A Practical Guide to Modern Natural Language Processing

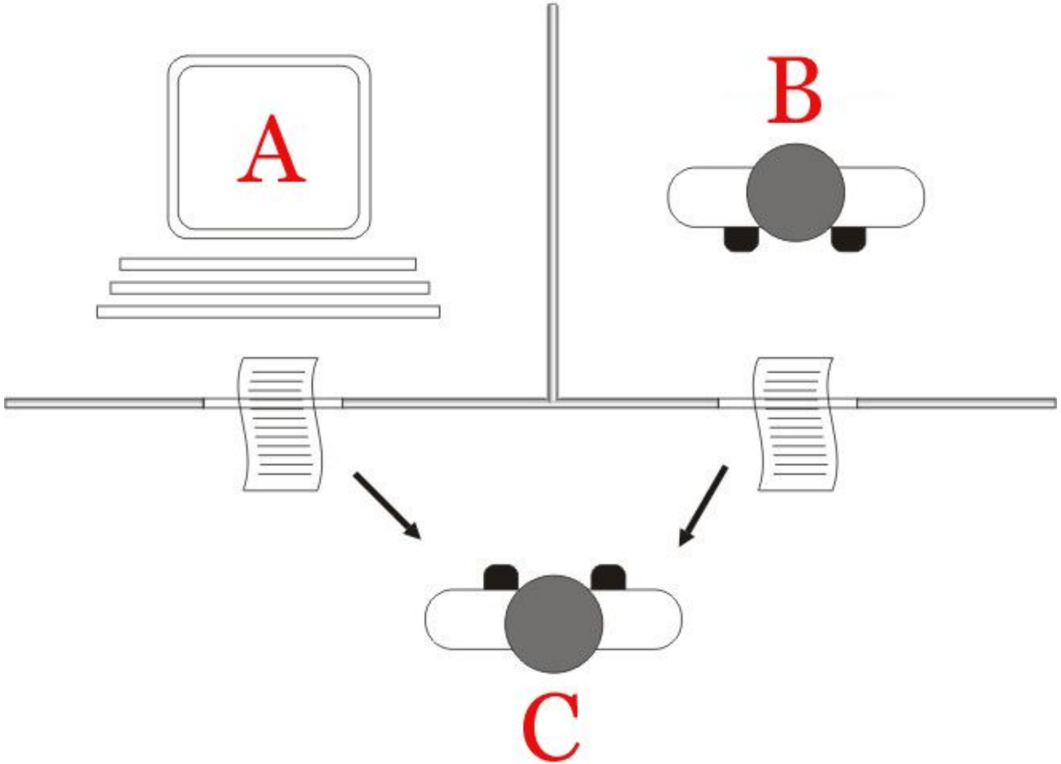
Cristian Schuszter

CERN School of Computing

Outline

1. **Ancient history**
2. Common tasks
3. Classical models & applications
4. Feature engineering
5. The deep learning years
6. State of the art
7. Q & A

The imitation game

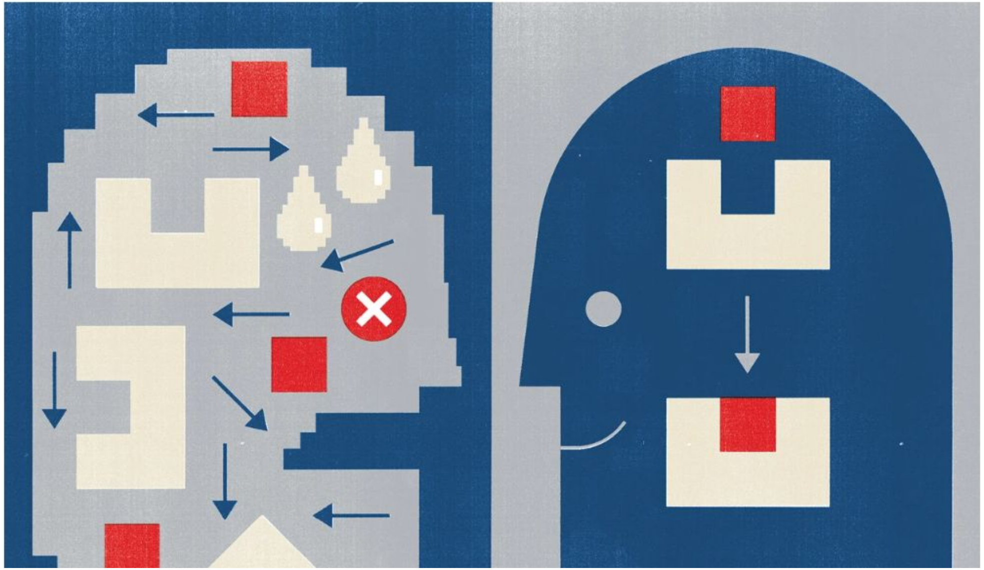


NEWS FEATURE | 25 July 2023

ChatGPT broke the Turing test — the race is on for new ways to assess AI

Large language models mimic human chatter, but scientists disagree on their ability to reason.

By Celeste Biever



😞😞😞

What’s going on behind the scenes

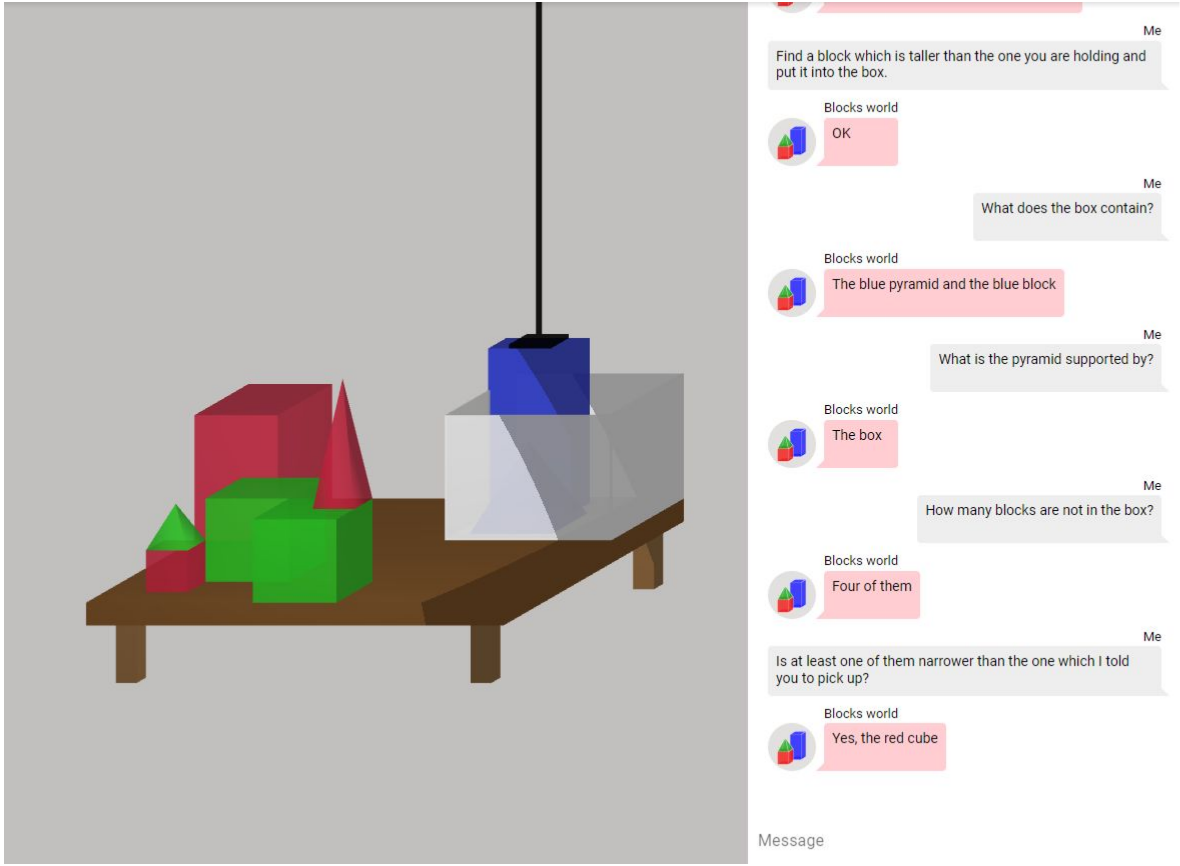
The dark ages - rule-based systems

```
  Welcome to
                    EEEEEE  LL      IIII   ZZZZZZ   AAAAA
                    EE      LL       II       ZZ   AA   AA
                    EEEEE   LL       II     ZZZ    AAAAAAA
                    EE      LL       II    ZZ      AA   AA
                    EEEEEE  LLLLLL  IIII ZZZZZZ    AA   AA

  Eliza is a mock Rogerian psychotherapist.
  The original program was described by Joseph Weizenbaum i
  This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

