Strive

SCHOOL

/ Text Data

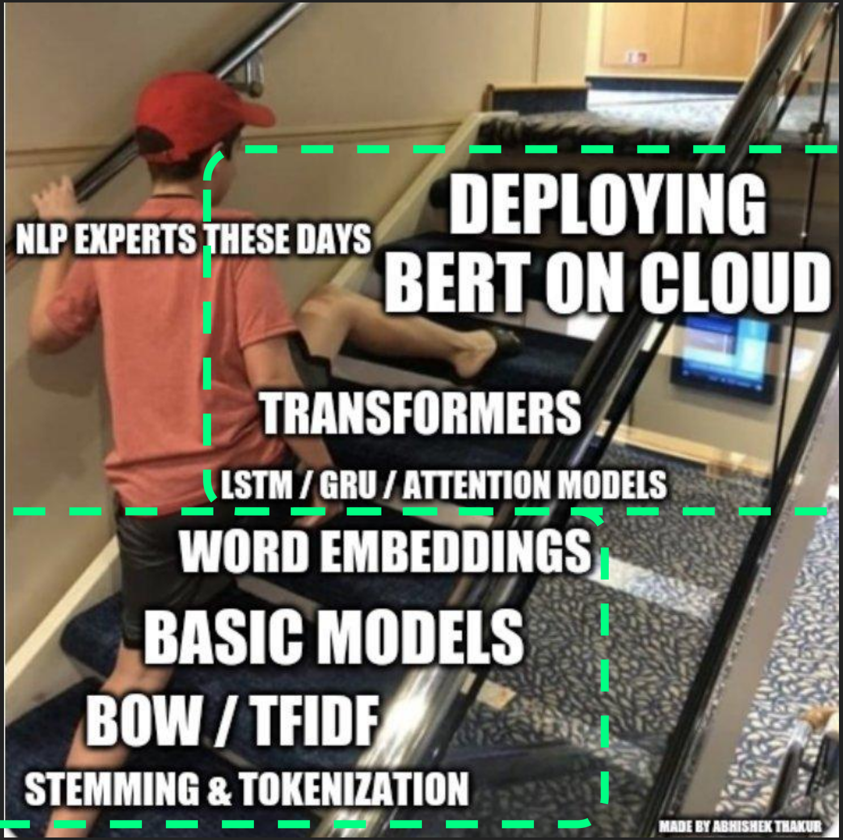
/ Free text features

Some tabular datasets contains free text variables. In this lesson we are going to see how to process it.

Remember: If you data consists on solely text data you should consider stronger methods like RNNs or Transformers.

Many options for NLP

Many options for NLP



NLP Deep Learning Session

Today

Free Text Features

/ Common text features

- Name and surname (like Titanic dataset)
- House Direction (Geoposition encoding)
- Item (ad, car, house, 2nd hand product)
  - Title
  - Subtitle
  - Description

name
Braund, Mr. Owen Harris
Cumings, Mrs. John Bradley (Florence Briggs Thayer)
Heikkinen, Miss. Laina
Futrelle, Mrs. Jacques Heath (Lily May Peel)
Allen, Mr. William Henry
Moran, Mr. James
McCarthy, Mr. Timothy J
Palsson, Master. Gosta Leonard
Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
Nasser, Mrs. Nicholas (Adele Achem)
Sandstrom, Miss. Marguerite Rut
Bonnell, Miss. Elizabeth

Part 1:

BoW and TF-IDF

Bag of Words (BoW)

/ One simple method is Bag of Words. You can consider this method as a N-Hot Encoding where we put a 1 if the word appears and 0 otherwise.

The dog is on the table

/ In sklearn there is a similar method called Count Vectorizer where we count the number of occurrences of a word.

```
from sklearn.feature_extraction.text import CountVectorizer
```

CountVectorizer

/ With CountVectorizer where we count the number of occurrences of a word.

