W08D4

NLP II

Instructor: Eric Elmoznino

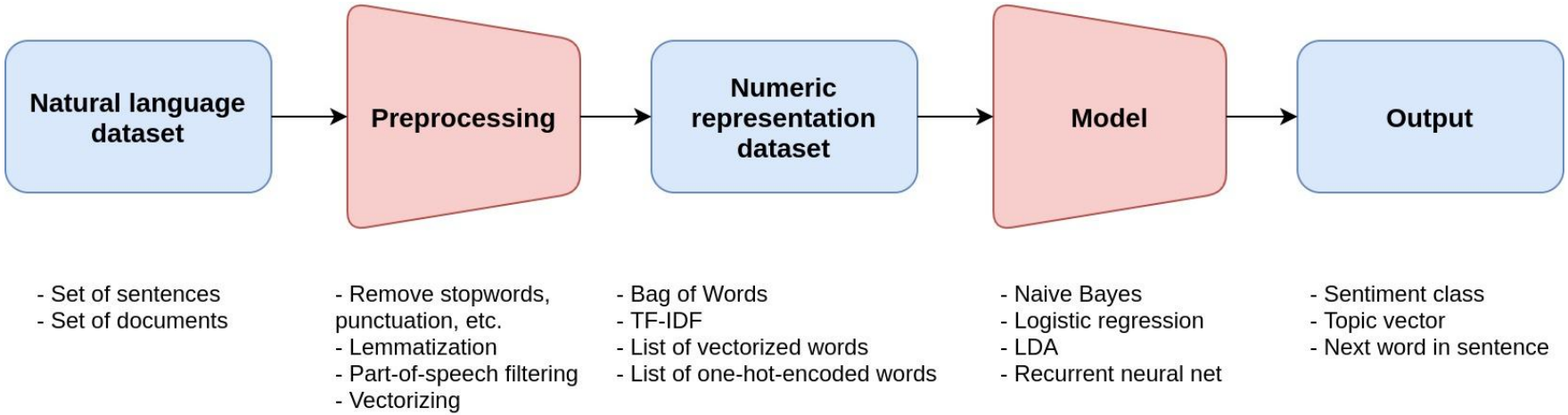
Outline for today

- 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.....”

Challenges?

large +/- word vocabulary

syntax

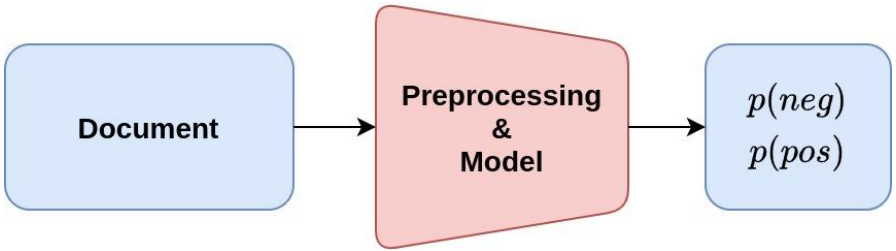
sarcasm

(long range) Negation

writing style

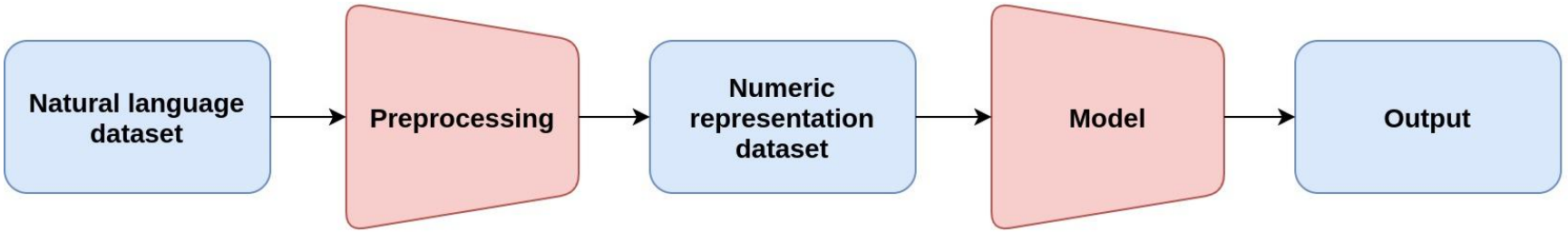
Technical description

- **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.