```
["alike", 10, [
 ["*", [
     "In what way ?",
     "What resemblence do you see ?",
     "What does that similarity suggest to you ?",
     "What other connections do you see ?",
     "What do you suppose that resemblence means ?",
     "What is the connection, do you suppose ?",
     "Could there really be some connection ?",
     "How ?"
  ]]
]],
```

https://github.com/oren/eliza-bot

The dark ages - rule-based systems (2)



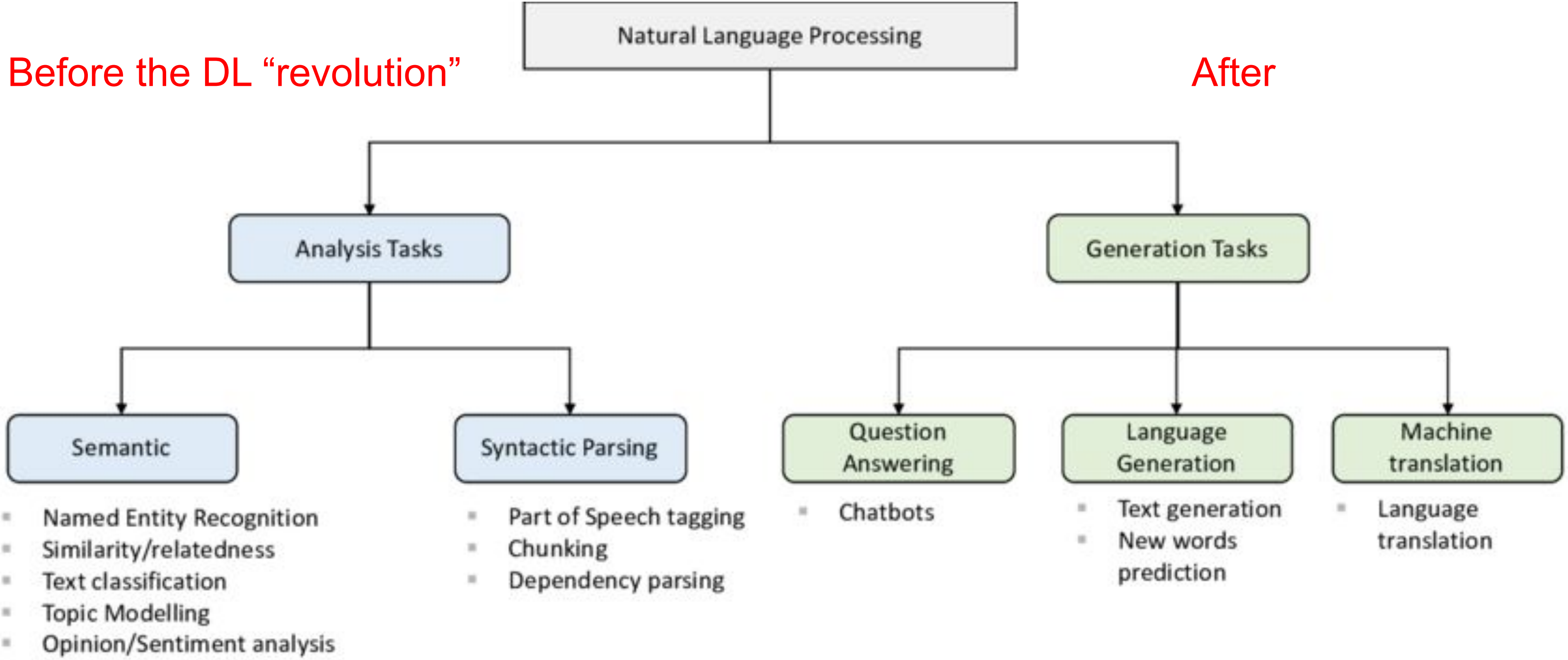
https://patrickvanbergen.com/blocks-world/

Outline

1. Ancient history
2. **Common tasks**
3. Classical models & applications
4. Feature engineering
5. The deep learning years
6. State of the art
7. Q & A

Before the DL “revolution”

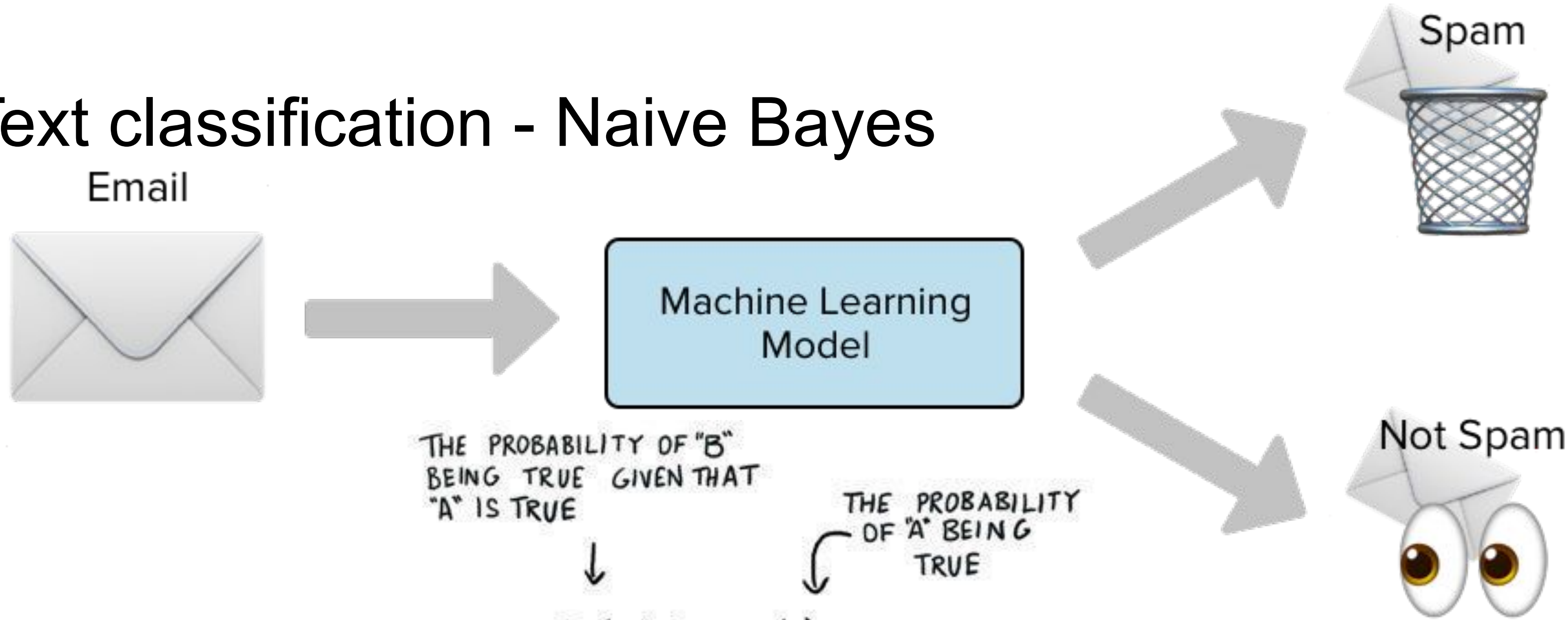
After



Outline

1. Ancient history
2. Common tasks
3. **Classical models & applications**
4. Feature engineering
5. The deep learning years
6. State of the art
7. Q & A

Text classification - Naive Bayes



THE PROBABILITY OF "B" BEING TRUE GIVEN THAT "A" IS TRUE

THE PROBABILITY OF "A" BEING TRUE

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

THE PROBABILITY OF "A" BEING TRUE GIVEN THAT "B" IS TRUE

THE PROBABILITY OF "B" BEING TRUE

Naive Bayes

Naive = words in the sentence are independent

Spam example:

("Hello, this is a normal email", 0),
("Congratulations! You've won a free vacation", 1),
("Click here to claim your prize", 1),
("Meeting scheduled for tomorrow", 0),
("Limited time offer, buy now", 1)

P(spam) = 3/5
P(ham) = 2/5

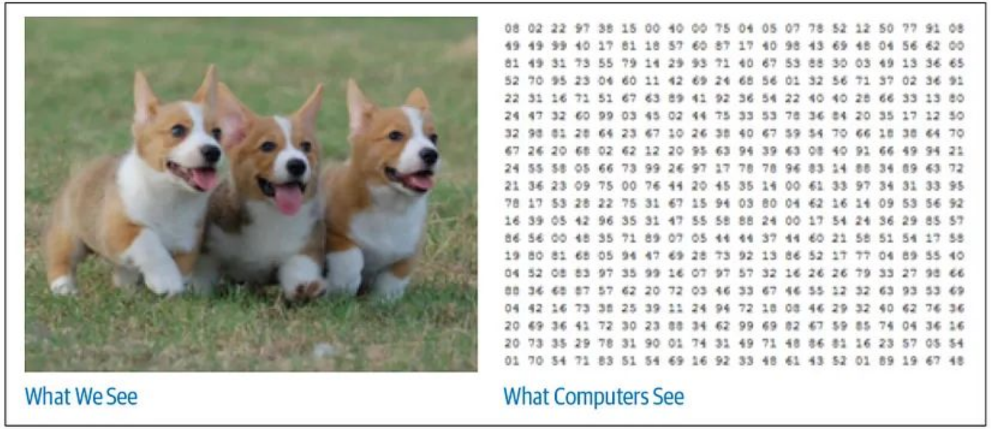
$$P(SPAM \mid W) = \frac{P(W \mid SPAM) * P(SPAM)}{P(W \mid SPAM) * P(SPAM) + P(W \mid HAM) * P(HAM)}$$

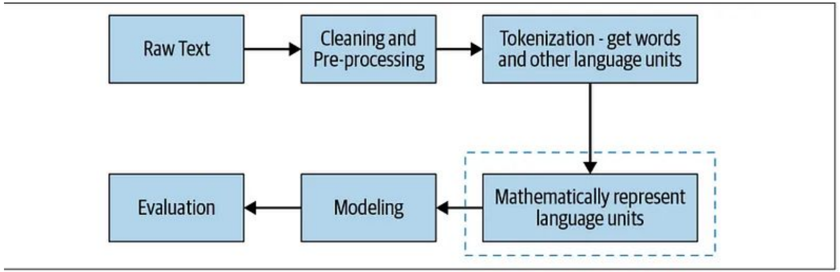
Outline

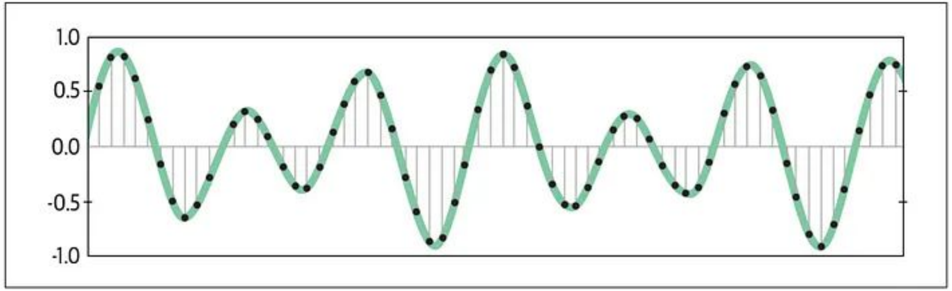
1. Ancient history
2. Common tasks
3. Classical models & applications
4. **Feature engineering**
5. The deep learning years
6. State of the art
7. Q & A

Feature engineering

- Highly essential for building useful ML algorithms
- Range from simple to probabilistic distributions





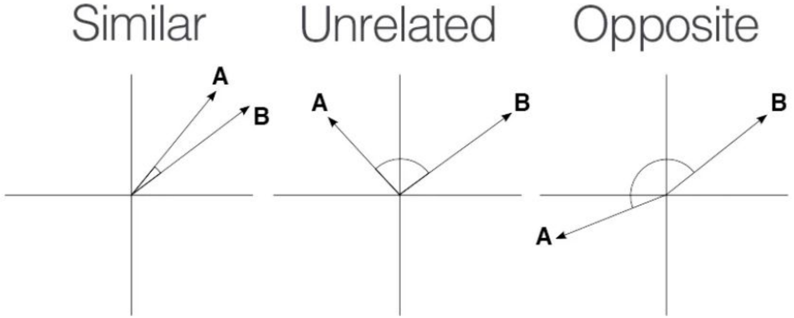


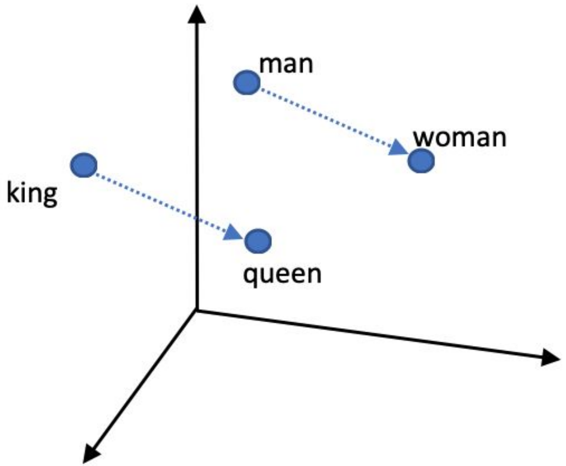

