

# W08D4

## NLP II

Instructor: Eric Elmoznino

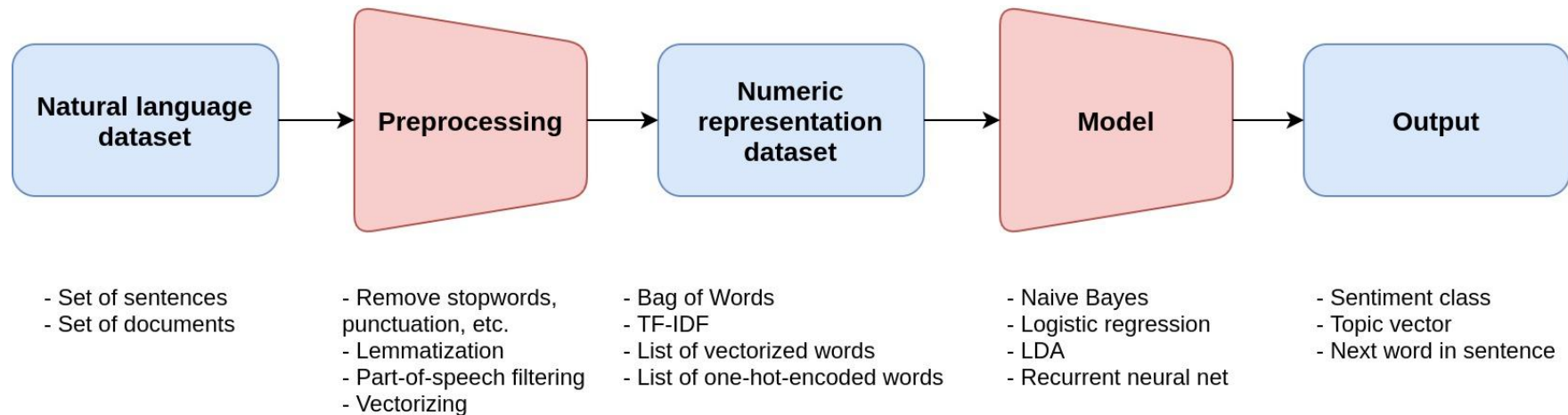
# Outline for today

- Unified NLP framework
- Sentiment analysis
  - Demo using Naive Bayes
- Topic modeling
  - Latent Dirichlet Allocation (LDA)
  - Demo using LDA
- Language modeling (i.e. predict the next  $n$  words)
  - Sketch using deep neural networks
- Translation
  - Sketch using deep neural networks

# Unified NLP framework

# From dataset to output

- **Goal:** to extract information encoded in language
  - Supervised tasks: learn to predict annotated labels for sentences/documents
  - Unsupervised tasks: learn a task that implicitly requires some understanding of language



# Sentiment analysis

# Overview

*“I love this movie! It's sweet, but with satirical humor. The dialogs are great and the adventure scenes are fun. It manages to be romantic and whimsical while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I have seen it several times and I'm always happy to see it again.....”*



