), bad (-2.5), catastrophe (-3.4), rapist (-3.9)
- Features based on the occurrence of (positive or negative) sentiment-biased words
  - Useful when training data is sparse or vocabulary usage in test and training sets do not match
  - Dense lexicon features may generalize better than sparse individual-word features

Building other Features

- Predefine likely sets of words or phrases
  - Spam detection: “viagra”, “password will expire”, “Your mailbox has exceeded the storage limit”, “millions of dollars”, “click here”, “urgent reply”, …
- Paralinguistic and extra-linguistic features
  - Words in capital letters
  - HTML with low ratio of text-to-image, sender email address, …
- N-grams (character or word level)
  - Sequences of two (bigrams), three (trigrams) or even more words or characters
  - Can help alleviate the conditional independence assumption of NB
  - But typically generates a very sparse feature space (many bigrams will rarely occur)

Generative vs Discriminative Classifiers

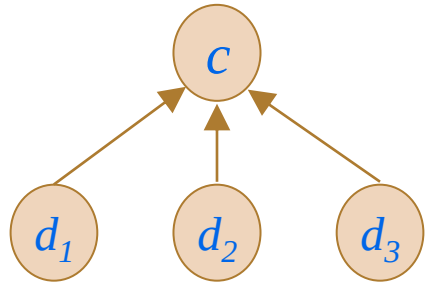
- A generative model makes use of a likelihood term: how to generate the features of a document if we knew it was of class c?



- Examples:
  - Naïve Bayes, Hidden Markov Models, …

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- A discriminative model tries to learn to distinguish the classes, and attempts to directly compute $P(c|d)$



- Examples:
  - Logistic Regression, Decision Trees, Support Vector Machines, Neural Networks, Conditional Random Fields, …

Generative vs Discriminative Classifiers

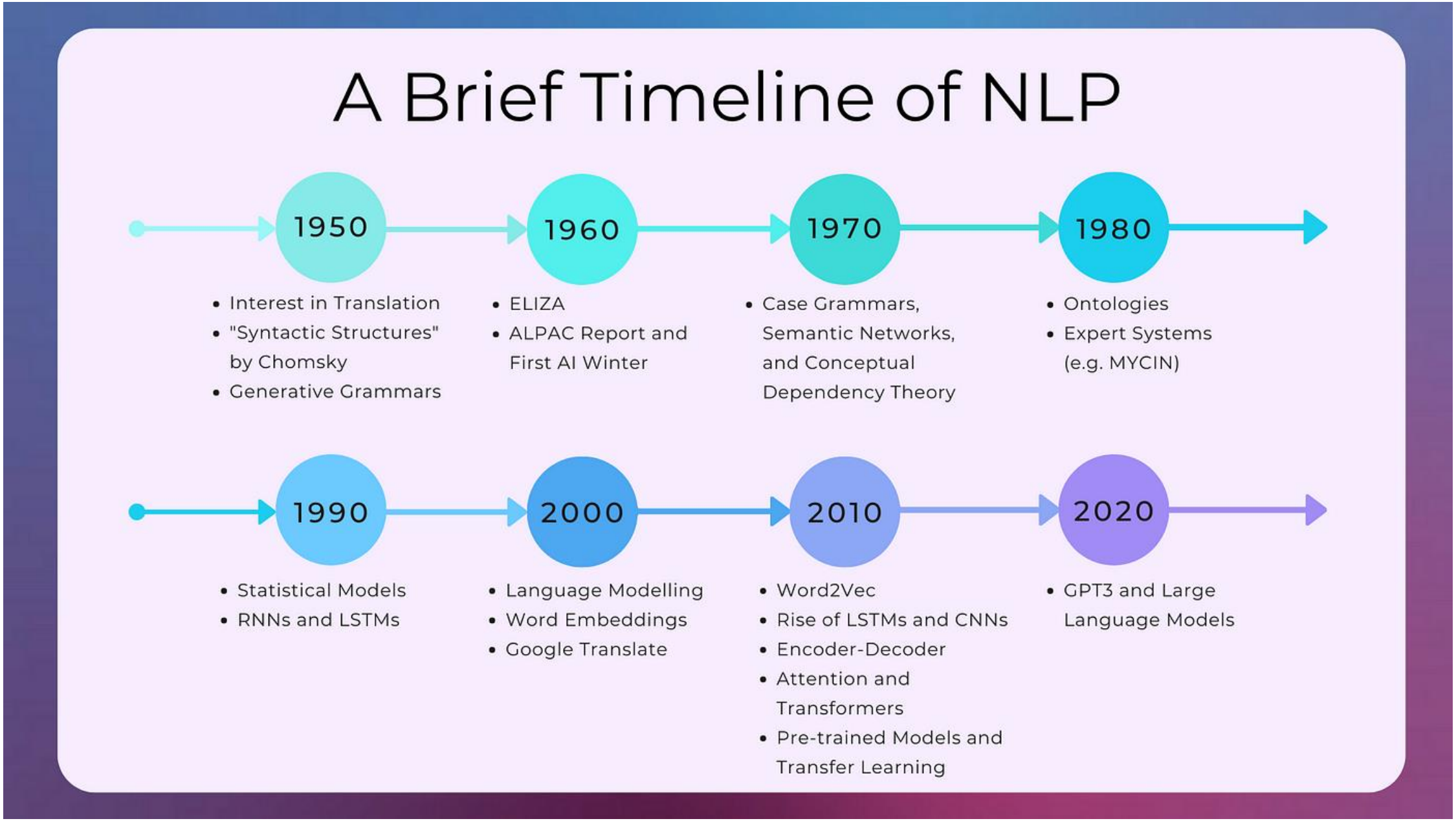
- Naïve Bayes has overly strong conditional independence assumptions
  - Edge case: two strongly correlated features, e.g., using the same feature twice
    - NB treats both copies of the feature as if they were separate
  - If multiple features tell mostly the same thing, such evidence is overestimated
- Discriminative classifiers (e.g., Logistic Regression) assign more accurate probabilities when there are many correlated features
- Naïve Bayes is easy to implement and very fast to train (there is no optimization step)
- Logistic Regression generally works better on larger documents or datasets