```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41,
-169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448,
-397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451,
1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461,
4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499,
-488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148,
-1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325,
350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```

Vector space models for text

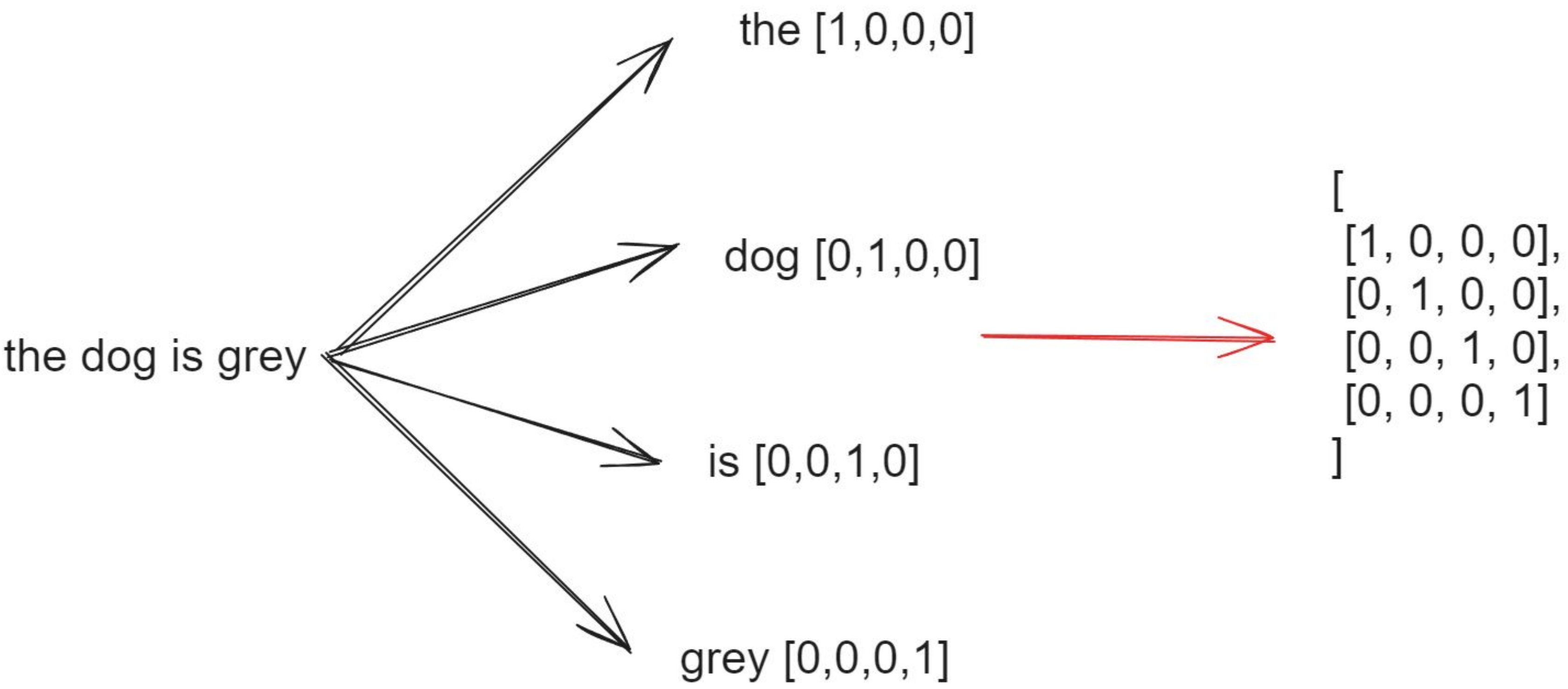
The formula for cosine similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