Negative



Neutral



Positive



## Challenges?

large +/- word vocabulary

syntax

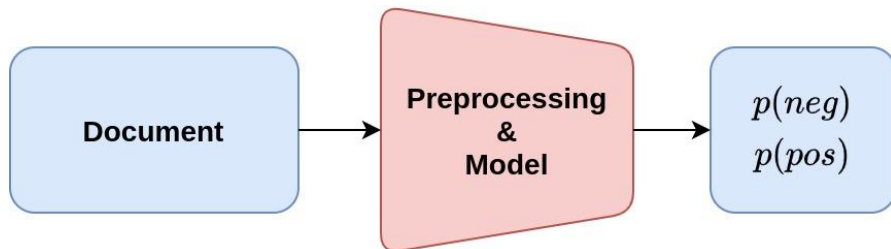
sarcasm

(long range) Negation

writing style

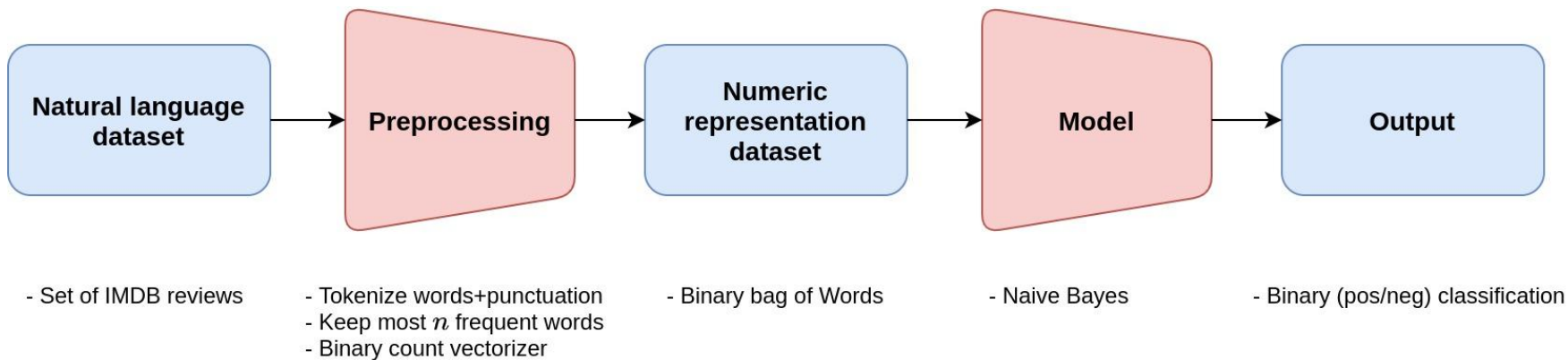
# Technical description

- **Input:** document, sentence, survey, voice recording, biometrics, etc.
- **Output:** binary classification, multi-class classification, univariate regression
- **Type of problem:** supervised learning
- **Datasets:** product reviews, customer service dialogue, social media, etc.
- **Use cases**: marketing, business strategy insights, brand monitoring, automated customer service actions, market predictions, etc.



# Demo

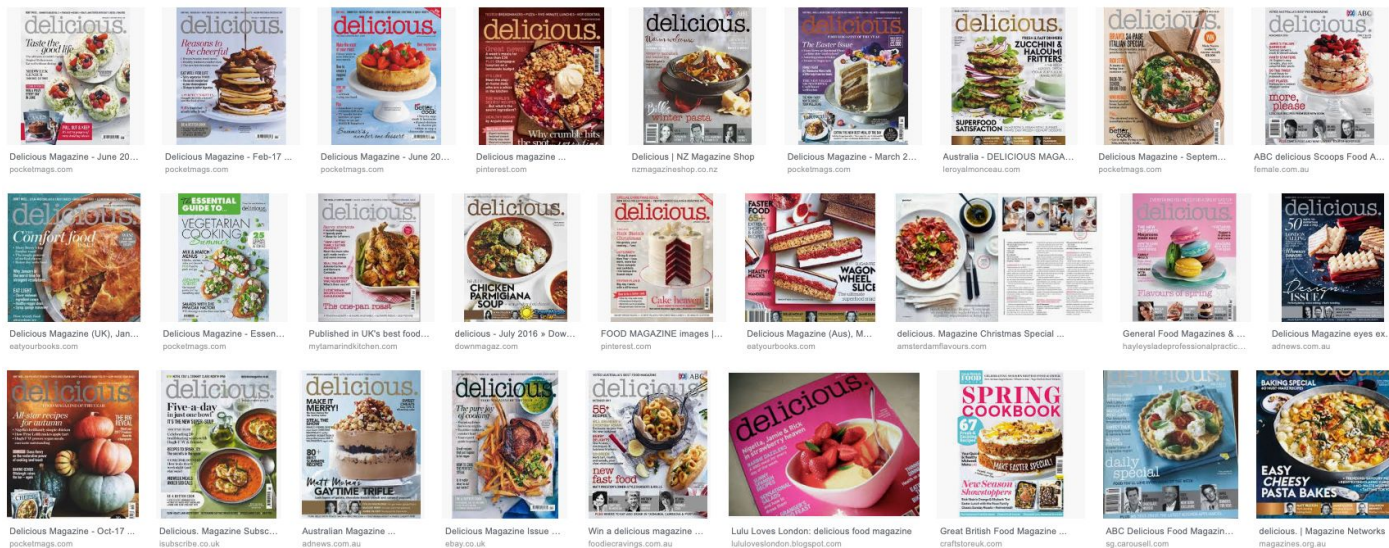
- [Repository](#)/sentiment\_analysis\_demo.ipynb