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

LIACC

Modern NLP

word embeddings, deep learning, language models



https://medium.com/nlplanet/a-brief-timeline-of-nlp-bc45b640f07d

Word Embeddings

- Representing a sentence based on a bag-of-words model obtains very sparse representations
  - Given a vocabulary of size |V|, a document is represented as a vector with many 0's and a few 1's

- We can represent the meaning of a word based on the contexts in which it occurs
  - Unsupervised approach: observe word usage on large (non-annotated) corpora
  - Sparse vectors (|V| = 20k? 50k?): words represented by a function of the counts of nearby words
  - Dense vectors: short vectors (50-1000 mostly non-zero real numbers), trained representations

- Representing sentences/documents: compute centroid of the word vectors

Word Embeddings

- Unsupervised learning of word embeddings: observe word usage in large collections of text!
  - Learn representations of the meaning of words from their distributions in texts

- Several pre-trained embeddings available, using different techniques
  - word2vec [Mikolov *et al.*, 2013]
  - GloVe [Pennington *et al.*, 2014]
  - fastText [Bojanowski *et al.*, 2017]
  - …

Corpus	Tokens	Types	Genre	Description
LX-Corpus [Rodrigues et al. 2016]	714,286,638	2,605,393	Mixed genres	A huge collection of texts from 19 sources. Most of them are written in European Portuguese.
Wikipedia	219,293,003	1,758,191	Encyclopedic	Wikipedia dump of 10/20/16
GoogleNews	160,396,456	664,320	Informative	News crawled from GoogleNews service
SubIMDB-PT	129,975,149	500,302	Spoken language	Subtitles crawled from IMDb website
G1	105,341,070	392,635	Informative	News crawled from G1 news portal between 2014 and 2015.
PLN-Br [Bruckschen et al. 2008]	31,196,395	259,762	Informative	Large corpus of the PLN-BR Project with texts sampled from 1994 to 2005. It was also used by [Hartmann 2016] to train word embeddings models
Literacy works of public domain	23,750,521	381,697	Prose	A collection of 138,268 literary works from the Domínio Público website
Lacio-web [Aluísio et al. 2003]	8,962,718	196,077	Mixed genres	Texts from various genres, e.g., literary and its subdivisions (prose, poetry and drama), informative, scientific, law, didactic technical
Portuguese e-books	1,299,008	66,706	Prose	Collection of classical fiction books written in Brazilian Portuguese crawled from Literatura Brasileira website
Mundo Estranho	1,047,108	55,000	Informative	Texts crawled from Mundo Estranho magazine
CHC	941,032	36,522	Informative	Texts crawled from Ciência Hoje das Crianças (CHC) website
FAPESP	499,008	31,746	Science Communication	Brazilian science divulgation texts from Pesquisa FAPESP magazine
Textbooks	96,209	11,597	Didactic	Texts for children between 3rd and 7th-grade years of elementary school
Folhinha	73,575	9,207	Informative	News written for children, crawled in 2015 from Folhinha issue of Folha de São Paulo newspaper
NILC subcorpus	32,868	4,064	Informative	Texts written for children of 3rd and 4th-years of elementary school
Para Seu Filho Ler	21,224	3,942	Informative	News written for children, from Zero Hora newspaper
SARESP	13,308	3,293	Didactic	Text questions of Mathematics, Human Sciences, Nature Sciences and essay writing to evaluate students
Total	1,395,926,282	3,827,725		

