Natural Language Processing

Thymen Wabeke | CENTR R&D

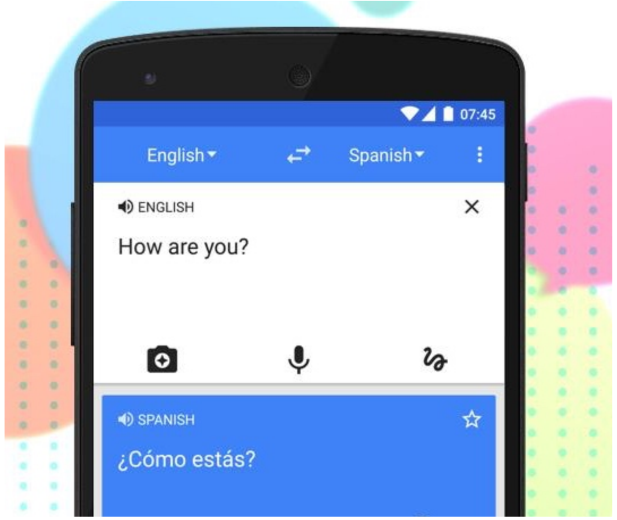
28/11/2019

“Natural language processing is the set of methods for making human language accessible to computers.”

Photo by Gabriel Sollmann on Unsplash

SIDN LABS

Examples of NLP applications



Machine translation

Text classification

Dialog systems

NLP tasks

- Tokenization
- Part-of-speech tagging
- Sentiment analysis
- Named-entity recognition
- Topic modelling
- Machine translation
- Text classification
- …

Why relevant for TLDs?

- Detect (active) use of domains
- Classify content of webpages
- Domain name recommendation
- Clean WHOIS database
- …

- What's your interest?
- What's your experience?

Today's topics

Tools:

- Jupyter Notebooks
- Scikit-learn
- NLTK

Concepts:

- Bag of Words
- Cross validation
- Model selection

Techniques:

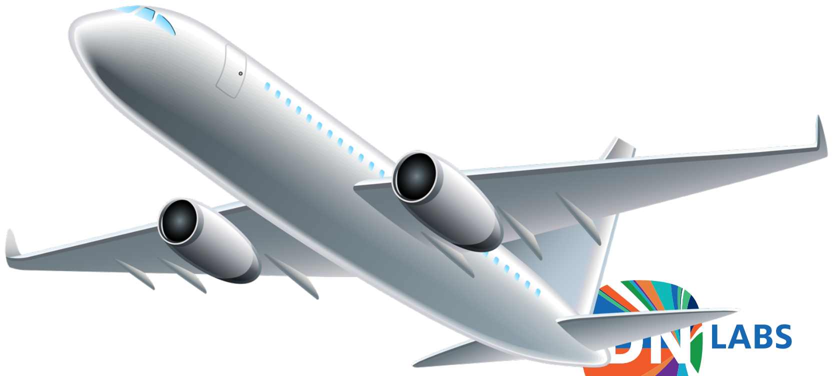
- Normalization
- Stemming/lemmatization
- TF-IDF
- Latent Dirichlet Allocation
- Random forest
- Support vector machine

Agenda

- Task 0: Text encoding (9:15-9:45)
- Task 1: Topic extraction (9:45-10:30)

- Coffee break? - -

- Task 2: Text classification (10:45-11:30)



“Natural language processing is the set of methods for making human language accessible to computers.”