# Topic modeling

# Overview



## Challenges?

multiple word meanings

dataset-dependent

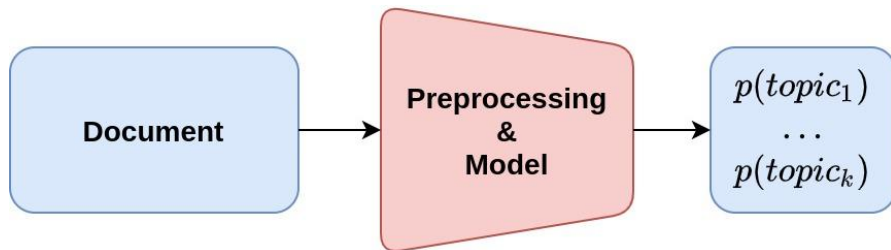
large documents

topic mixtures

domain knowledge

# Technical description

- **Input:** document or sentence
- **Output:** probabilities/scores over  $k$  topics
- **Type of problem:** unsupervised learning
- **Datasets:** articles, social media, product descriptions, company docs., etc.
- **Use cases:** marketing, business strategy insights, brand monitoring, searching a database, etc.



# Latent Dirichlet Allocation (LDA)

- Bayesian model: infer the hidden variables (topics) that generated the data
  - $p(\text{topics} \mid \text{document}) \propto p(\text{document} \mid \text{topics}) p(\text{topics})$
- Developed by [David Blei](#) — one of the most cited [papers](#) in the last 15 years
- Insight:
  - Each document is a mixture of corpus-wide topics (i.e. a probability distribution over topics)
  - Each topic is a mixture words (i.e. a probability distribution over words)

# Latent Dirichlet Allocation (LDA)

Every topic is a mixture words

"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

gene 0.04  
dna 0.02  
genetic 0.01  
...

life 0.02  
evolve 0.01  
organism 0.01  
...

brain 0.04  
neuron 0.02  
nerve 0.01  
...

data 0.02  
number 0.02  
computer 0.01  
...

Each document is a mixture of topics

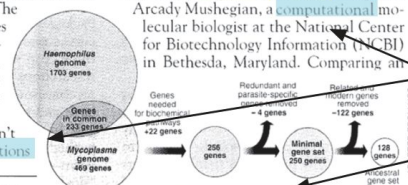
## Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to **survive**? Last week at the genome meeting here,\* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those **predictions**

"are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a **genetic numbers** game, particularly as more and more **genomes** are completely mapped and sequenced. "It may be a way of organizing any newly **sequenced genome**," explains

Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



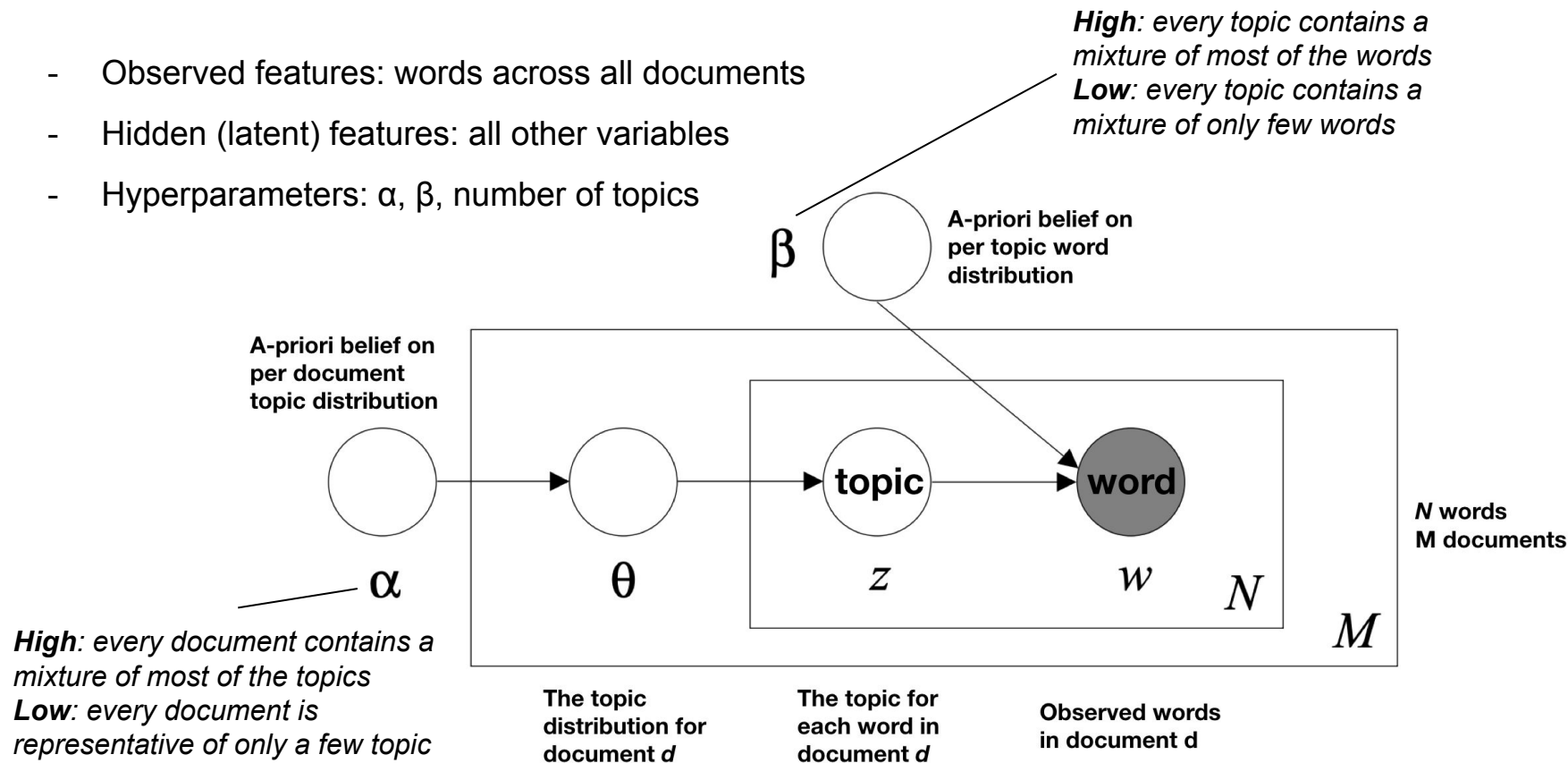
\* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

