AMERICAN UNIVERSITY OF BEIRUT

ARTIFICIAL INTELLIGENCE,
DATA SCIENCE,
& COMPUTING HUB

AIcademy

Text Mining
Natural Language Processing

Processing Text Data

Allows us to build systems like ChatGPT

which was trained on **560 GB** of data

processing **300 billion words**

The data we will use

Arabic news dataset of about **60 MB**

Containing 45500 news articles

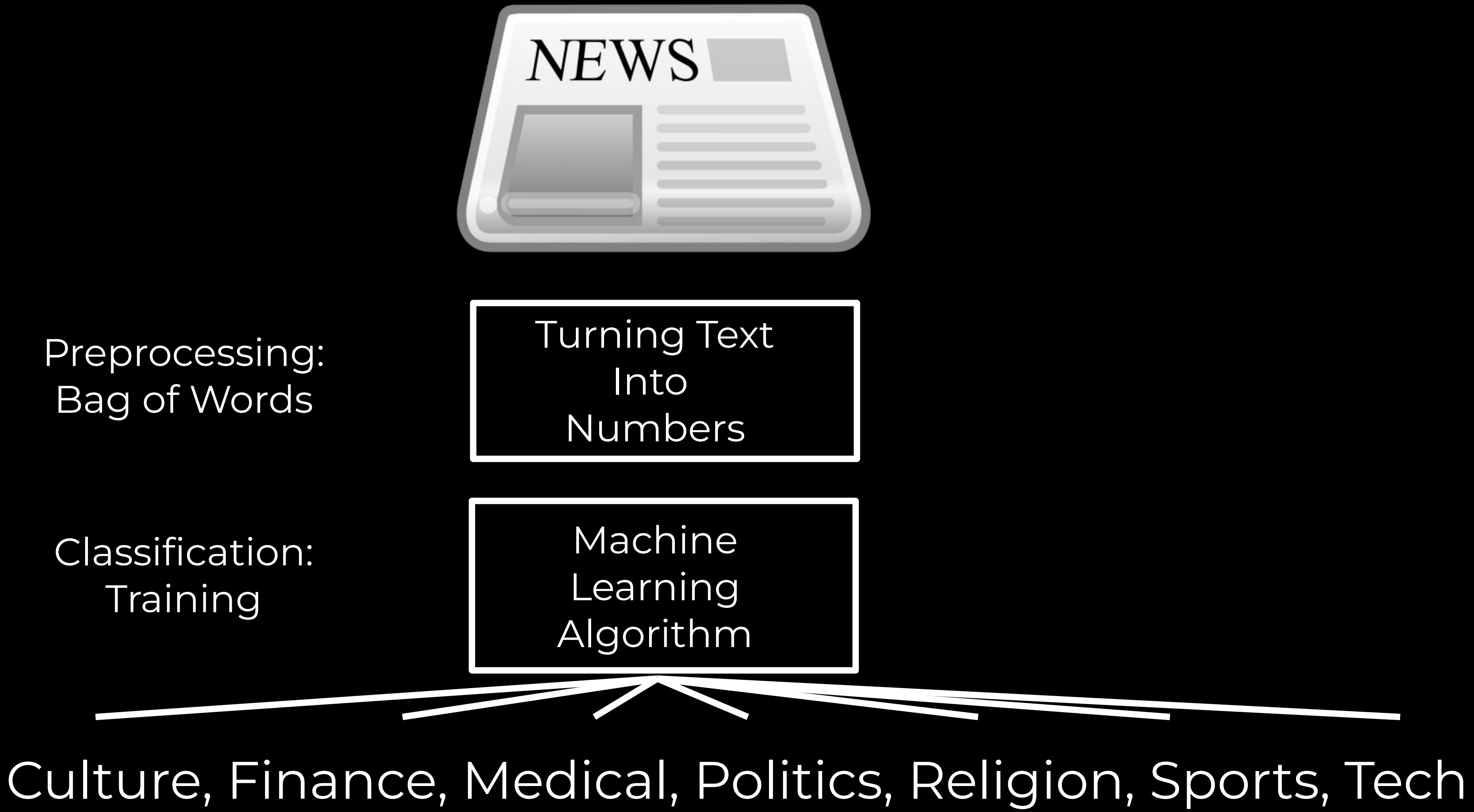
of the types:

Culture, Finance, Medical, Politics, Religion, Sports, Tech

We want to build a Classifier

As our data is limited we will not build ChatGPT today.

We will build a classifier that automatically decides for us into which category a news article goes.



Preprocessing:
Bag of Words

Turning Text
Into
Numbers

Classification:
Training

Machine
Learning
Algorithm

Culture, Finance, Medical, Politics, Religion, Sports, Tech

How to Make Text Machine Readable

A typical dictionary, here in Arabic contains: 120 000 words

For instance this one:
كتاب "تاج العروس" موسوعة ثقافية شاملة من كوثر الغانم الكويت

Turning Text into Features and Samples

- Samples: 1 Sample corresponds to one news article.