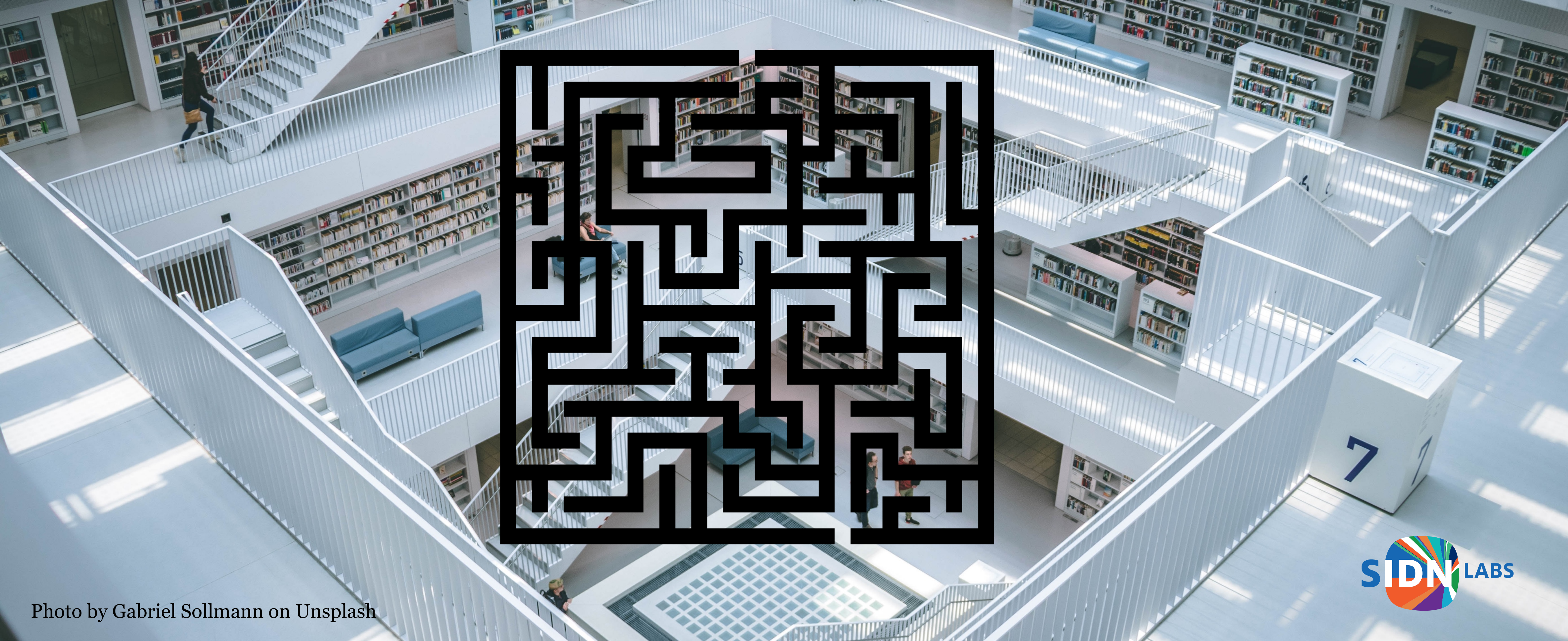


Photo by Gabriel Sollmann on Unsplash

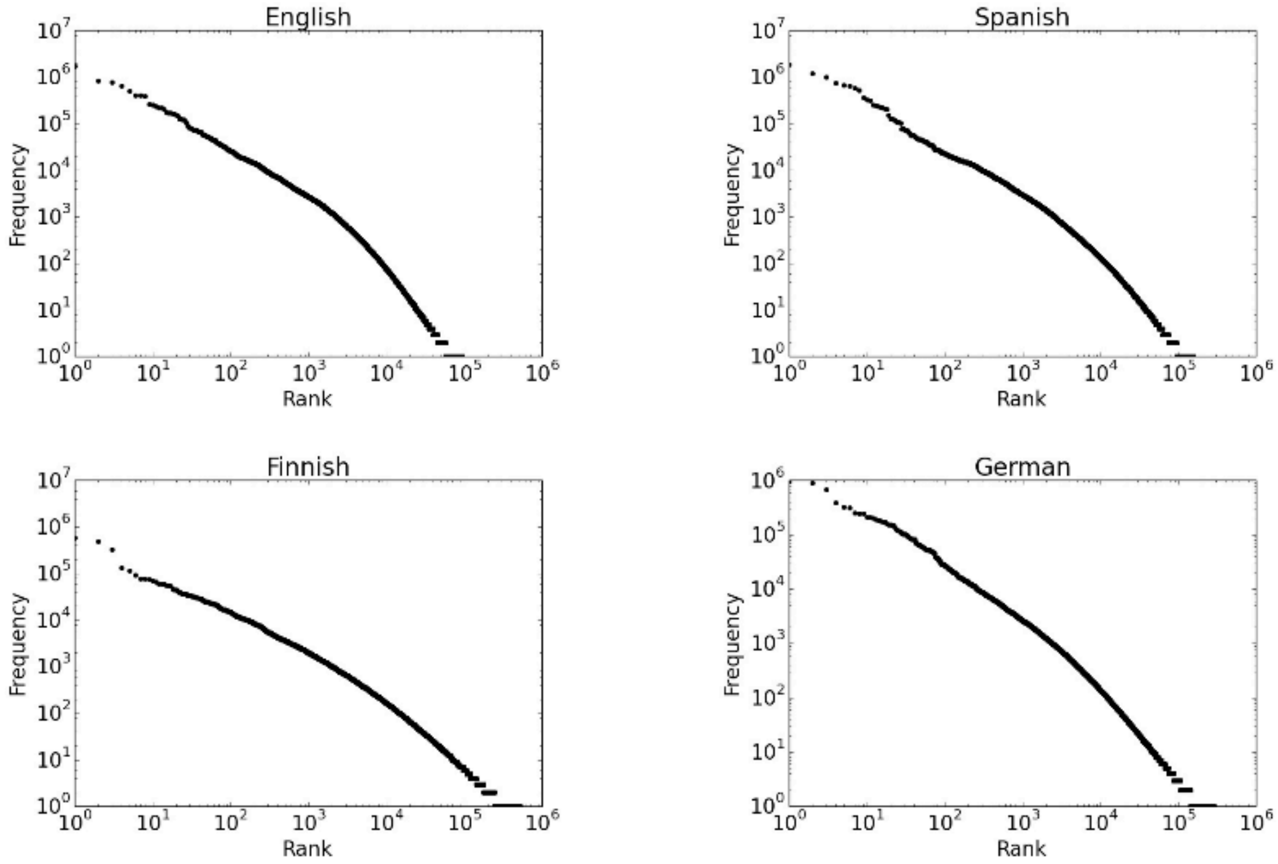
Why is NLP hard? Language is ambiguous

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Quantifier scope: Every child loves some movie
- Reference: John dropped the goblet onto the glass table and it broke.
- Discourse: The meeting is cancelled. Nicholas isn’t coming to the office today.

https://www.inf.ed.ac.uk/teaching/courses/fnlp/lectures/01_slides.pdf

Why is NLP hard?

Sparse data due to Zipf's Law:

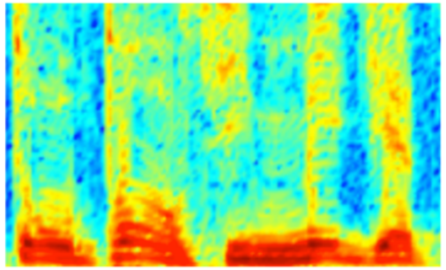


https://www.inf.ed.ac.uk/teaching/courses/fnlp/lectures/01_slides.pdf

Why is NLP hard?