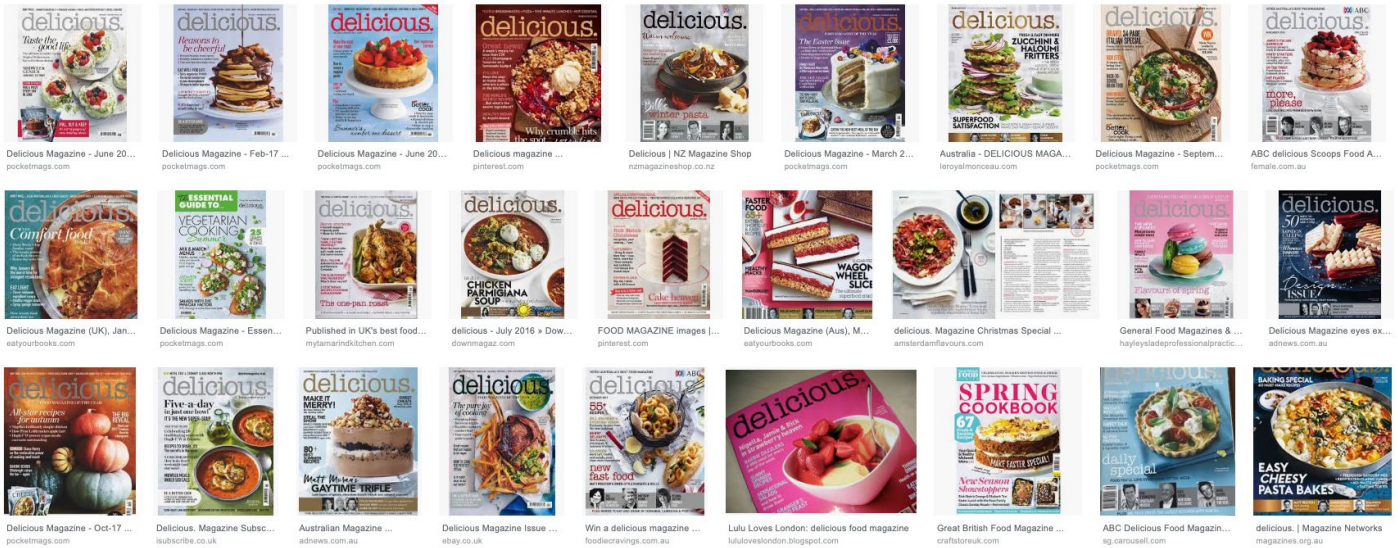
Demo

- Repository/sentiment_analysis_demo.ipynb



Topic modeling

Overview



Challenges?

multiple word meanings

dataset-dependent

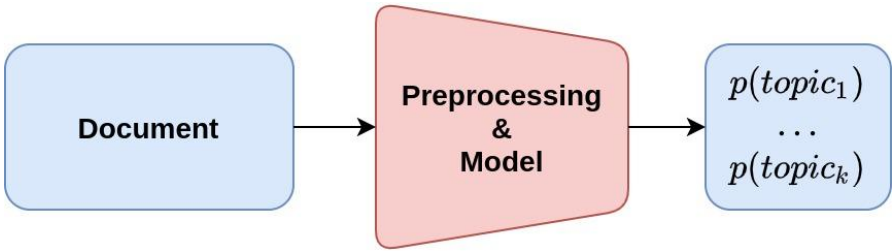
large documents

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)