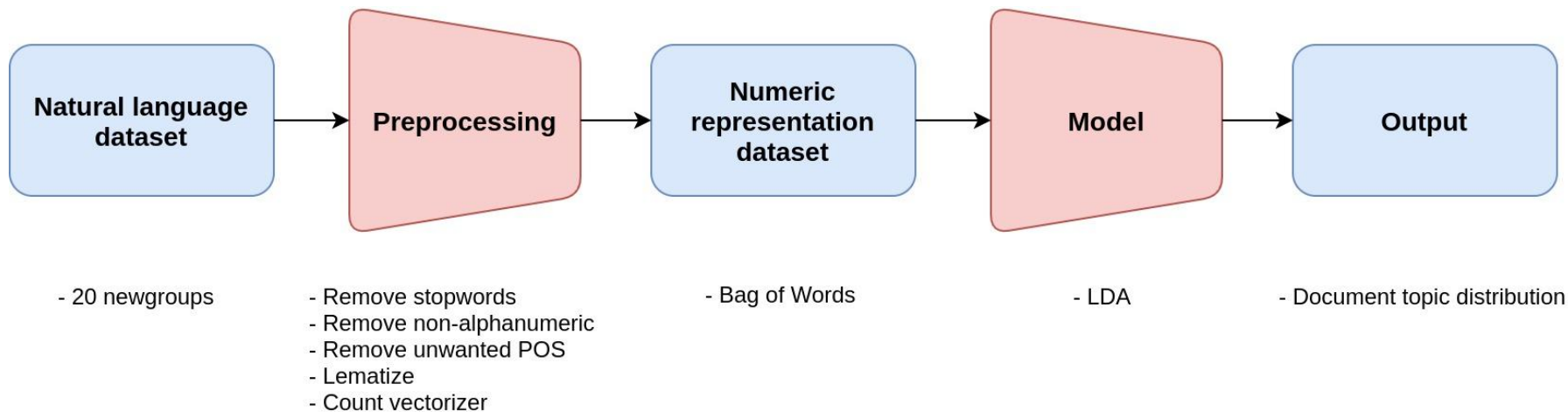
# Latent Dirichlet Allocation (LDA)

- Observed features: words across all documents
- Hidden (latent) features: all other variables
- Hyperparameters:  $\alpha$ ,  $\beta$ , number of topics



# Demo

- [Repository](#)/topic\_modeling\_demo.ipynb



# Language modeling



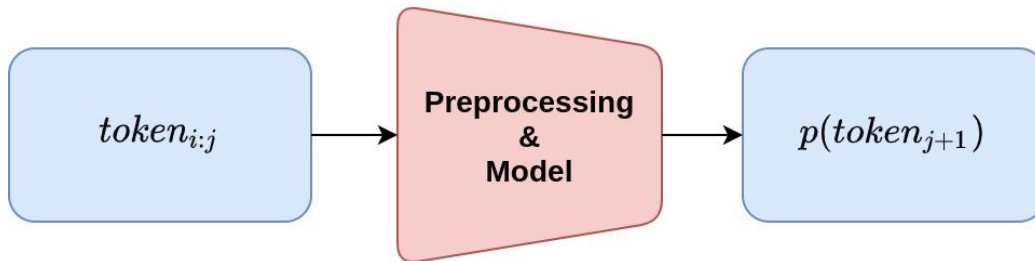
# Overview

*“The children went to play at the \_\_\_\_”*

- What is required to solve this task?
  - Word meanings/associations (*and not just dictionary surface-level meanings*)
  - Phrase/sentence meanings
  - Subject/object
  - Syntax
  - Part-of-speech (e.g. noun, verb, adjective, etc.)

# Technical description

- **Input:** sequence of words (either a fixed-length sequence, or must use a model that can handle variable-lengths)
- **Output:** multi-class classification (number of classes = size of vocabulary)
- **Type of problem:** supervised learning (but without having to annotate)
- **Datasets:** *any* corpus of text
- **Use cases:** representation/transfer learning, typing assistance