[Hartmann *et al.*, 2017: Portuguese Word Embeddings]

Word Embeddings

- Represent the meaning of a word through vectors of real values (embeddings)

'inteligência' = [-0.087885, -0.415836, 0.421546, -0.655831, 0.422223, 0.220453, -0.73256 , 0.358606, 0.090995, 0.574769, -0.105547, -0.079609, -0.140724, -0.140189, -0.813335, -0.209022, 0.855916, 0.452126, 0.560651, 0.004266, -0.031785, -0.061654, 0.595562, -0.83947 , 0.581227, 0.208089, 0.738459, -0.03487 , -0.226058, 0.123312, -0.091426, -0.012113, -0.135737, 0.459097, -0.495695, -0.166633, 0.282449, -0.265621, -0.296086, -0.866863, -0.105252, 0.215159, -0.546122, -0.16616 , 0.064767, 0.129924, 0.048979, 0.528899, -0.337607, 0.067035, -0.126727, -0.293169, 0.028446, 0.095485, -0.062157, -0.102588, 0.349045, -0.475593, -0.064273, -0.706759, 0.275839, 0.188931, 0.235828, 0.368505, 0.022879, 0.123644, 0.591248, -0.021005, 0.099674, 0.291861, -0.161369, 0.733945, 0.273807, -0.359583, 0.214092, -0.456652, -1.297653, 0.346513, -0.122812, -0.240472, 0.143202, 0.272271, 0.245684, 0.238851, -0.093797, 0.031591, 0.720266, -0.251591, 0.249342, -0.470848, 0.009957, 0.516043, 0.039894, 0.466373, 0.14746 , 0.30734 , -0.511812, -0.455825, 0.161481, 0.038873]

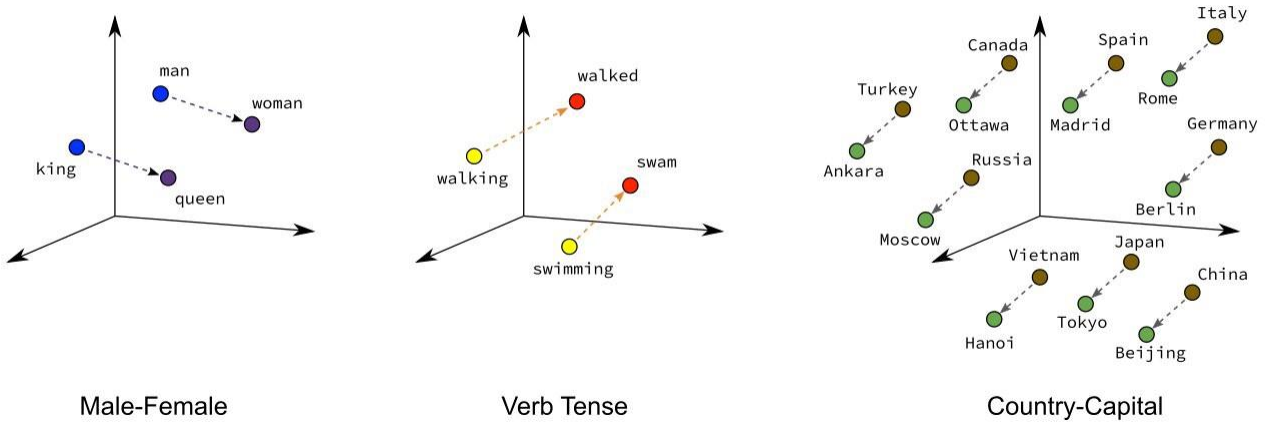
- We can compute the semantic similarity between words using operators such as cosine similarity