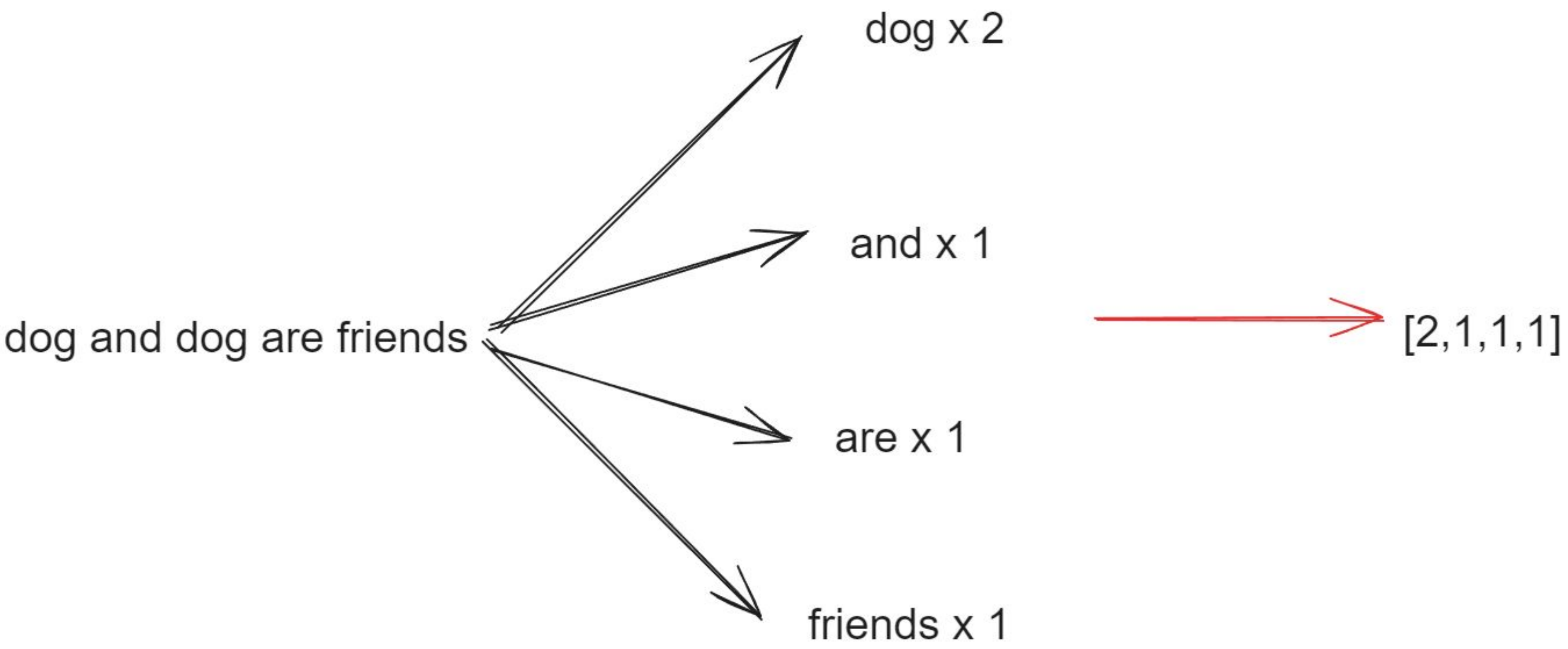




One-hot encoding



Bag of words



Bag of n-grams

2-grams are called “bigrams”

To be, or not to be, that is the question $\longrightarrow$

{("To","be"),
("be","or"),
("or","not"),
("not","to"),
("to","be"),
("be","that"),
("that","is"),
("is","the"),
("the","question")}

TF-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Term Frequency - Inverse Document Frequency

Example:

Our document contains 100 words, 3 times **cat.**

*TF(cat) = 3 / 100 = **0.03***

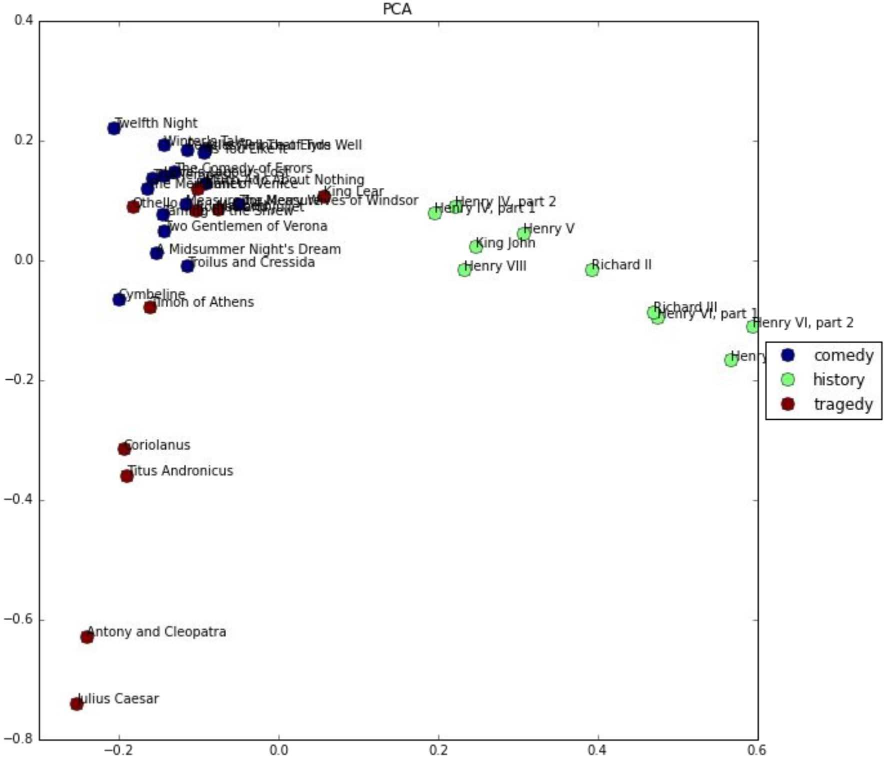
The database contains 10 million documents, and cat is in 1000 of those.

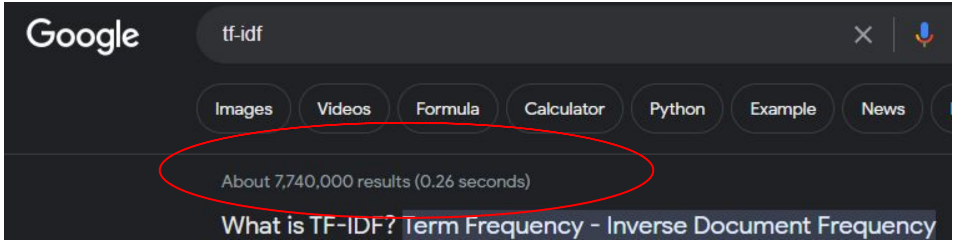
*IDF(cat) = log10(10.000.000 / 1000) = **4***

*TF_IDF(cat) = TF(cat) * IDF(cat) = 4 * 0.03 = **0.12***

What can I do with this?

- Topic modelling
  - Compute TF-IDF vectors of the documents in your corpus
  - Use a dimensionality reduction technique to visualize them
- Simple text classification
- Simple search engines
  - Pre-compute tf-idf for all your documents
  - Query comes in -> compute TF-IDF for the query
  - Find top k documents



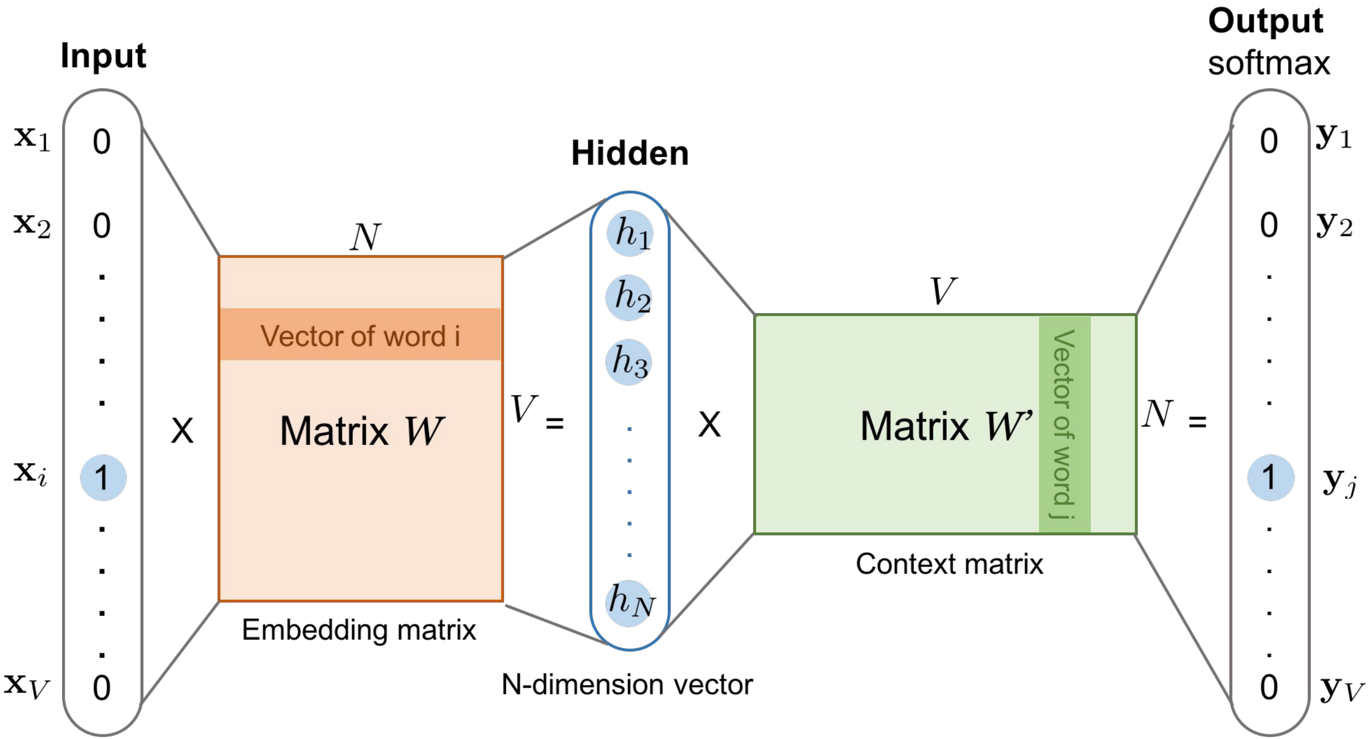


Outline

1. Ancient history
2. Common tasks
3. Classical models & applications
4. Feature engineering
5. **The deep learning years**
6. State of the art
7. Q & A

But how can we capture meaning?