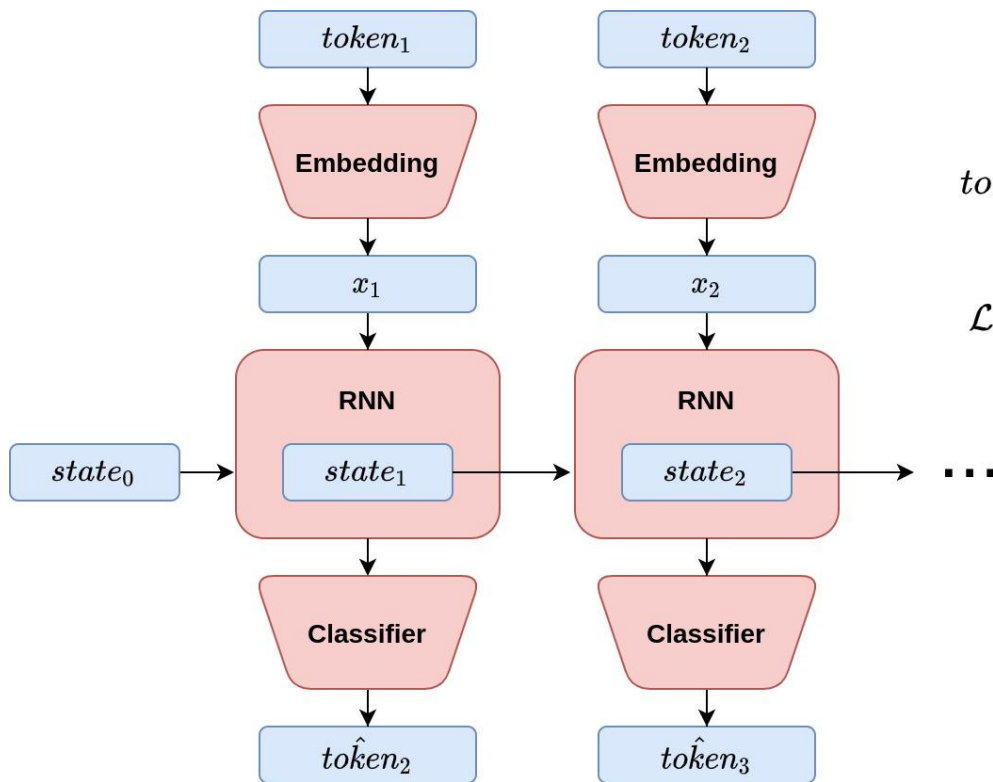


# Example: recurrent neural network intuition

*“The children went to play at the \_\_\_\_”*

1. Read one word at a time, up to the last available word
2. With each word you read, update your memory of the meaning and syntax
3. Knowing the current meaning and syntax, predict what the next word will be

# Example: recurrent neural network



$$\begin{aligned}x_t &= \text{Embedding}(\text{token}_t) \\state_t &= \text{RNN}(x_t, \text{state}_{t-1}) \\ \hat{\text{token}}_{t+1} &= \text{Classifier}(\text{state}_t)\end{aligned}$$

$$\mathcal{L} = \sum_{t=2}^{t_f} \text{CrossEntropy}(\hat{\text{token}}_t, \text{token}_t)$$

Translation

# Overview

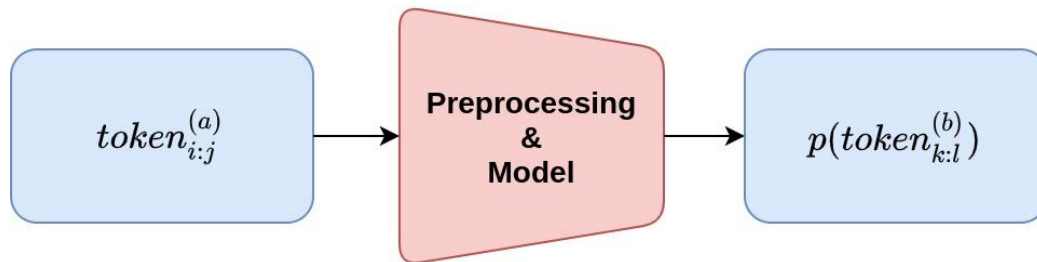
*“The children went to play at the park.”*

*“Les enfants sont allés jouer au parc.”*

- What is required to solve this task?
  - Meaning and syntax in both languages
- What are the challenges?
  - Ill-defined problem (i.e. no single correct answer)
  - Clean 1-to-1 dataset difficult to obtain
  - Very high-dimensional input *and* output (long sequences required for a legitimate translation)
  - Model must 2 languages *and* a transfer function between them

# Technical description

- **Input:** sequence of words (with variable length)
- **Output:** sequence of multi-class classifications (number of classes = size of vocabulary, sequence length = length of translated input)
- **Type of problem:** supervised learning
- **Datasets:** corpus of text, its translations, and annotated mapping
- **Use cases:** automatic translation