- Algorithms cannot work with raw text directly;
- Text must be converted into numbers.

AUDIO



Audio Spectrogram

DENSE

IMAGES

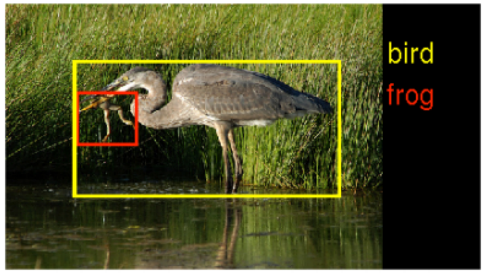


Image pixels

DENSE

TEXT

0	0	0	0.2	0	0.7	0	0	0	...	...

Word, context, or document vectors

SPARSE

https://www.tensorflow.org/tutorials/text/word_embeddings

Text encoding: Bag-of-Words model

Bag-of-Words (BoW)

Two steps:

- Construct vocabulary of known words (the items).
- Measure of the presence of known words documents (what’s in the bag).

	is	this	a	an	question	answer
Is this a question?	1	1	1	0	1	0
This is an answer.	1	1	0	1	0	1

https://machinelearningmastery.com/gentle-introduction-bag-words-model/

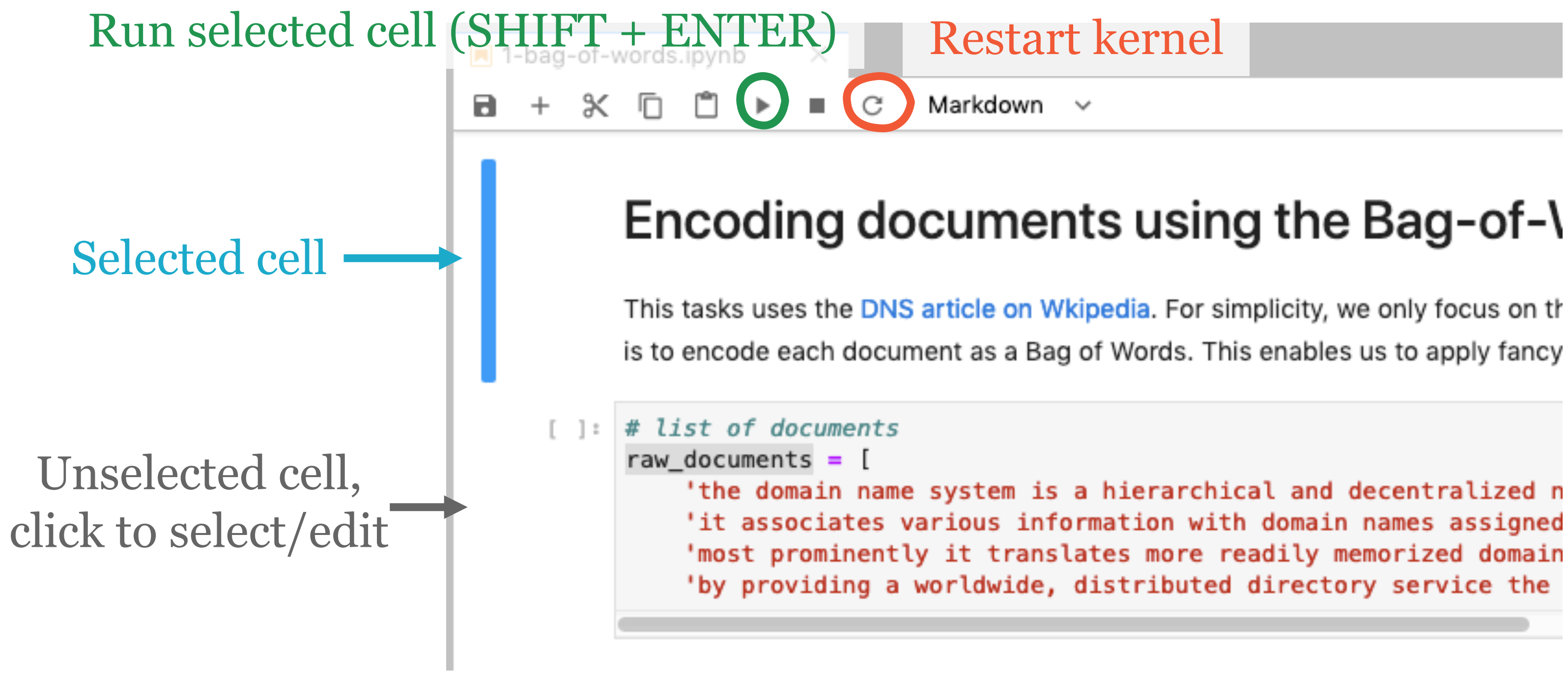
Warm up task: create Bag-of-Words (BoW)

- We will use Jupyter Notebooks
- Collaboration is encouraged, ensure experience is balanced

- Open [http://nlp-workshop.sidnlabs.nl/notebook-*[your-number*]](http://nlp-workshop.sidnlabs.nl/notebook-[your-number])
- Enter the password `centrpasswordverysafestring`

Warm up task: create Bag-of-Words (BoW)