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|}$$

  - 'inteligência' ≈ 'perspicácia', 'clarividência', 'persuasão', 'intuição', 'sagacidade', 'astúcia', 'inteligencia', 'sabedoria', 'inteligências', 'habilidade'

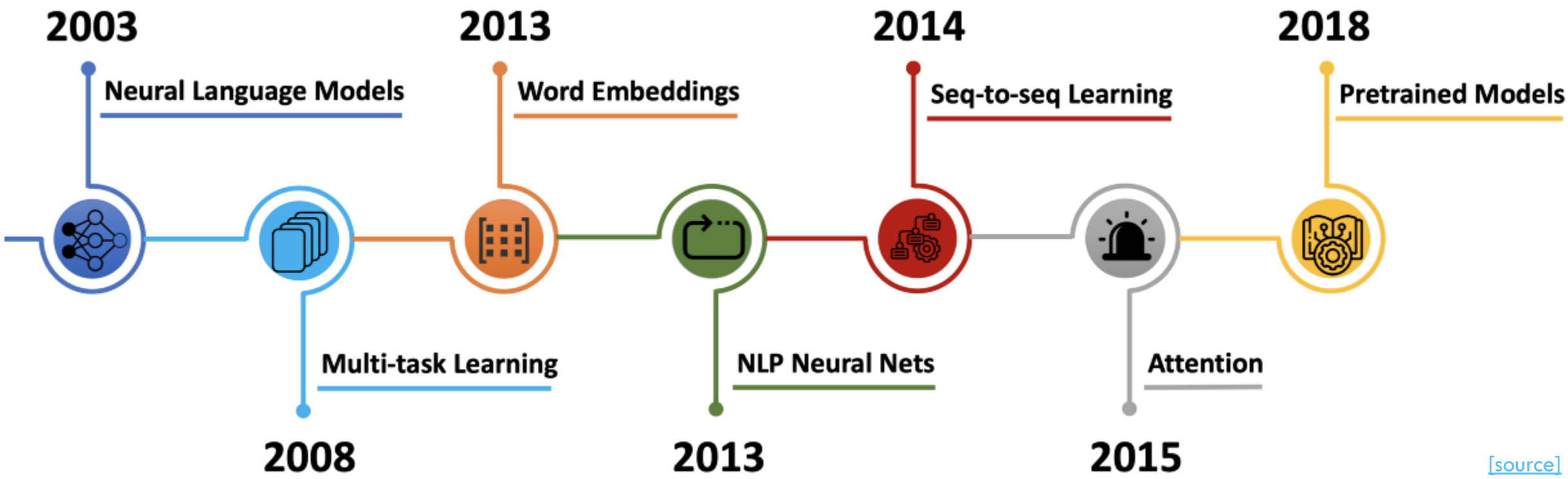
Semantic Properties of Embeddings

- Analogy: ability to capture relational meanings
  - The offsets between vector embeddings can capture some analogical relations between words
    - *king – man + woman ≈ queen*
    - *Paris – France + Italy ≈ Rome*