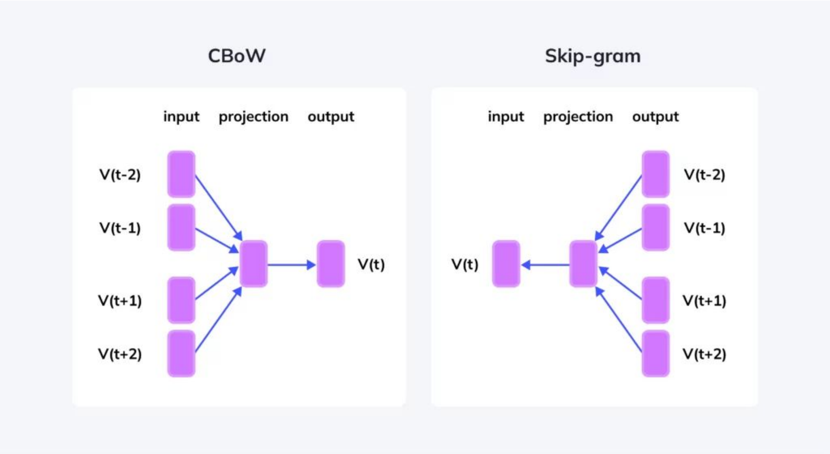
Feature engineering 2.0 - using deep learning



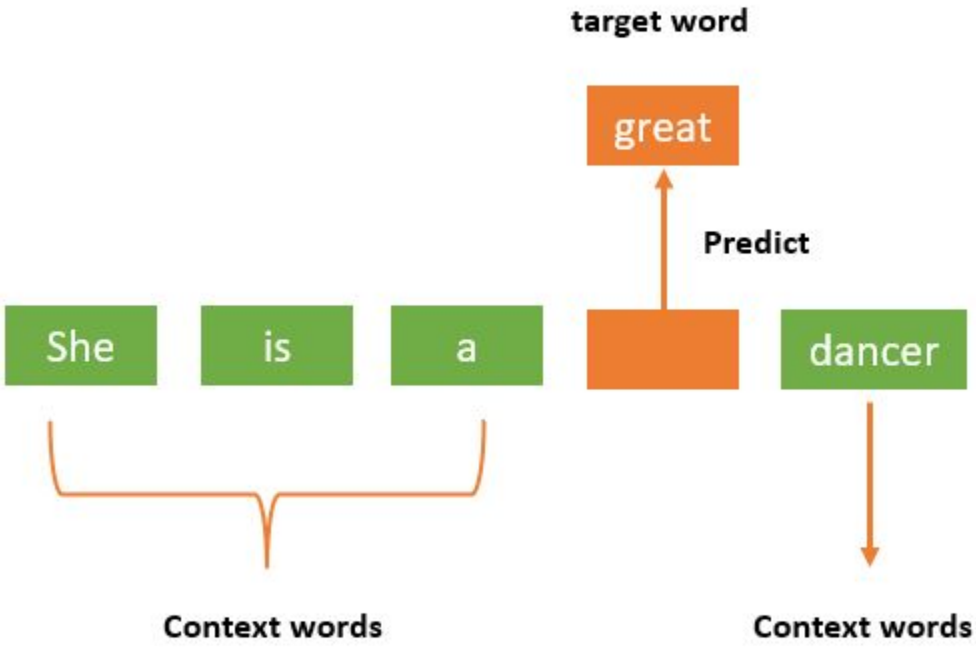
Word2Vec

Intuition: Given a document, for each word represent

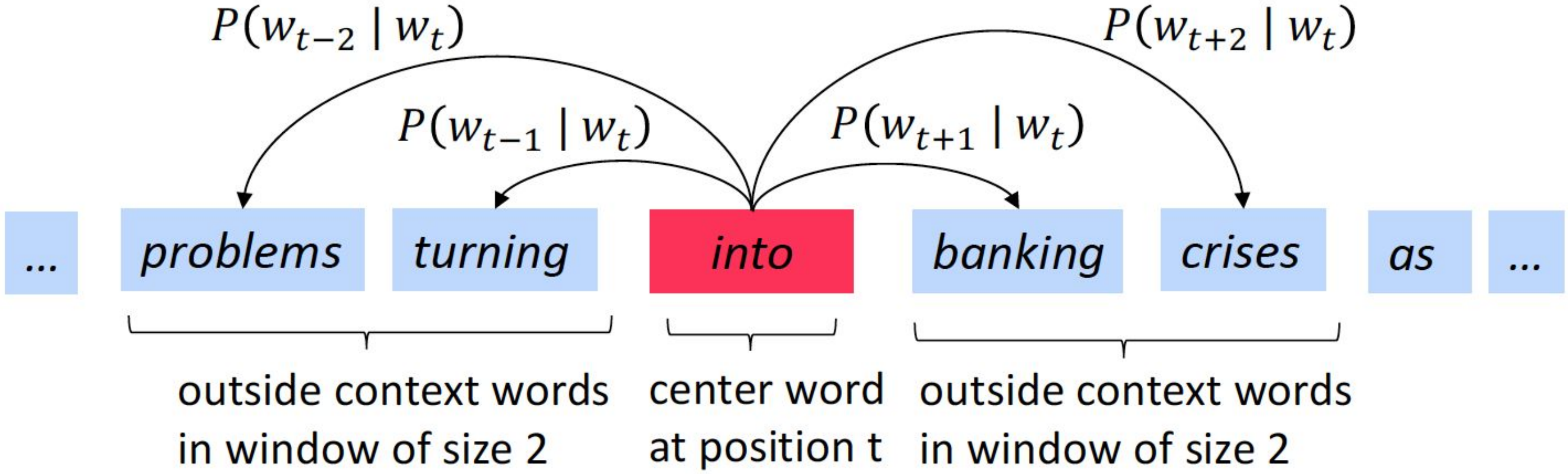
- The probability of obtaining **a word** given the context around it (CBOW)
  - The sum of all the context should give this word as the most likely
- The probability of obtaining the context words given a word (Skip-gram)
  - More computationally expensive, better results in training
- Back-propagate (train the model) and adjust the representation of the word



Word2Vec: CBOW



Word2Vec: Skip-gram



Similar algorithms

- GloVe: Global Vectors for Word Representation
  - Using global word-to-word co-occurence in the corpus for training
  - Gradient descent for training
- Fasttext
  - Solves the problem of OOV (out of vocabulary) words.
  - Uses sub-words in the training and embedding process to make “educated guesses”

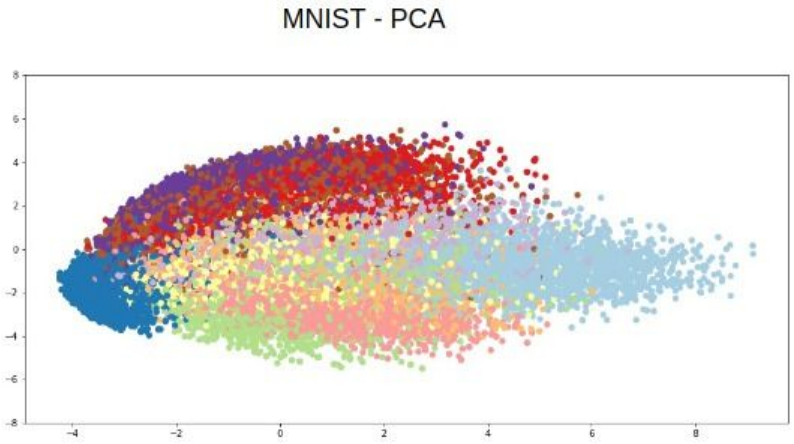
	anarchy	chy	<anar	narchy
	monarchy	monarc	chy	<monar
	kindness	ness>	ness	kind
	politeness	polite	ness>	eness>
EN	unlucky	<un	cky>	nlucky
	lifetime	life	<life	time
	starfish	fish	fish>	star
	submarine	marine	sub	marin
	transform	trans	<trans	form