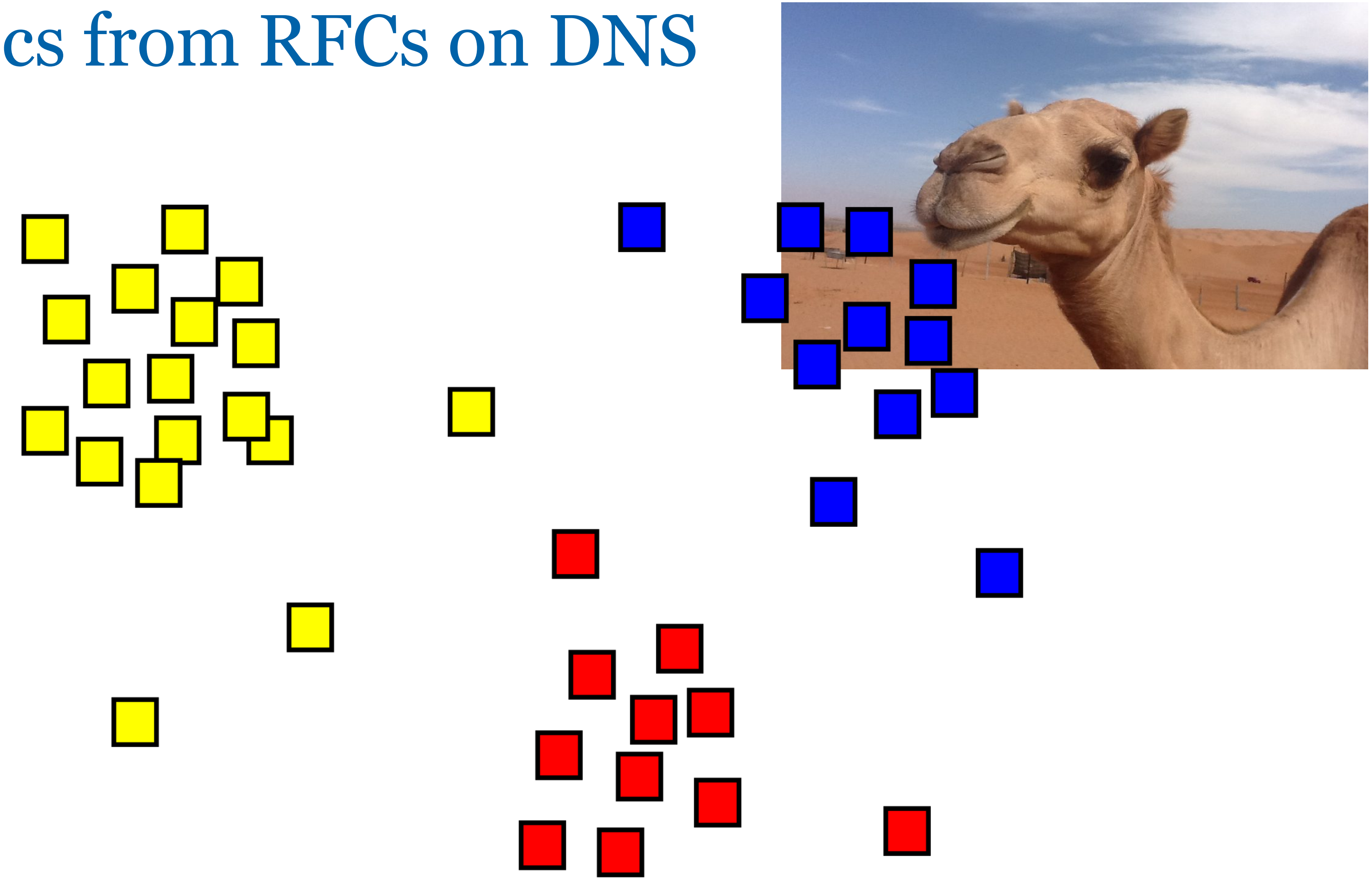
- Open `task-0/1-bag-of-words.ipynb`
- Run the cells. Try to understand what happens.



Bag-of-Word challenges

- Word order ignored (unless large n-grams)
- Sparse vectors (choose right algorithm)
- Noisy data
  - Pre-processing
  - Remove stop words
  - Use significant words only
- No awareness of word semantics
  - Not necessarily a problem
  - Use word embeddings

Extract topics from RFCs on DNS



Task 1: Latent Dirichlet Allocation (LDA)

Goal:

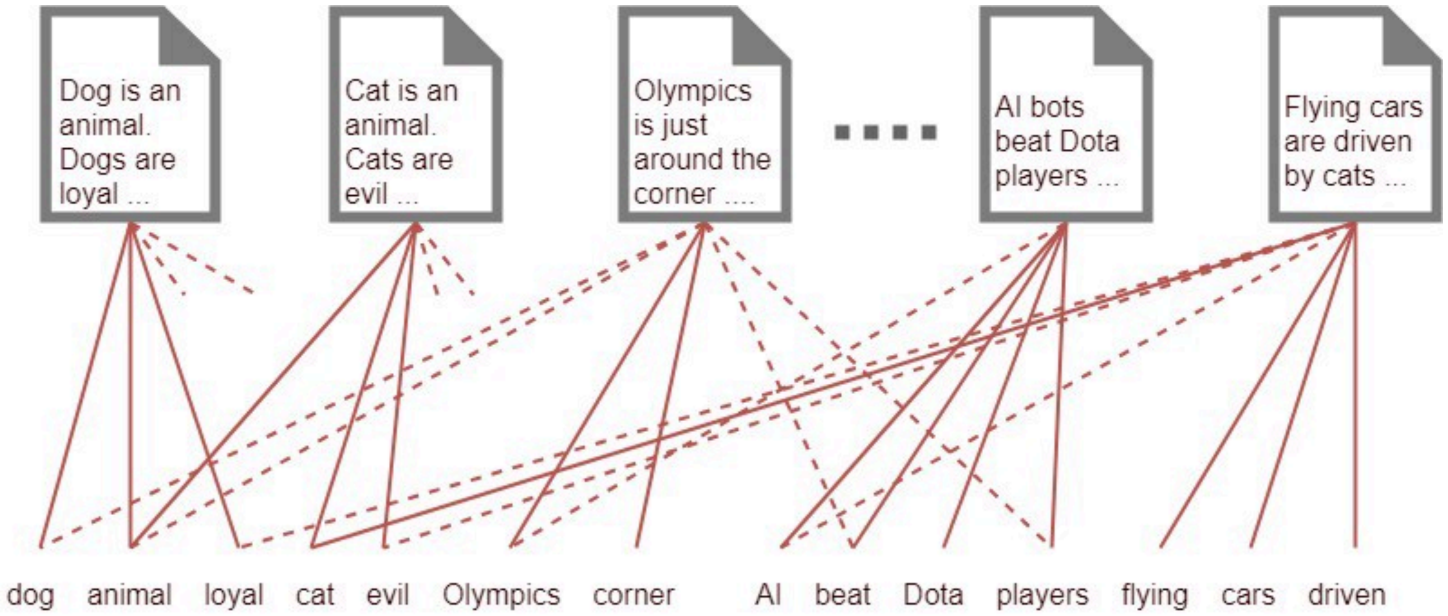
- Identify topics within documents (unsupervised)