(excited) Hi everyone!	I’m so excited about this course!	So excited. SO EXCITED. EXCITED, I AM!

hi	every one	I’m	so	excited	about	this	course
1	1			1			
		1	1	1	1	1	1
		1	2	3			

TF-IDF

Term Frequency (TF)

```
tf = 1 / x.sum(axis=1) [:,None]
x = x * tf
```

Inverse Document Frequency (IDF)

```
idf = np.log(x.shape[0] / (x > 0).sum(0)) x = x * idf
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

Term Frequency (TF)

/ With `TfidfVectorizer(use_idf = False)` we count the percentage of occurrences of a word.in a sentence.

(excited) Hi everyone!	I’m so excited about this course!	So excited. SO EXCITED. EXCITED, I AM!

⬇

hi	every one	I’m	so	excited	about	this	course
0.33	0.33			0.33			
		0.16	0.16	0.16	0.16	0.16	0.16
		0.16	0.33	0.5			

Summation of each row is equal to 1

Term Frequency - Inverse Document Frequency

/ With `TfidfVectorizer(use_idf = True)` we count the percentage of occurrences of a word in a sentence versus the other rows.

Useful for boost most unique words and ignore most frequent word (like "excited")

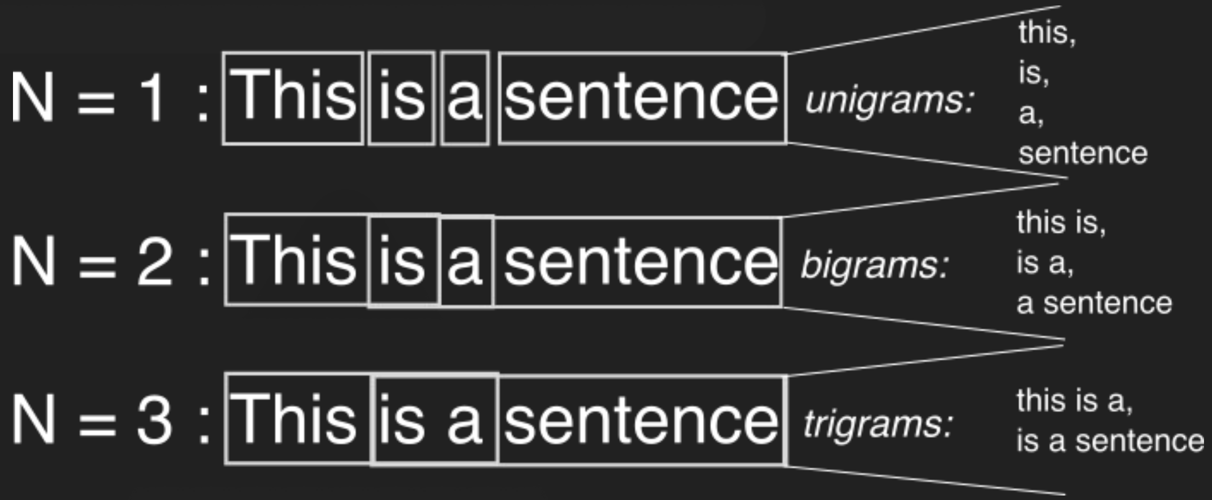
(excited) Hi everyone!	I'm so excited about this course!	So excited. SO EXCITED. EXCITED, I AM!

hi	every one	I'm	so	excited	about	this	course
0.36	0.36			0			
		0.06	0.06	0	0.18	0.18	0.18
		0.06	0.13	0			

Technically we postprocess TF matrix by normalizing data column-wise

Parameter: N-Grams

/ Helps to to use local context. Specially words starting with “no” or “not”.



ngram_range : *tuple (min_n, max_n), default=(1, 1)*