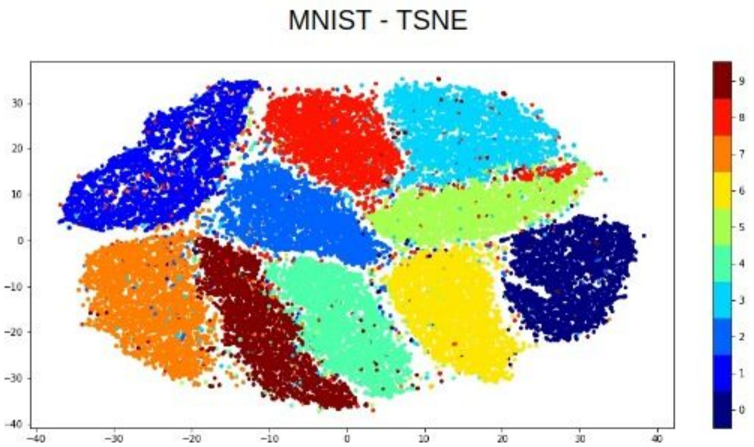
Visualizing your word embeddings

Problem: High-dimensional data (300+ dimensions)

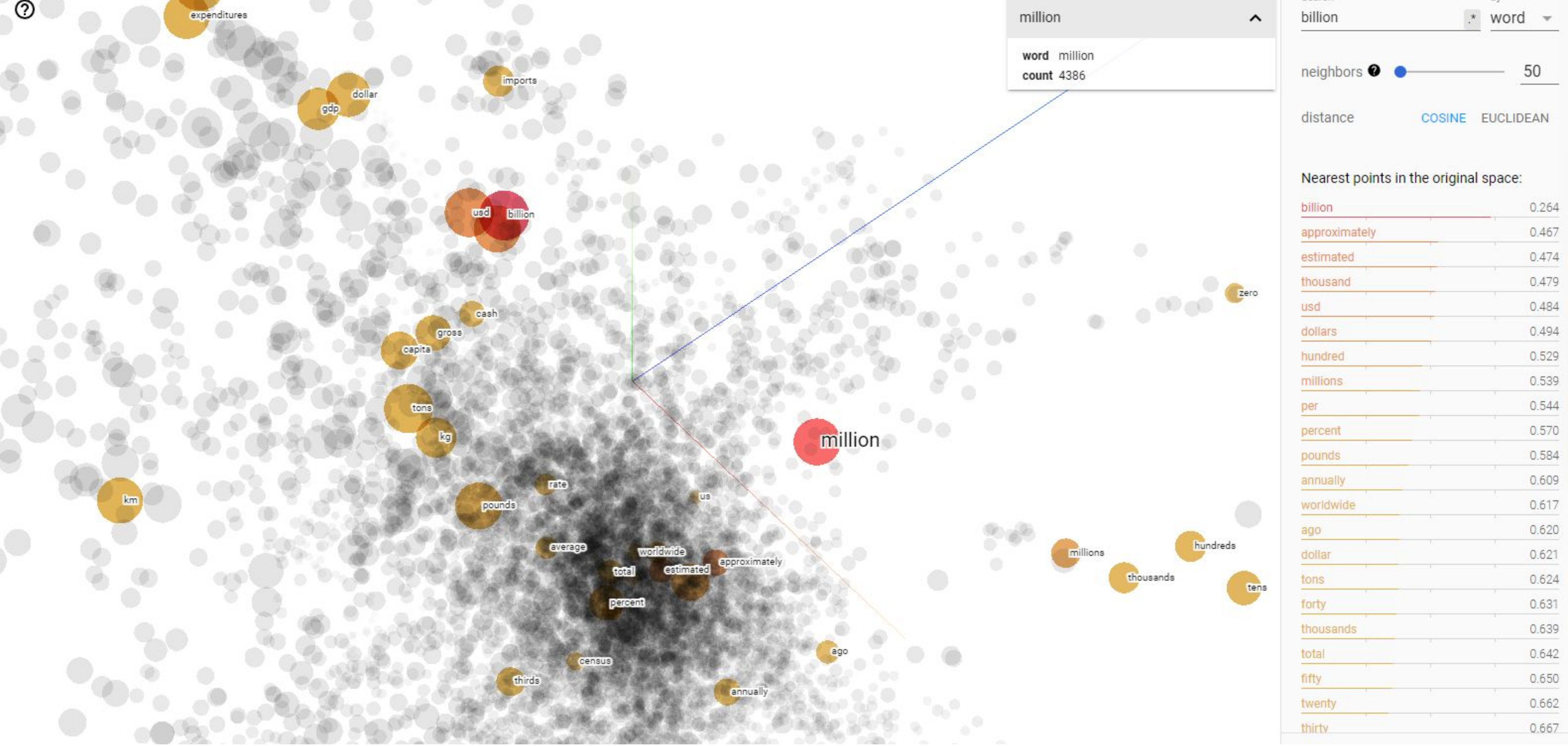
How do we get over this? **Dimensionality reduction techniques**

Most popular: Principal Component Analysis (PCA) & tSNE





https://distill.pub/2016/misread-tsne/



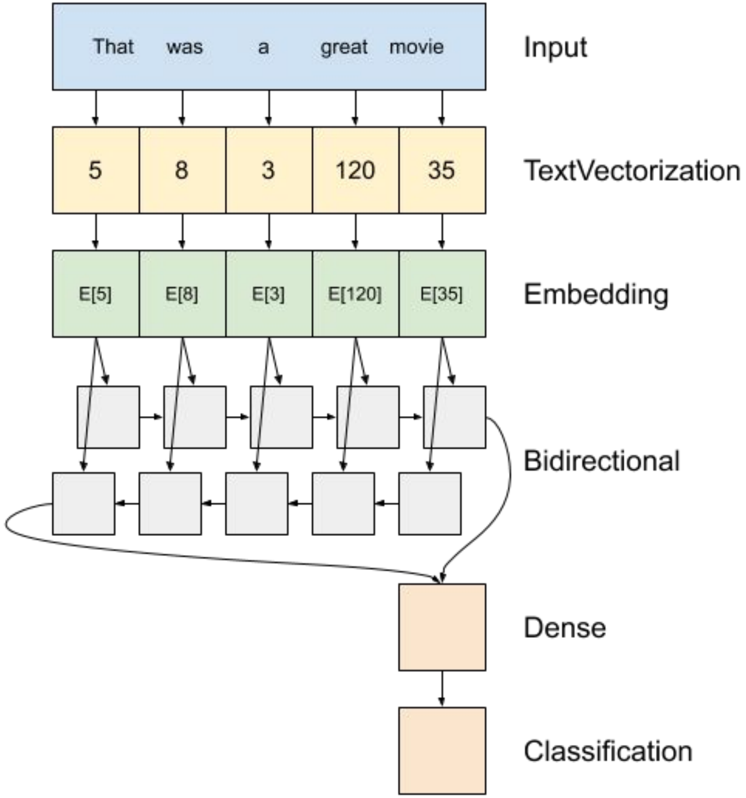
https://projector.tensorflow.org/

Deep learning models

- Mainly 2 use-cases:
  - Classification
  - Language modeling (question answering, translation)
- Much more performant than classical techniques