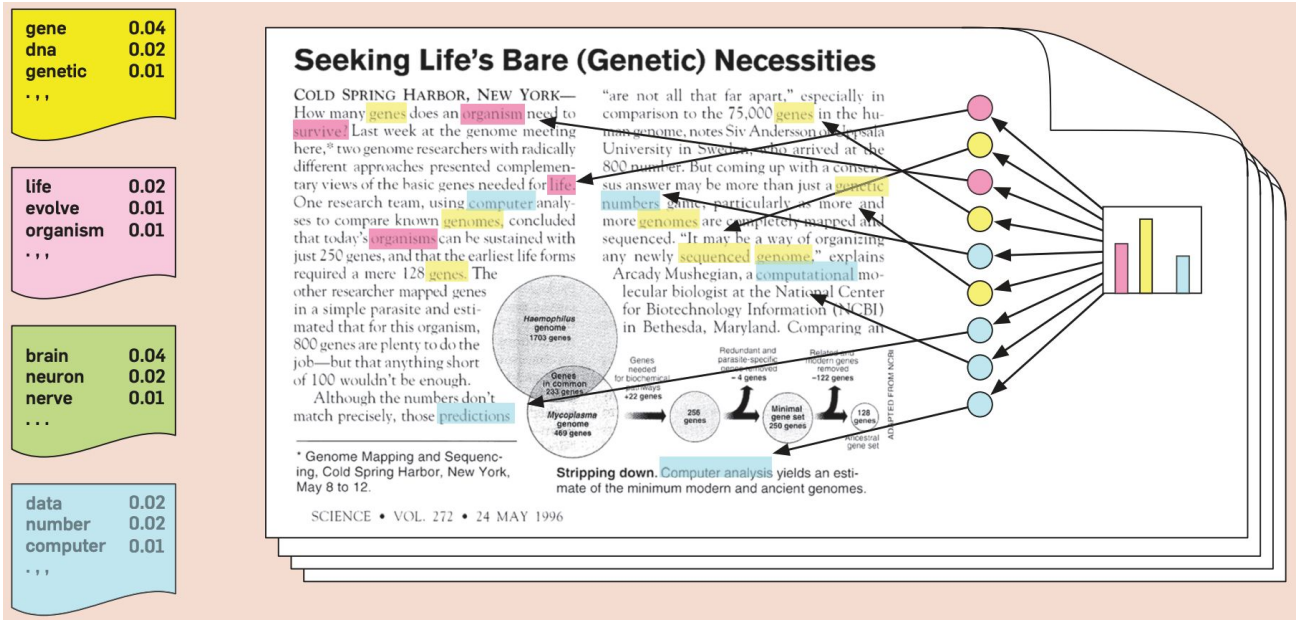
- Bayesian model: infer the hidden variables (topics) that generated the data
  - $p(topics \mid document) \propto p(document \mid topics)\, p(topics)$
- Developed by David Blei — one of the most cited papers in the last 15 years
- Insight:
  - Each document is a mixture of corpus-wide topics (i.e. a 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