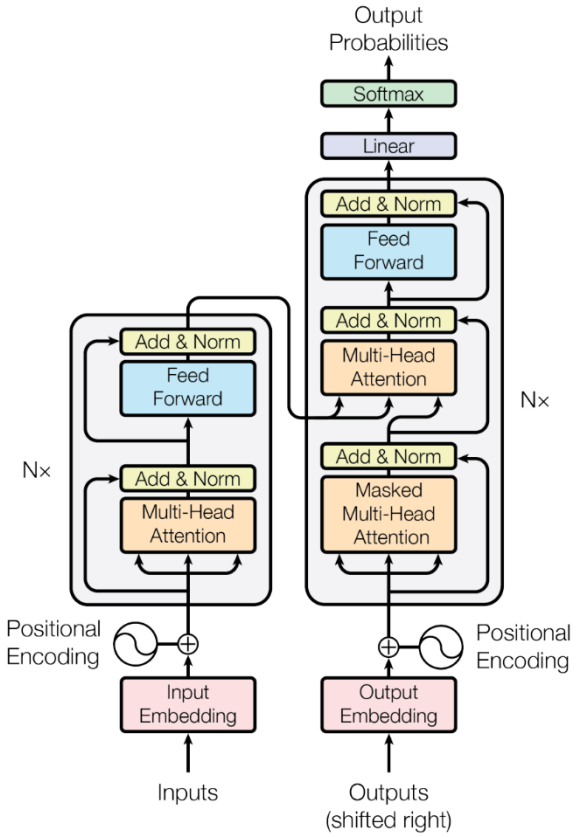
Male-Female Verb Tense Country-Capital

Neural History of NLP



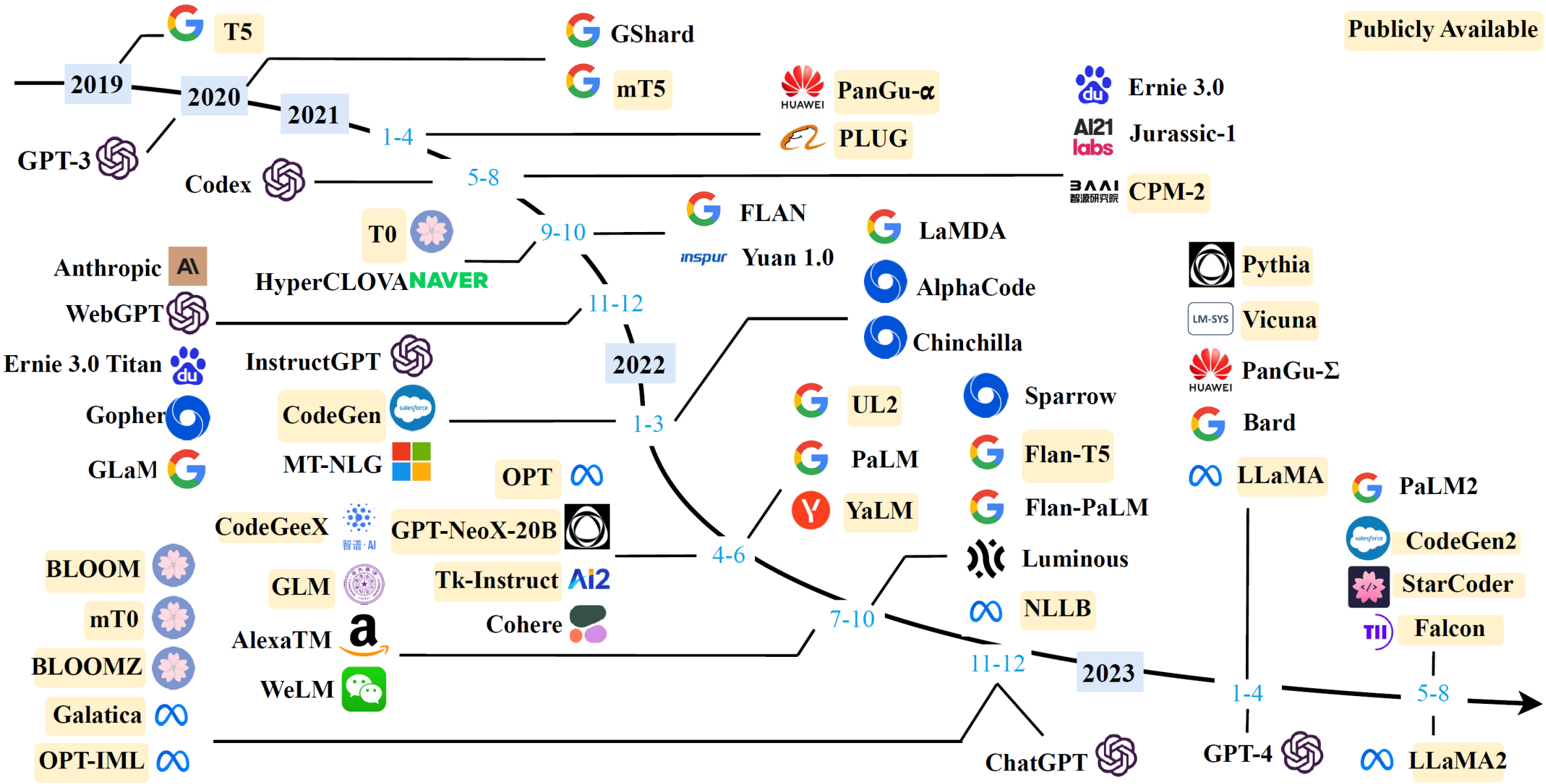
[source]

Recent Trends



[Vaswani et al., 2017]

- Transformer-based architectures
- Pre-trained Language Models: BERT, GPT, …
- Transfer Learning
  - fine-tuning, domain adaptation
  - cross-lingual learning
  - multi-task learning
- Prompt engineering



Zhao et al. (2023). **A Survey of Large Language Models.** *ArXiv*. /abs/2303.18223