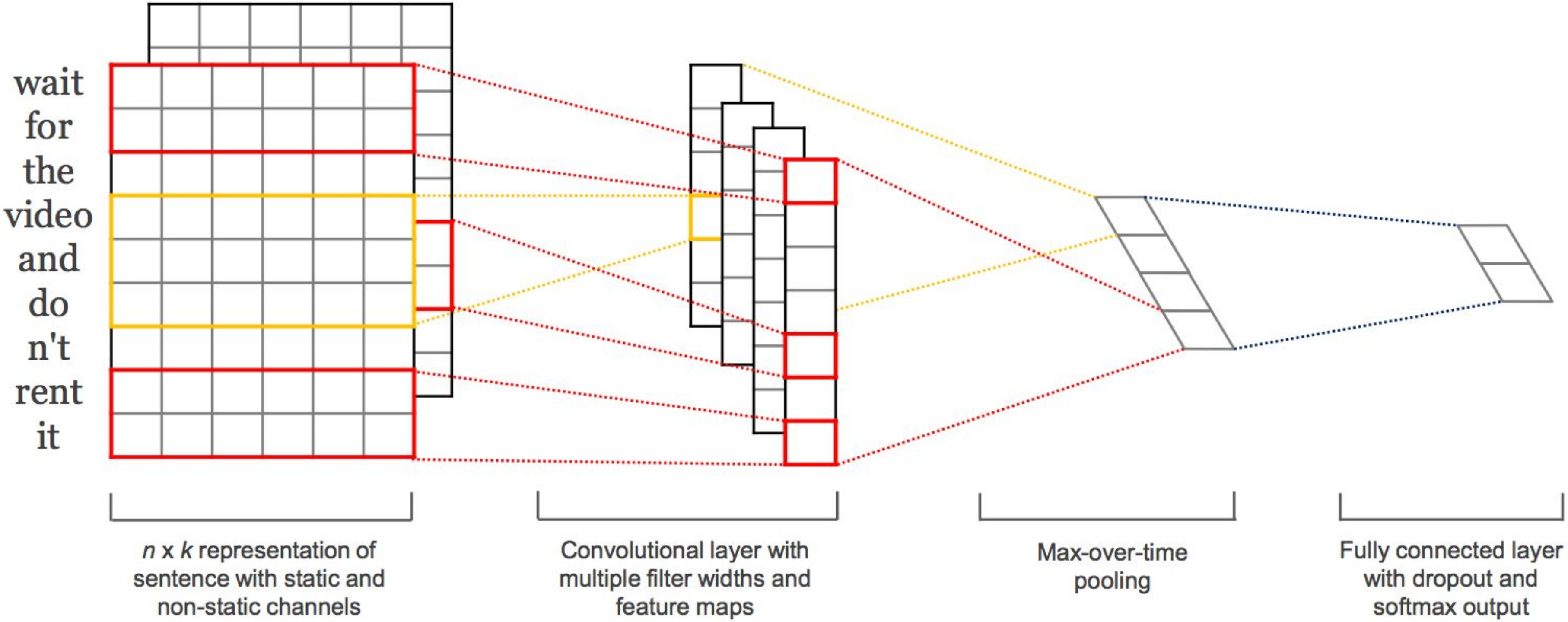
Caveats

- Computationally expensive to train
  - Many parameters for the weights
  - Hard to parallelize
- Doesn't do well on long inputs



Example classification architecture

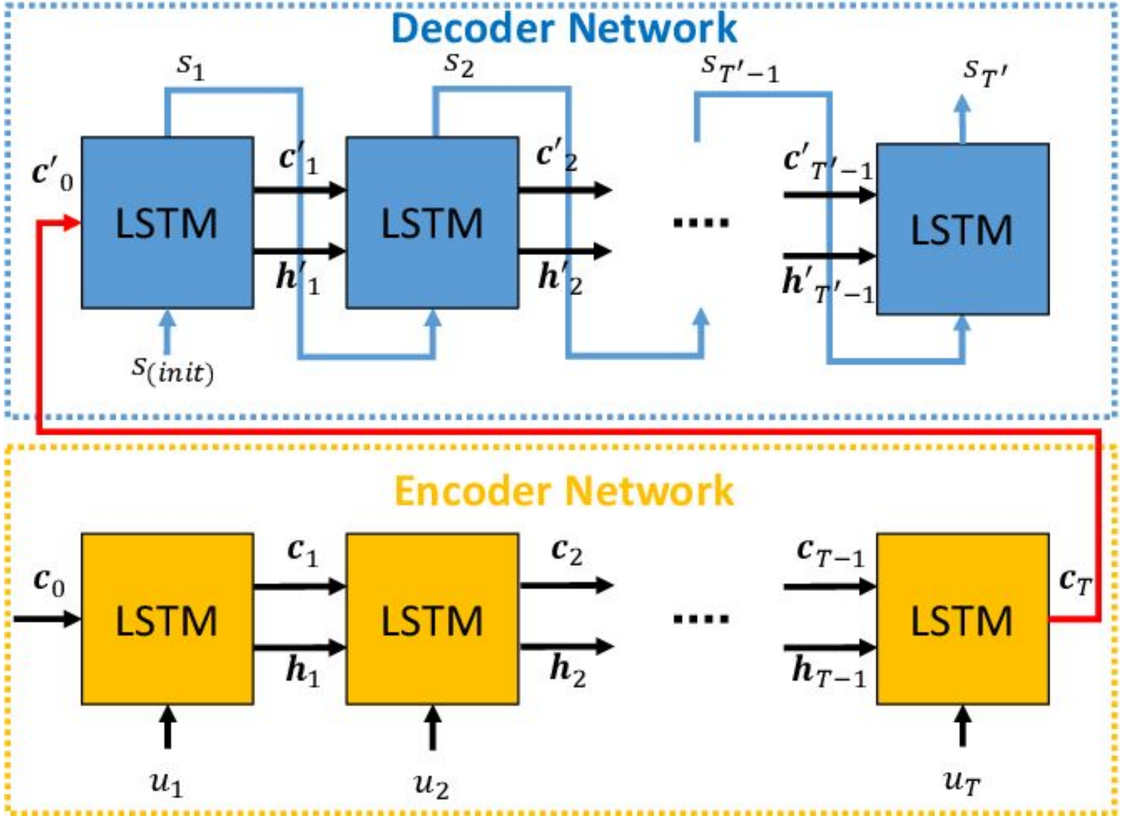
Text CNN



https://arxiv.org/abs/1408.5882

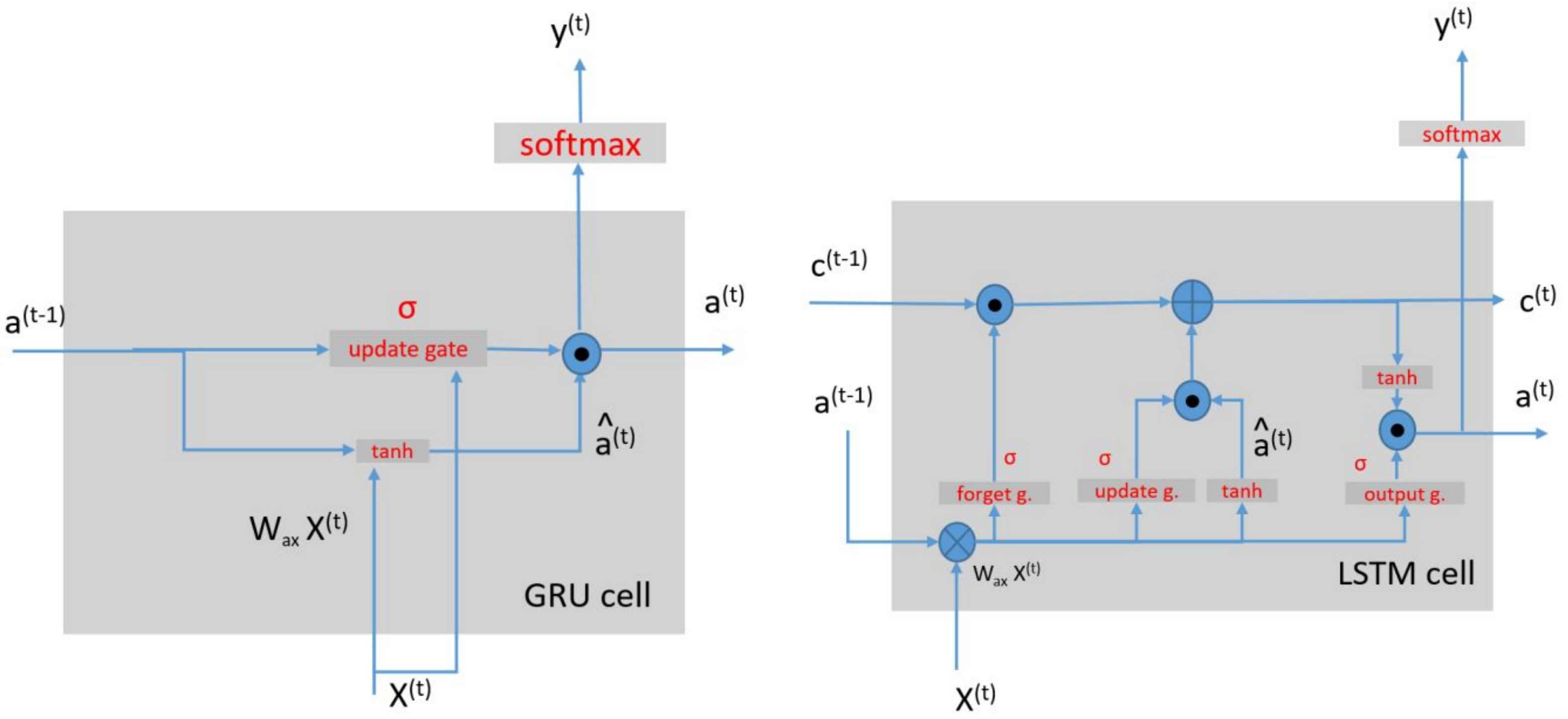
Encoder - decoder models - RNN, GRU, LSTM