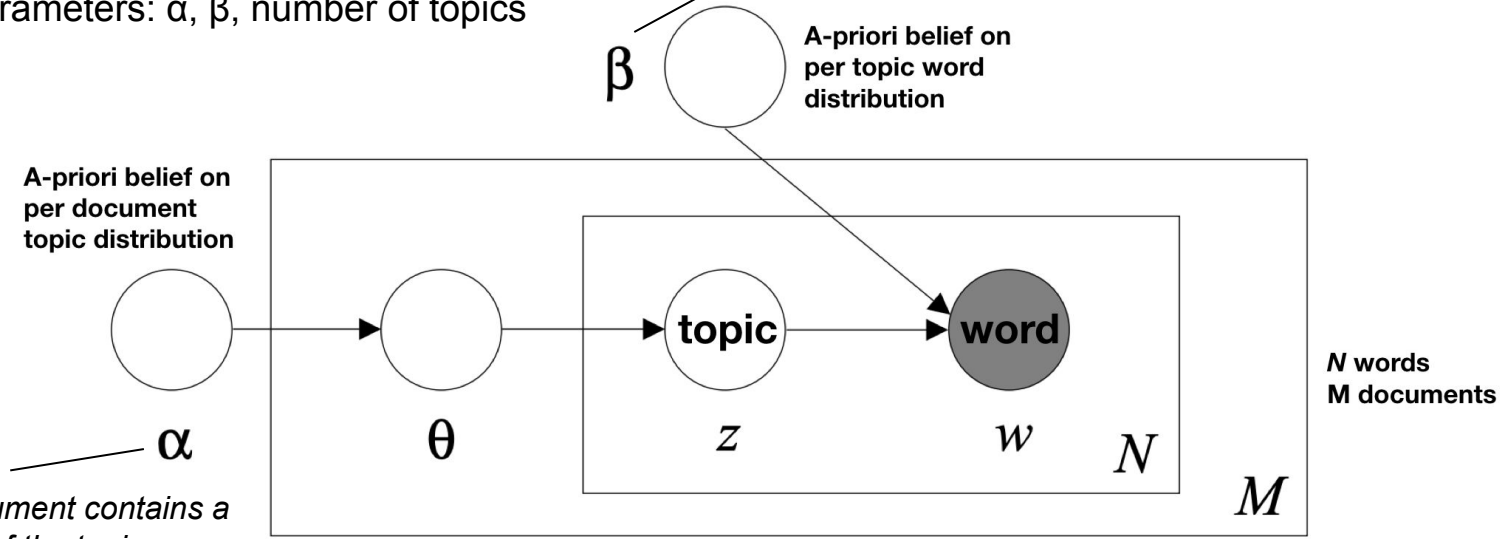
Each document is a mixture of topics



Latent Dirichlet Allocation (LDA)

- Observed features: words across all documents
- Hidden (latent) features: all other variables
- Hyperparameters: α, β, number of topics

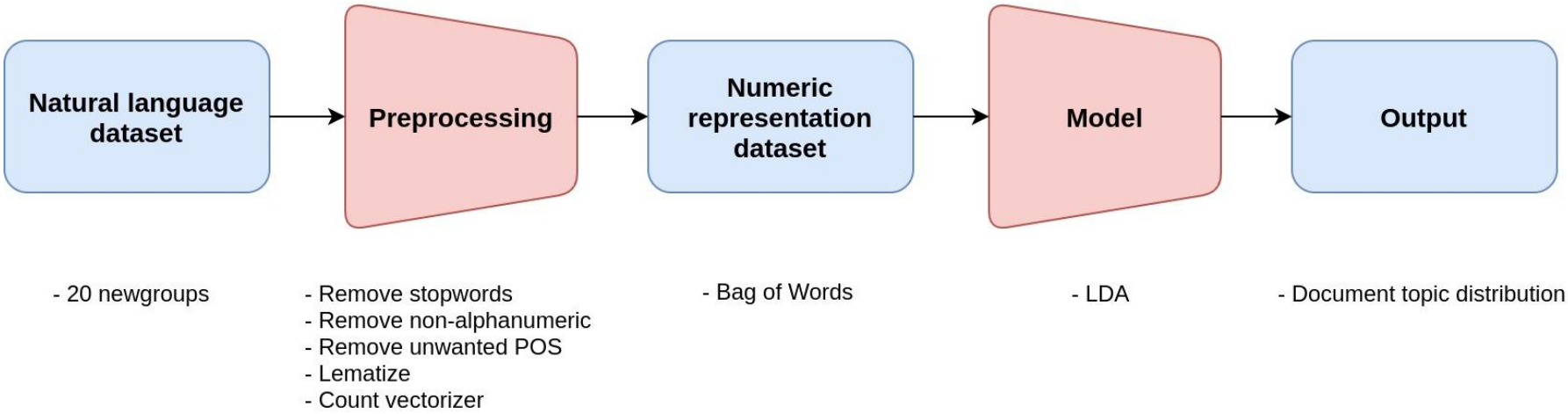
***High**: every topic contains a mixture of most of the words*
***Low**: every topic contains a mixture of only few words*



***High**: every document contains a mixture of most of the topics*
***Low**: every document is representative of only a few topic*