The lower and upper boundary of the range of n-values for different n-grams to be extracted. All values of n such that min_n <= n <= max_n will be used. For example an `ngram_range` of `(1, 1)` means only unigrams, `(1, 2)` means unigrams and bigrams, and `(2, 2)` means only bigrams. Only applies if `analyzer is not callable`.

Parameter: LowerCase

Very, very sunny.
Sunny... Sunny!

lowercase : ***bool, default=True***

Convert all characters to lowercase before tokenizing.

Very	very	Sunny	sunny
1	1	0	1
0	0	2	0

We don't want this. So we need to lowercase.

Stemming and Lemmatization

I had a car

We have cars

↓

I have car

We have car

Stemming

democracy, democratic, and democratization → democr

see, saw → s

Lemmatization

democracy, democratic, and democratization → democracy

see, saw → see or saw (depending on context)

Parameters: Min and Max Frequency

max_df : ***float or int, default=1.0***

When building the vocabulary ignore terms that have a document frequency strictly higher than the given threshold (corpus-specific stop words). If float in range [0.0, 1.0], the parameter represents a proportion of documents, integer absolute counts. This parameter is ignored if vocabulary is not None.

Useful for ignore common words: "a", "and", "the", ...

Also known as Stopwords

min_df : ***float or int, default=1***

When building the vocabulary ignore terms that have a document frequency strictly lower than the given threshold. This value is also called cut-off in the literature. If float in range of [0.0, 1.0], the parameter represents a proportion of documents, integer absolute counts. This parameter is ignored if vocabulary is not None.