Assumptions:

- Each document can be described by a distribution of topics, and each topic can be described by a distribution of words.
- Word and document order is not important.
- The number of topics known in advance.
- The same word can belong to multiple topics.

https://dzone.com/articles/lda-for-text-summarization-and-topic-detection

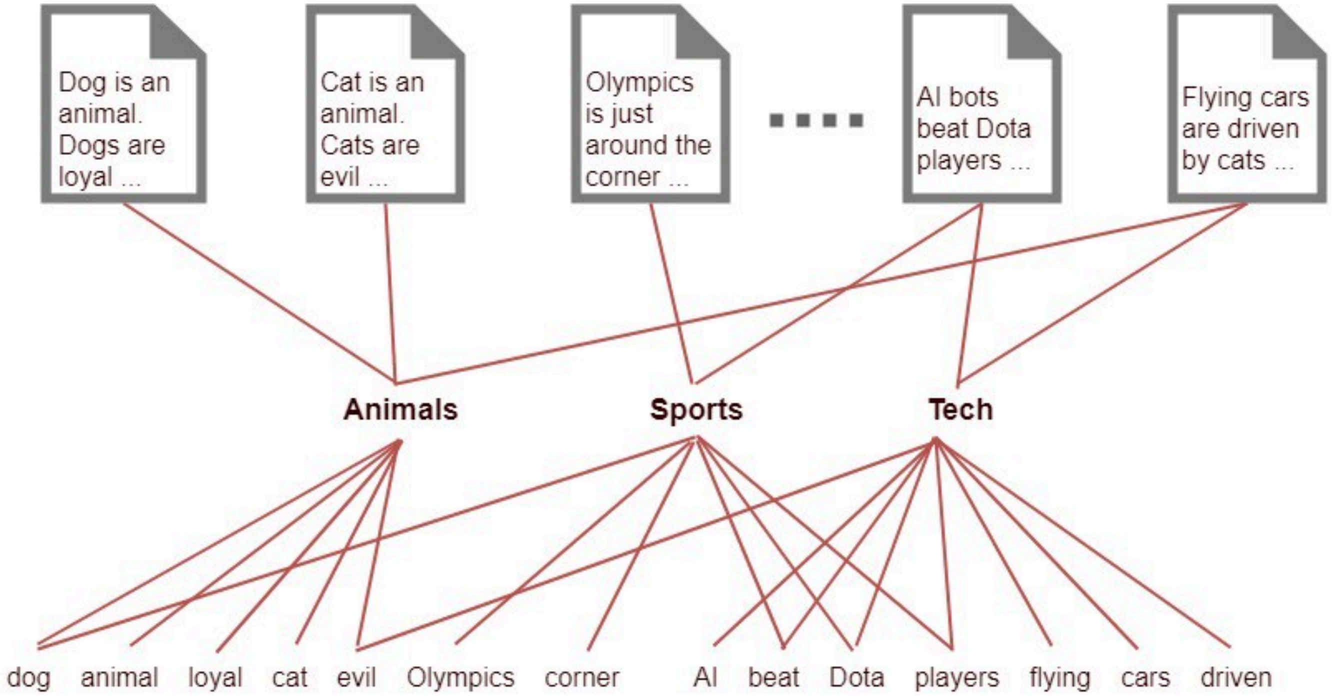
Brute force topic detection



https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation

Optimized topic detection with LDA

Latent topics



https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation

Task 1: topic modeling

- Open `task-1/1-topic-modelling.ipynb`
- Write pre-processing methods
- Fit LDA model
- Explore extracted topics

Data ingest → Pre-processing → Feature extraction (BoW) → Topic extraction (LDA) → Evaluation

Pre-processing methods

```
def remove_punctuation(document):
        filters = '!"\'#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n'
        translate_dict = dict((c, ' ') for c in filters)
        translate_map = str.maketrans(translate_dict)
        return document.translate(translate_map)


def remove_numbers_only(document):
        return re.sub(r'\b[0-9]+\b\s*', '', document)


def remove_short_words(document):
        return ' '.join(token for token in document.split() if len(token) > 1)
```

Topic modelling results

- Can you label the topics? Do they make sense?
- What are the best parameters?