Demo

- Repository/topic_modeling_demo.ipynb



Language modeling

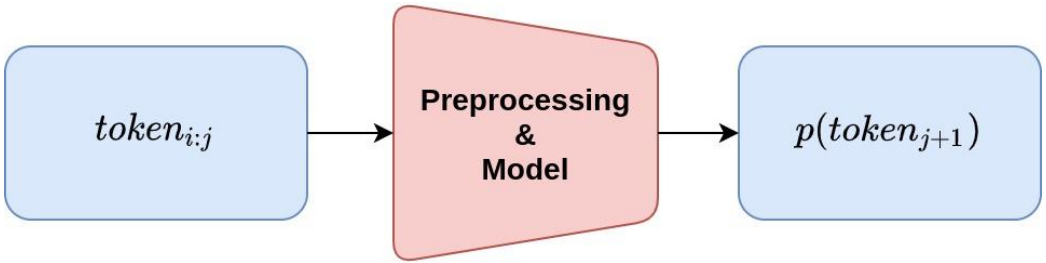
Overview

“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

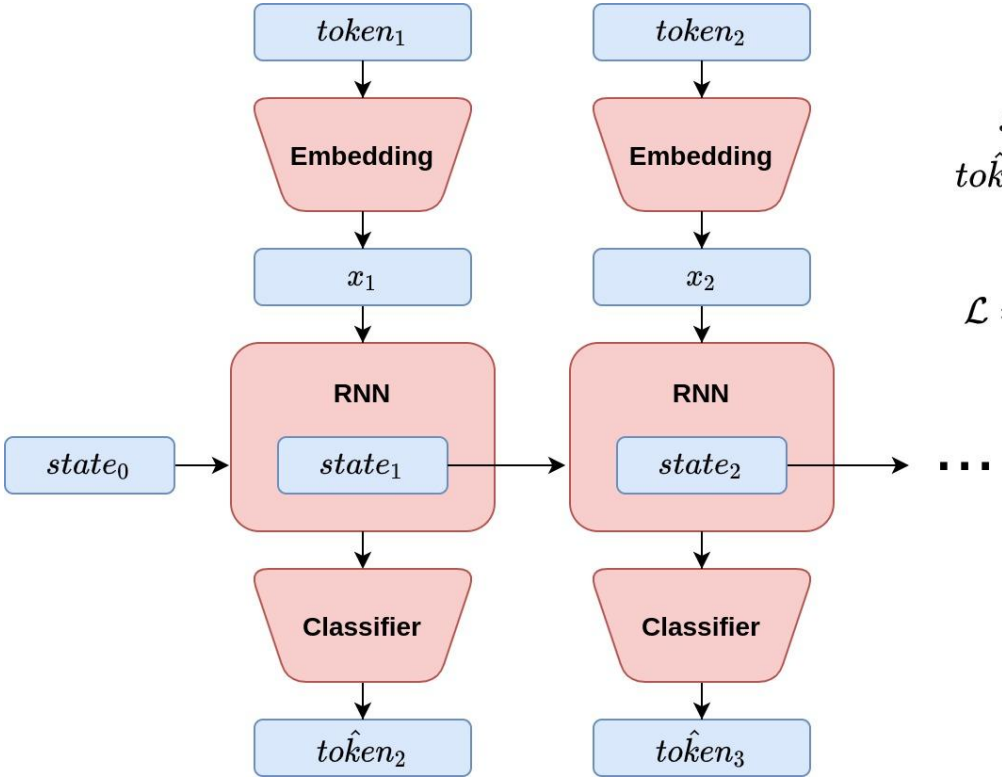


Example: recurrent neural network intuition

“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



$$x_t = Embedding(token_t)$$
$$state_t = RNN(x_t, state_{t-1})$$
$$\hat{token}_{t+1} = Classifier(state_t)$$