Useful for ignore too specific or rare words

Conclusion BoW & TF-iDF

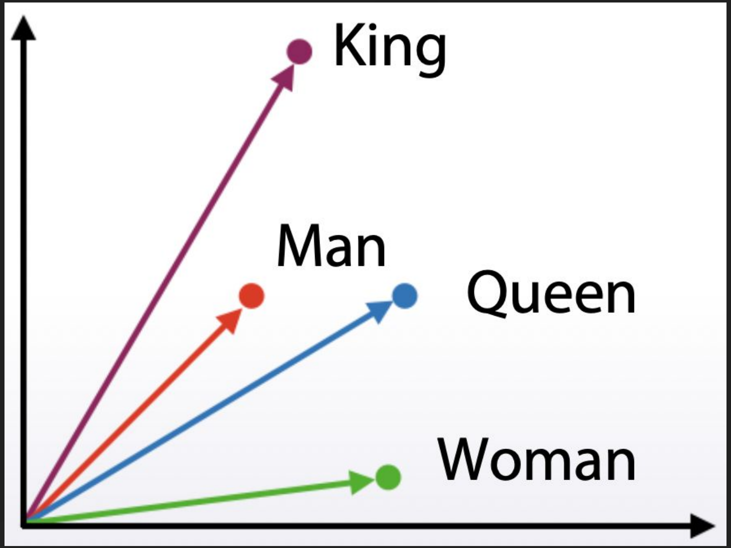
- Lowercase with: `lowercase`
- stemming/lemmatization
- Remove common words and stopwords: `min_df`
- Remove too specific and rare words: `max_df`
- Ngrams can help to use local context: `ngram_range`
- TFiDF
    - TF helps to normalize in frequencies
    - IDF helps to ignore most frequent word.

Part 2:

Vector representations

Vector representations

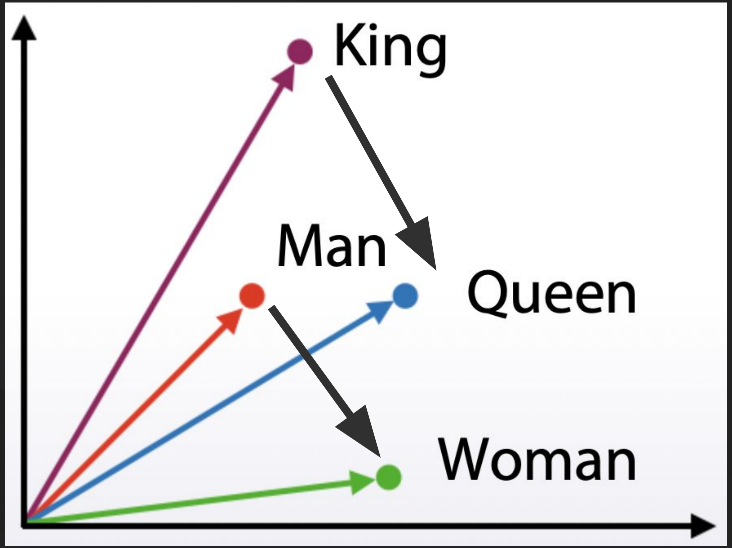
Words (and also sentences) can be represented with vectors also known as embeddings or word2vec.



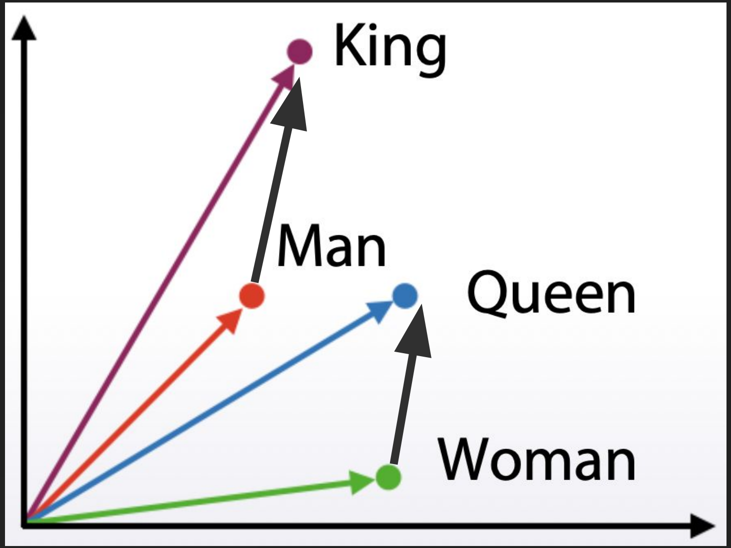
Vector representations

Vectors have semantic meaning encoded. King + woman - man ≈ Queen

Male-Female dimension



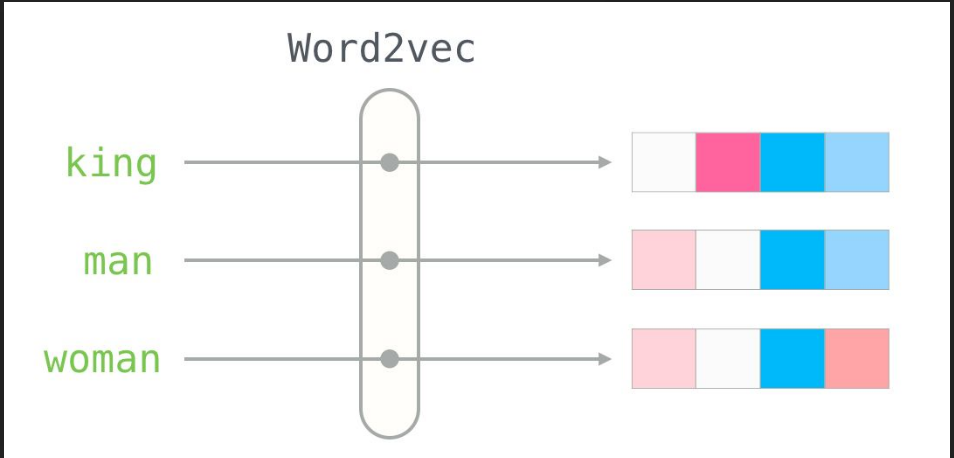
Man-King dimension



Pretrained models

Words

- Word2vec
- Glove
- FastText
- Etc.



Sentences

- Doc2vec
- Manually: Taking the mean() or sum() of all word vectors.

Word vectors VS BoW/TF-IDF

BoW/TF-IDF

- Very large vectors (size of vocabulary dictionary).
    - Super very large vectors if we do NGrams.
- Meaning of each value in vector is known

Word vectors

- Relatively small vectors (sizes typically from 64 to 1024).
- Values in vector can be interpreted only in some cases.
- The words with similar meaning often have similar vectors.

Very different but your solution can have both methods!

/ Q&A

What are your doubts?

Strive
SCHOOL