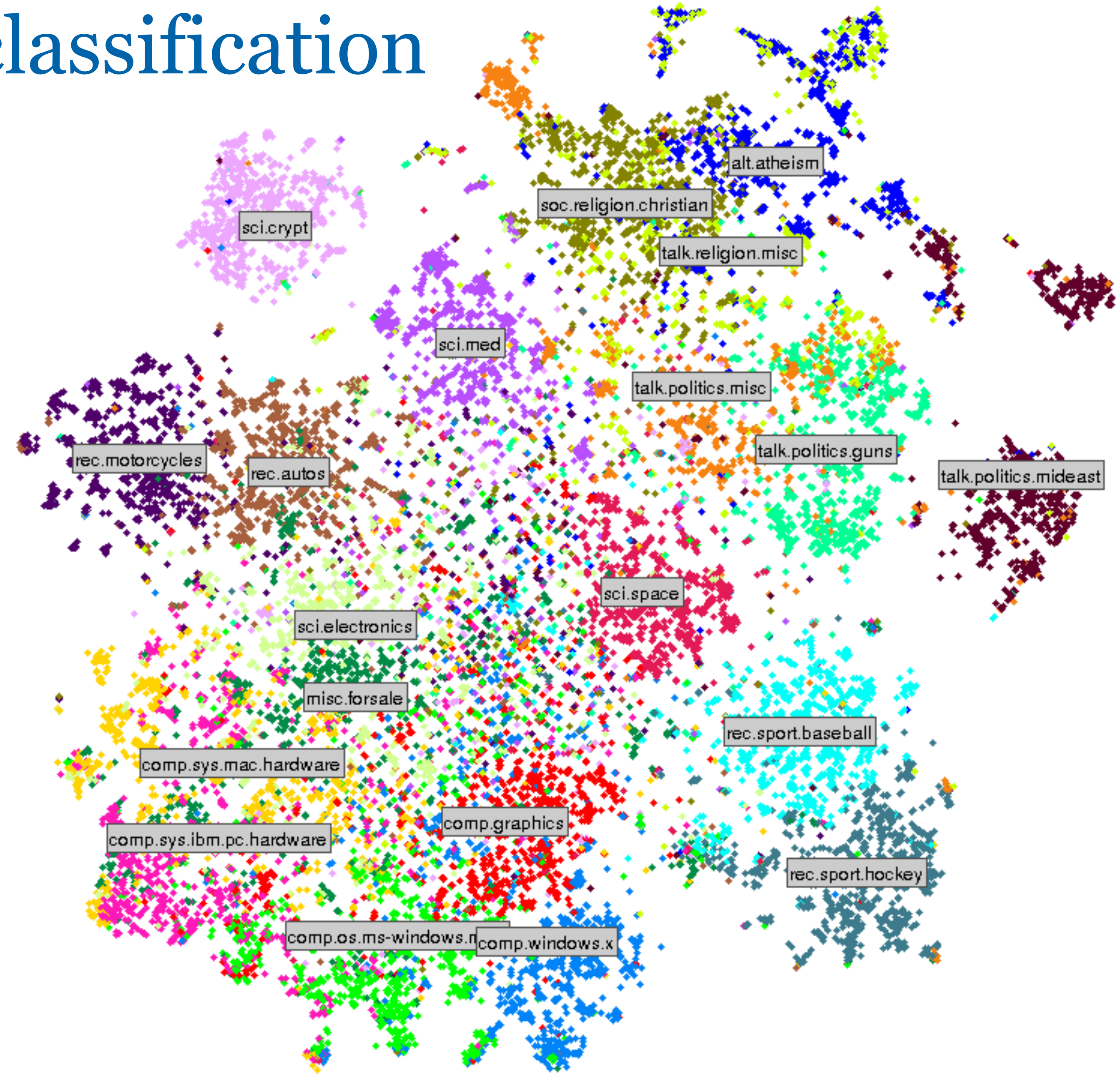
- What's the influence of pre-processing steps?
- Did you lemmatize and stem words?

- Did anyone implement "trending topics"?

Coffee break?

Please shutdown kernels to release memory.

Document classification



Text classification

Goal: assigns a label to unseen document based on its content

- Input (x): text of document
- Target (y): label of document

Supervised method: ground truth is required for training!

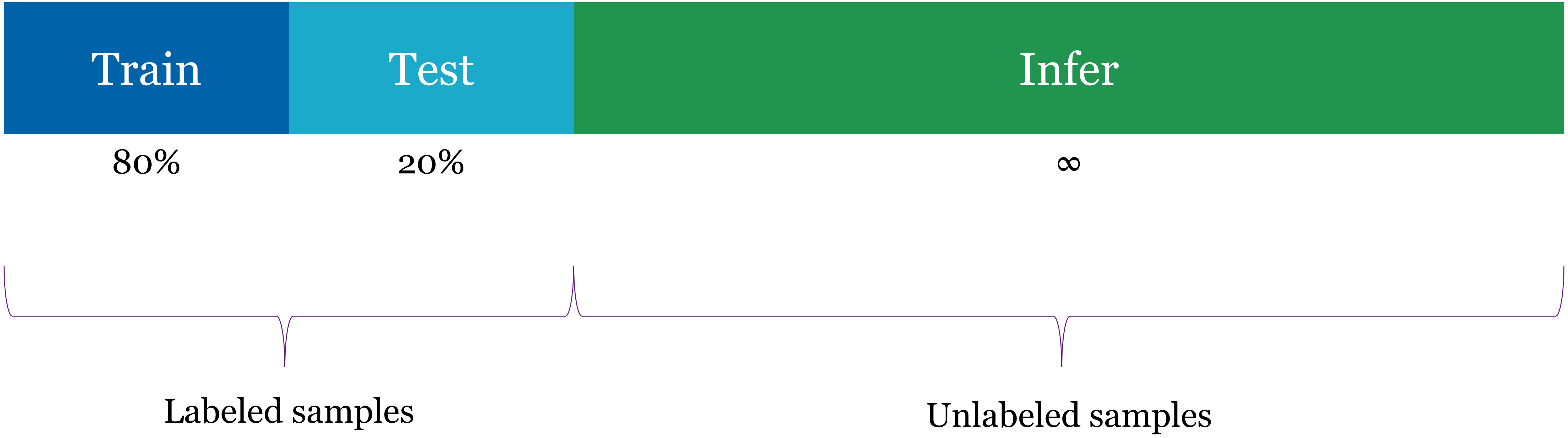
Example:

- Classify page type (parking page, eCommerce website, etc.)

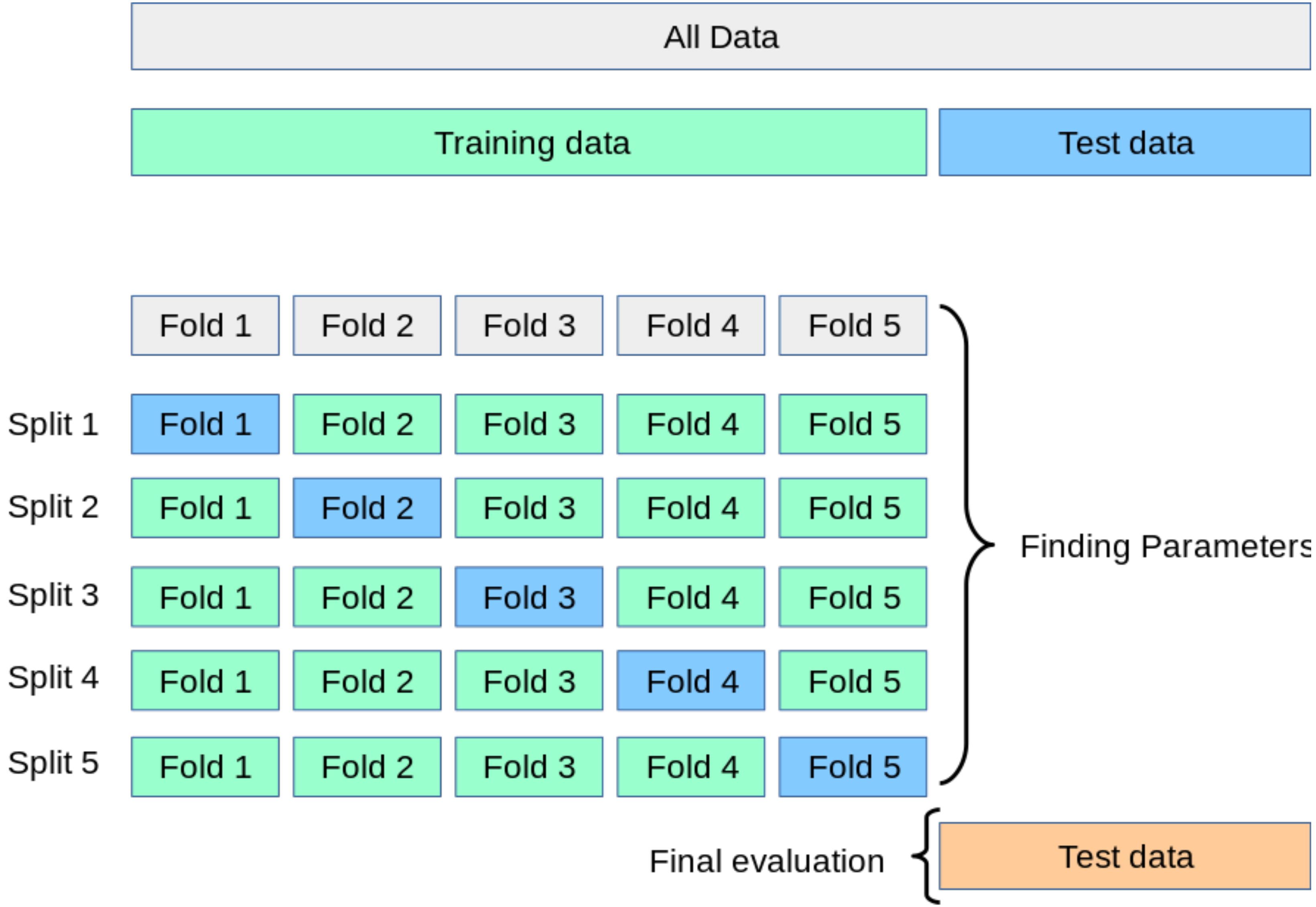
How do machines learn from data?