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

Translation

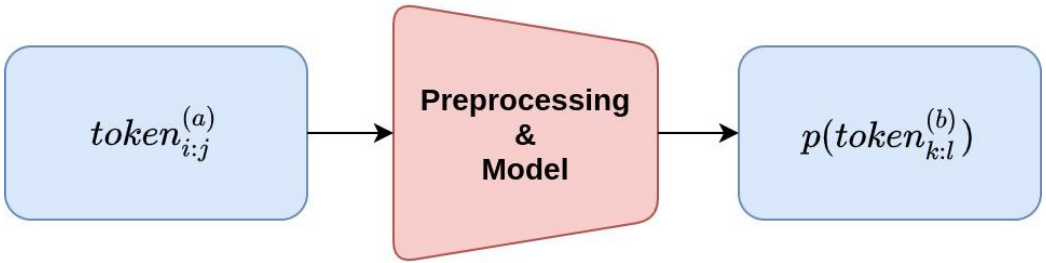
Overview

“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 langues *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

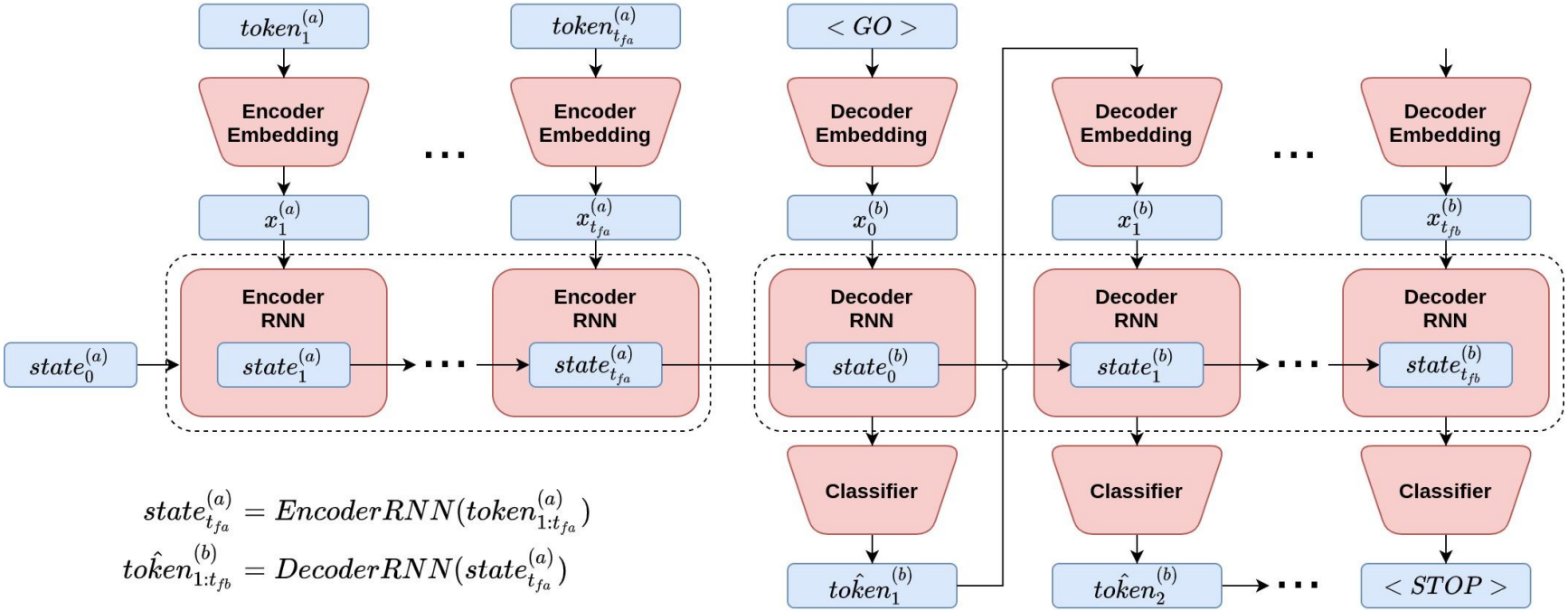


Example: recurrent neural network intuition

“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)



$$state_{t_{fa}}^{(a)} = EncoderRNN(token_{1:t_{fa}}^{(a)})$$

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

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

[Original seq2seq paper]
[Tutorial in PyTorch]