- Features: The words in the article will be transformed into feature vectors.

Count Vectorizer

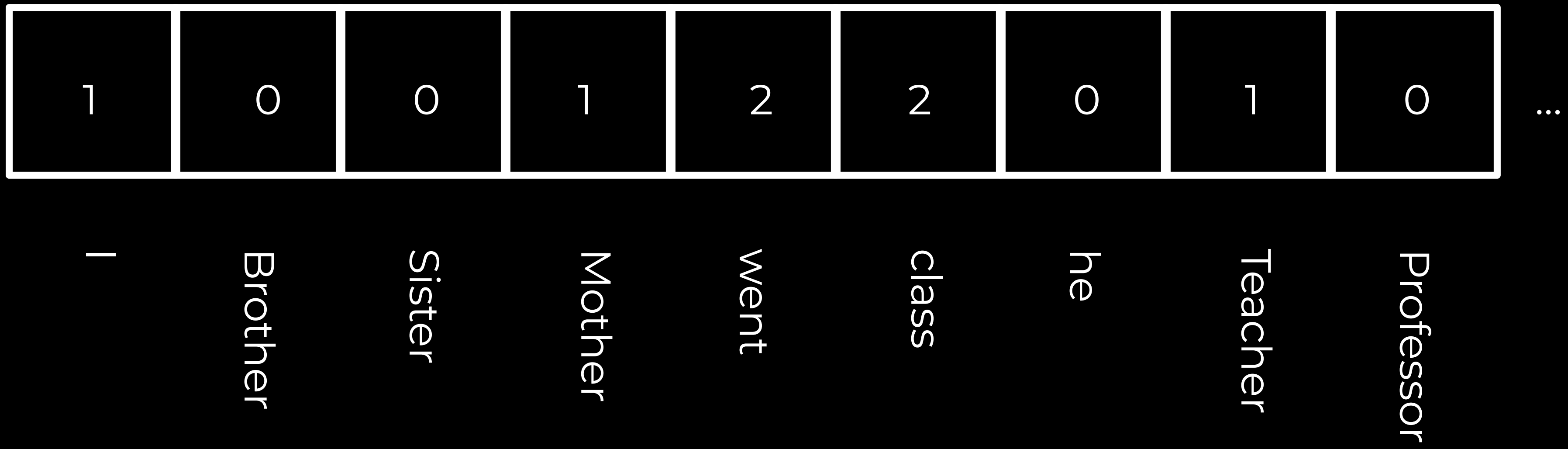
We build vectors that are as long as all the unique words that occur in our text.

For each news article (sample) we count the number a word has occurred and put it into the vector.

Count Vectorizer

Yesterday I went to class. My mother, she is a teacher, went to class too:

i.e. 100 000 words long



I	Brother	Sister	Mother	went	class	he	Teacher	Professor	
1	0	0	1	2	2	0	1	0	...

Count Vectorizer

1	0	0	1	10	2	0	1	0	...
2	1	0	4	2	28	1	8	0	...
5	0	8	1	5	2	5	1	0	...

...

In reality there will be many 0s in the vectors as one news article will only use a few words of the dictionary

Term Frequency (TF)

Problem:
Longer news articles have more words, which artificially creates higher counts. This is bad for our classification problem, where we want the vectors to be representative of a type of news: i.e. Sports

Term Frequency (TF)

Solution:
We divide each entry for each article by the number of words in that article.

Term Frequency (TF)

Yesterday I went to class. My mother, she is a teacher, went to class too: = 14 words

i.e. 100 000 words long

1/14	0	0	1/14	2/14	2/14	0	1/14	0	...
I	Brother	Sister	Mother	went	class	he	Teacher	Professor	

Inverse Document Frequency (IDF)

2nd Problem:
Many words that do not carry a lot of interesting information will be counted very frequent. Think of: I, and, it, is etc.

We want to prioritize on words that are unique to each news article.

Inverse Document Frequency (IDF)

2nd Solution:
For each word in the vector we multiply it further by the factor:

$$\frac{\text{How often the word occurs in this article}}{\text{How often the word occurs in all articles}}$$

Inverse Document Frequency (IDF)

2nd Solution:
As this solution is to drastic in general one takes the logarithm (base 2) of the IDF

$$\frac{\text{How often the word occurs in this article}}{\text{How often the word occurs in all articles}}$$

Let us get practical

In the practical we will

- Download the Arabic news dataset into google colab
- Vectorize it and apply the TF-IDF transforms
- Decide between two available classifier mechanisms by utilizing cross validation
- Build a complete news type predictor