I am a student



Je suis étudiant

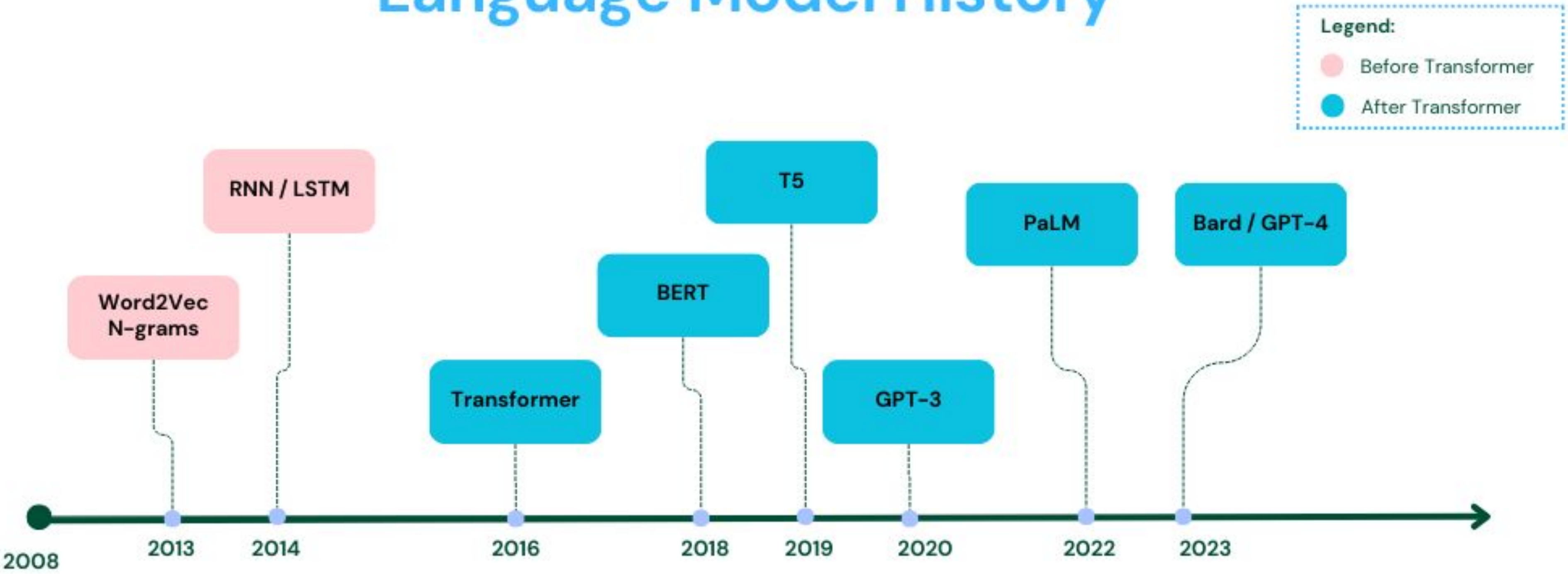
GRU vs LSTM cells



Outline

1. Ancient history
2. Common tasks
3. Classical models & applications
4. Feature engineering
5. The deep learning years
6. **State of the art**
7. Q & A

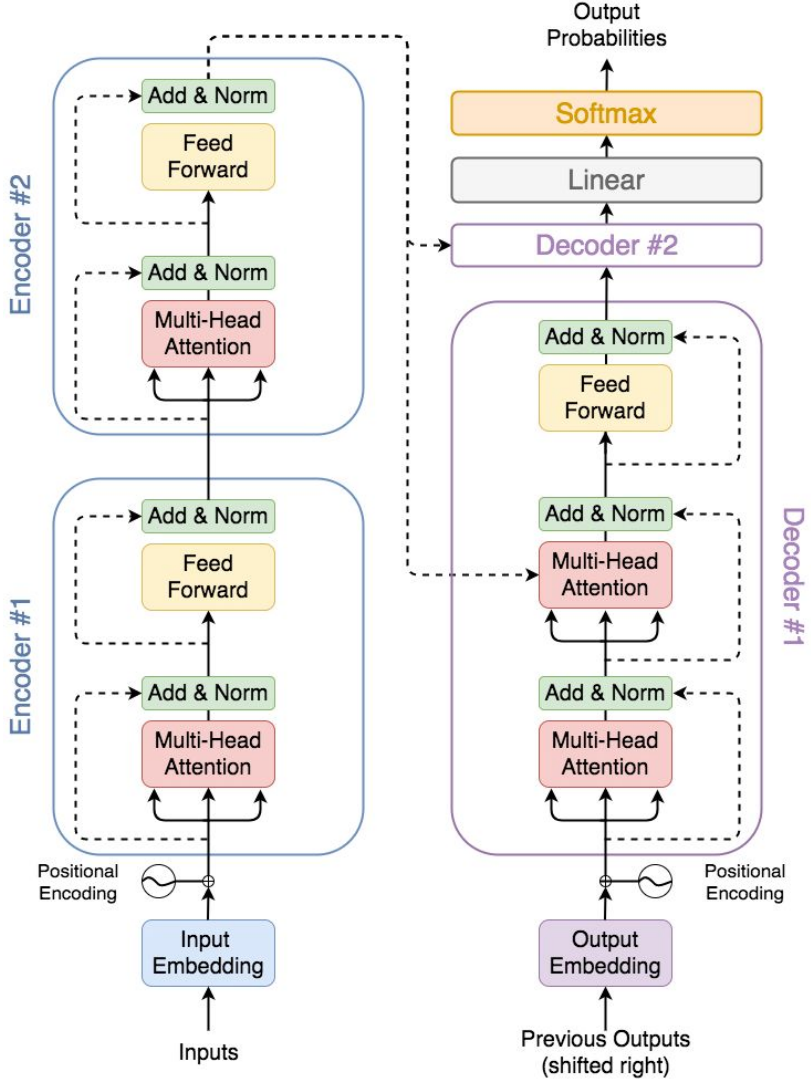
Language Model History



Transformers

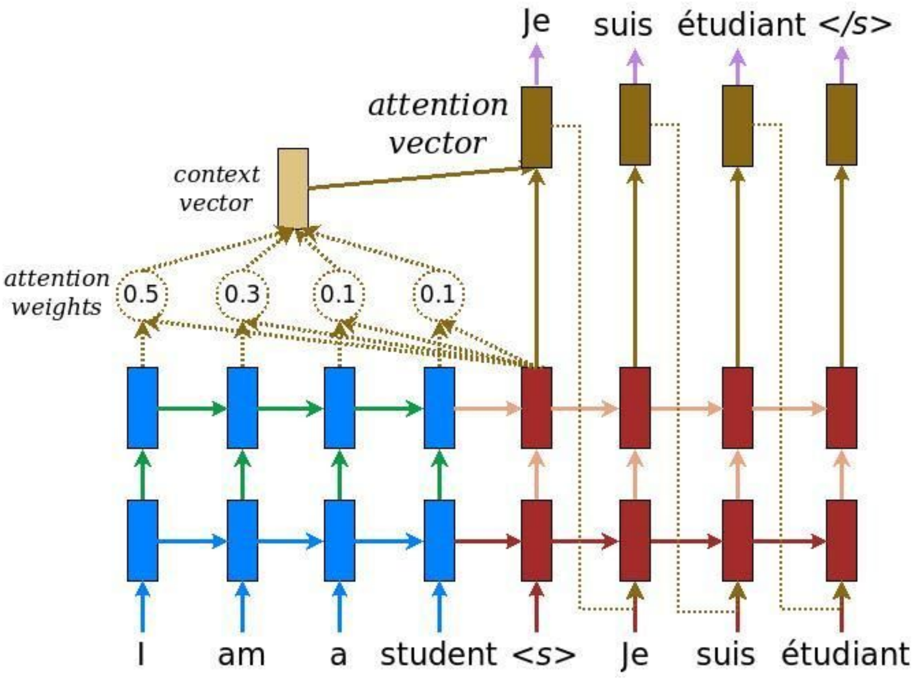
- Recurrent / convolutional models are really slow to train and don't scale

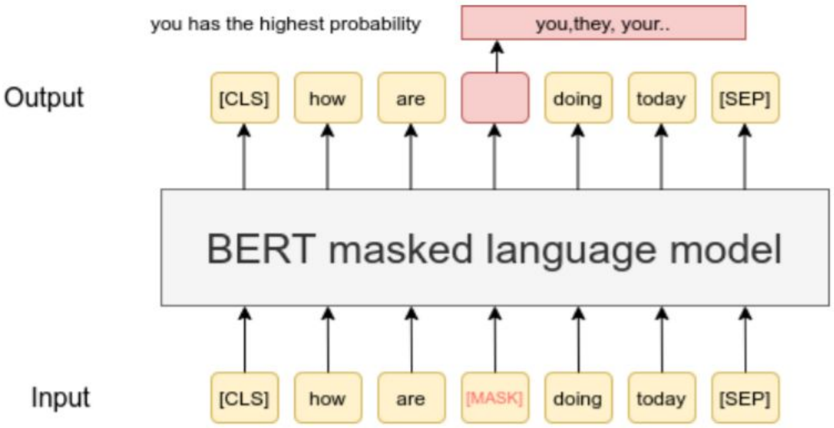
- Intuitive example: translation
    - Language A gets encoded
    - Gets decoded into sentence for Language B



https://arxiv.org/abs/1706.03762

Attention & masked language model training





Thank you!

Contact:

- Emails:
  - Work: cristian.schuszter@cern.ch
  - Not work: chrisschuszter@gmail.com

cschuszter

saibot94