Train → **Test** → **Infer**

Data is used for one purpose only



Re-use training samples with cross validation



Task 2: text classification

- 3k newsgroup articles from 5 categories
- Focus on conducting experiments and model validation

- Explore 2 document encodings:
  - Term Frequency (TF)
  - TF-IDF
- Explore 2 classification methods:
  - Random Forest
  - Support Vector Machine

TF-IDF: find relevant terms

Intuition:

- A words that occurs multiple times in a single document is more relevant (TF)
- A words that occurs in many documents is less relevant (IDF)

Example:

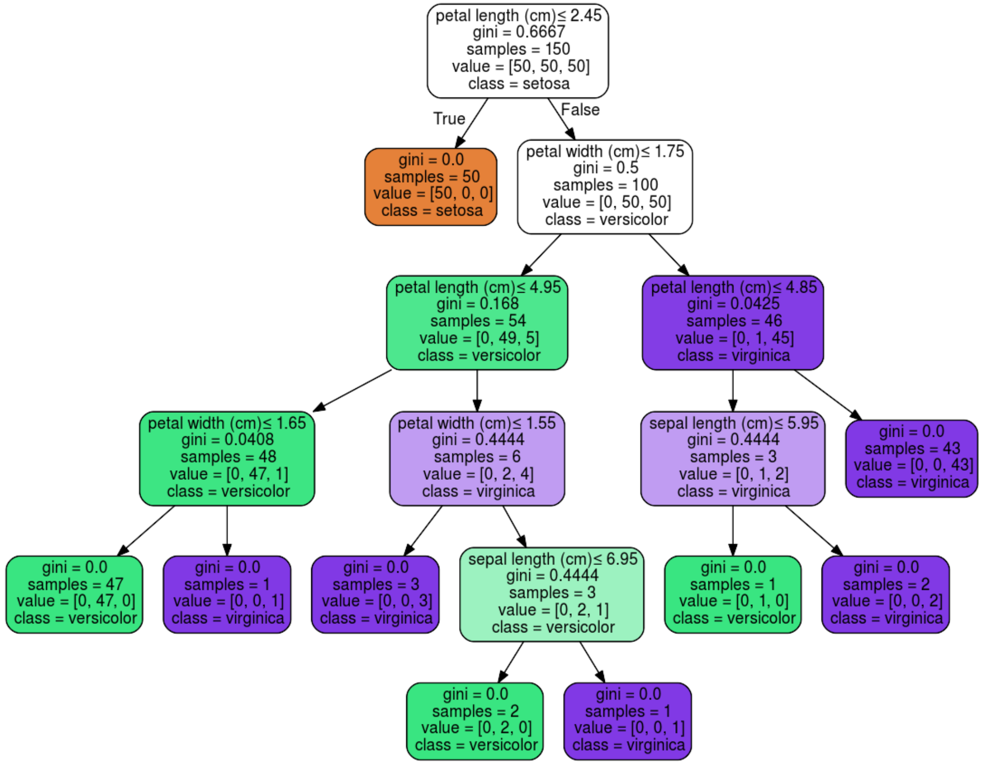
A. The *car* is driven on the *road*.

B. The *truck* is driven on the *highway*.

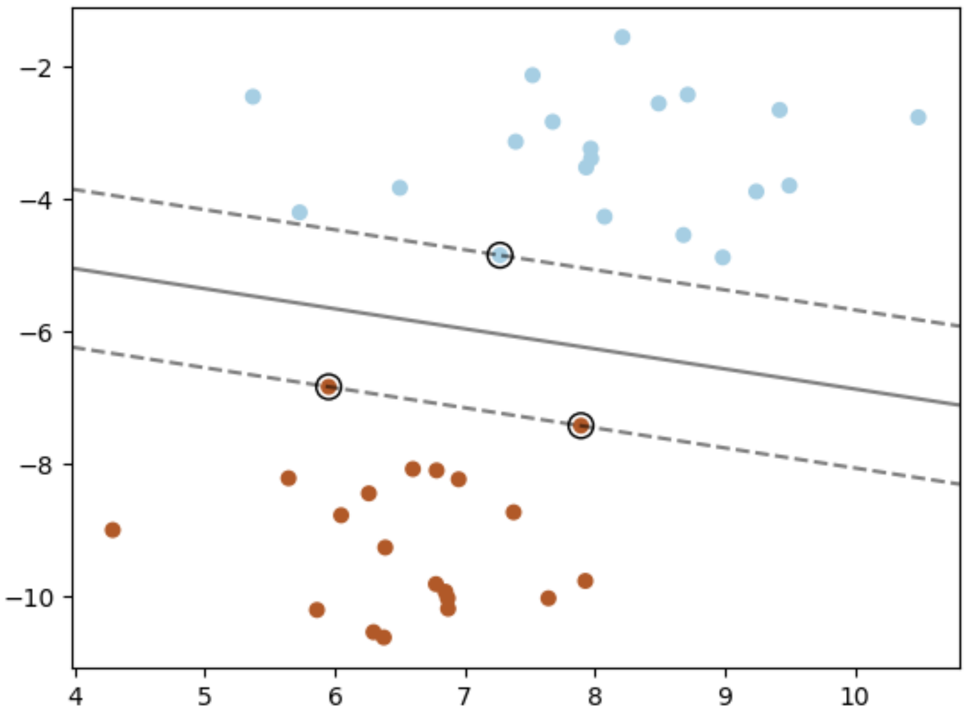
Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

SIDN LABS

Example taken from medium.freecodecamp.org (Tripathi, 2018)



Random Forest



Support Vector Machine

S DN LABS

Task 2: text classification

- Open `task-2/1-text-classification.ipynb`
- Write 4 pipelines
  - TF & RF
  - TF & SGD
  - TF-IDF & RF
  - TF-IDF & SGD
- Train and evaluate on training data (wrong approach!)
- Train and evaluate using cross-validation

Task 2: text classification

- Do test results differ with cross-validation?
- Which pipeline performs the best? Using what metric?

- Optimal parameters?

Zijn er nog vragen?

Volg ons

.nl SIDN.nl

@SIDN

in SIDN

Dankjewel voor je aandacht!

SIDN LABS