# Example: recurrent neural network intuition

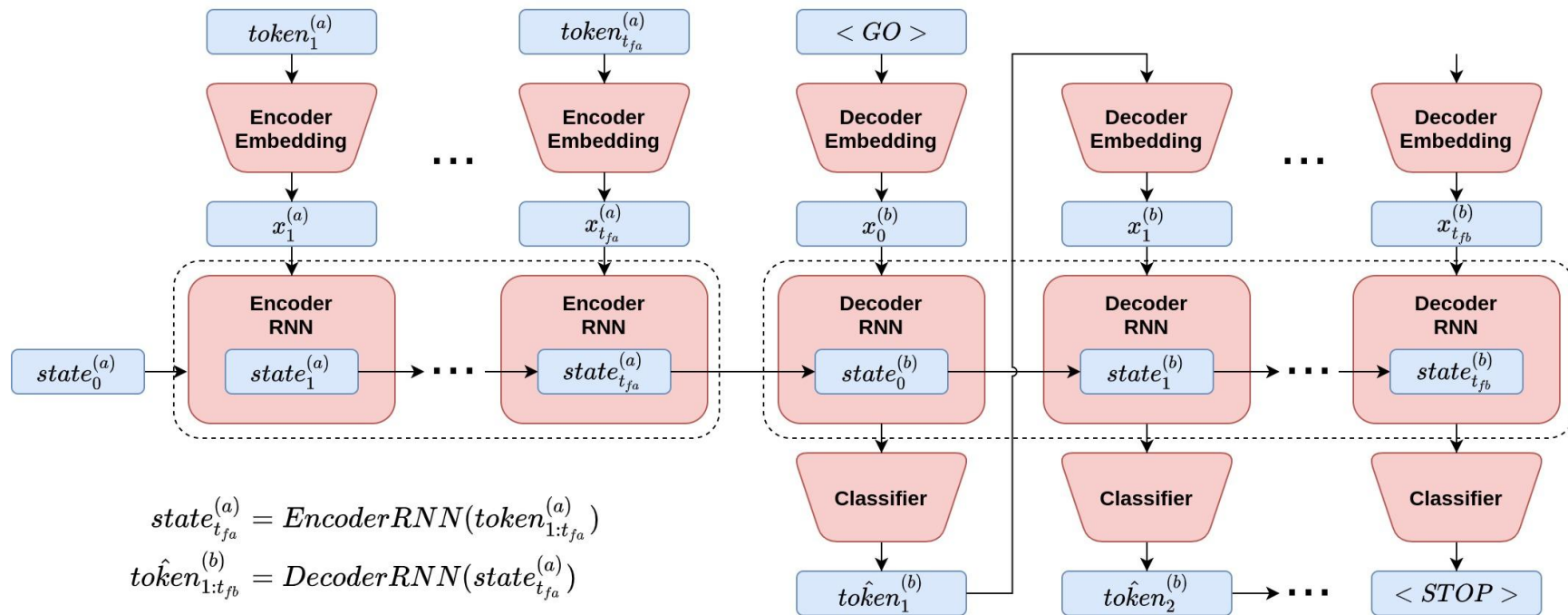
*“The children went to play at the park.”*

*“Les enfants sont allés jouer au parc.”*

1. Read one English word at a time, up to the end
2. With each word you read, update your memory of the meaning
3. Knowing the entire meaning, create the French translation one word at a time



# Example: recurrent neural network (seq2seq)



$$state_{t_{fa}}^{(a)} = \text{EncoderRNN}(token_{1:t_{fa}}^{(a)})$$
$$\hat{token}_{1:t_{fb}}^{(b)} = \text{DecoderRNN}(state_{t_{fa}}^{(a)})$$

$$\mathcal{L} = \sum_{t=1}^{t_{fb}} \text{CrossEntropy}(\hat{token}_t^{(b)}, token_t^{(b)})$$

[Original seq2seq paper](#)  
[Tutorial in PyTorch](#)