NLP-101-Introduction

Classical ML for NLP

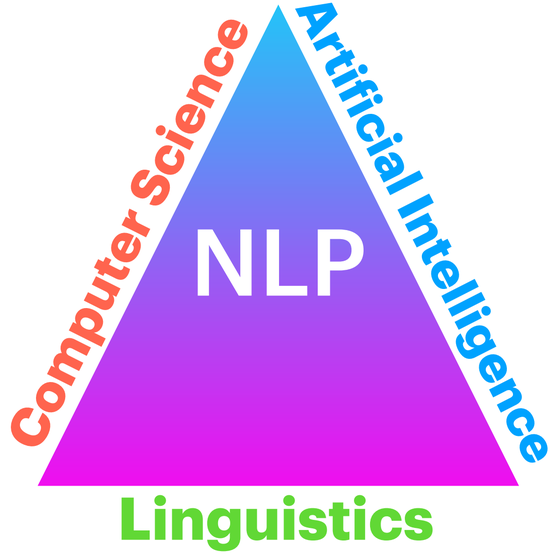
Ali Abdelaal, Language Engineer @Apple

Session Agenda

- Introduction to NLP
- NLP tasks and applications
- Text processing (regex, distance metrics, text normalization)
- Feature extraction (Bag-of-Words, TFIDF, N-grams)
- Text Classification
- Model Inspection and evaluation

Session timeline

What is NLP ?



Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language

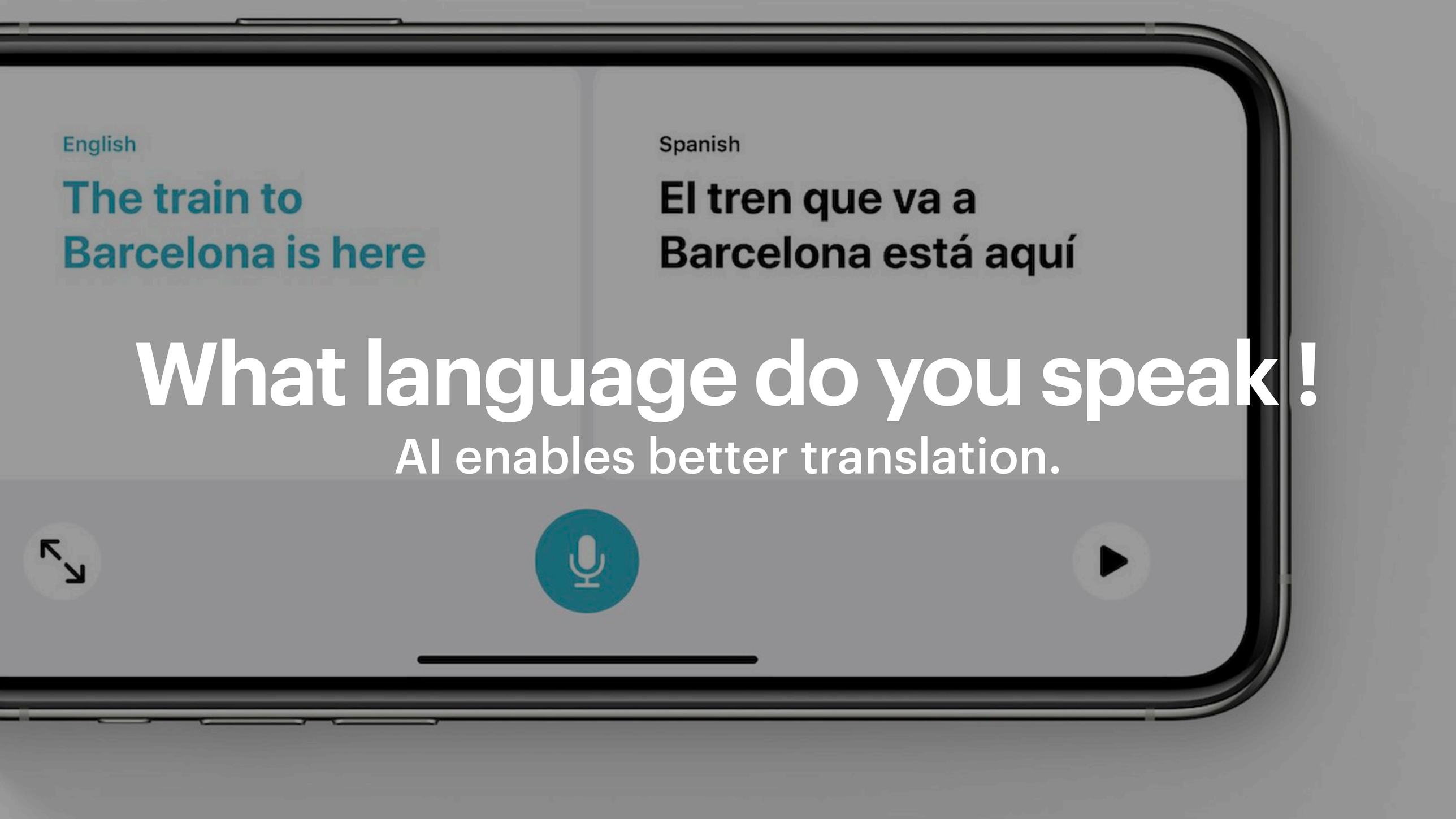
NLP Applications

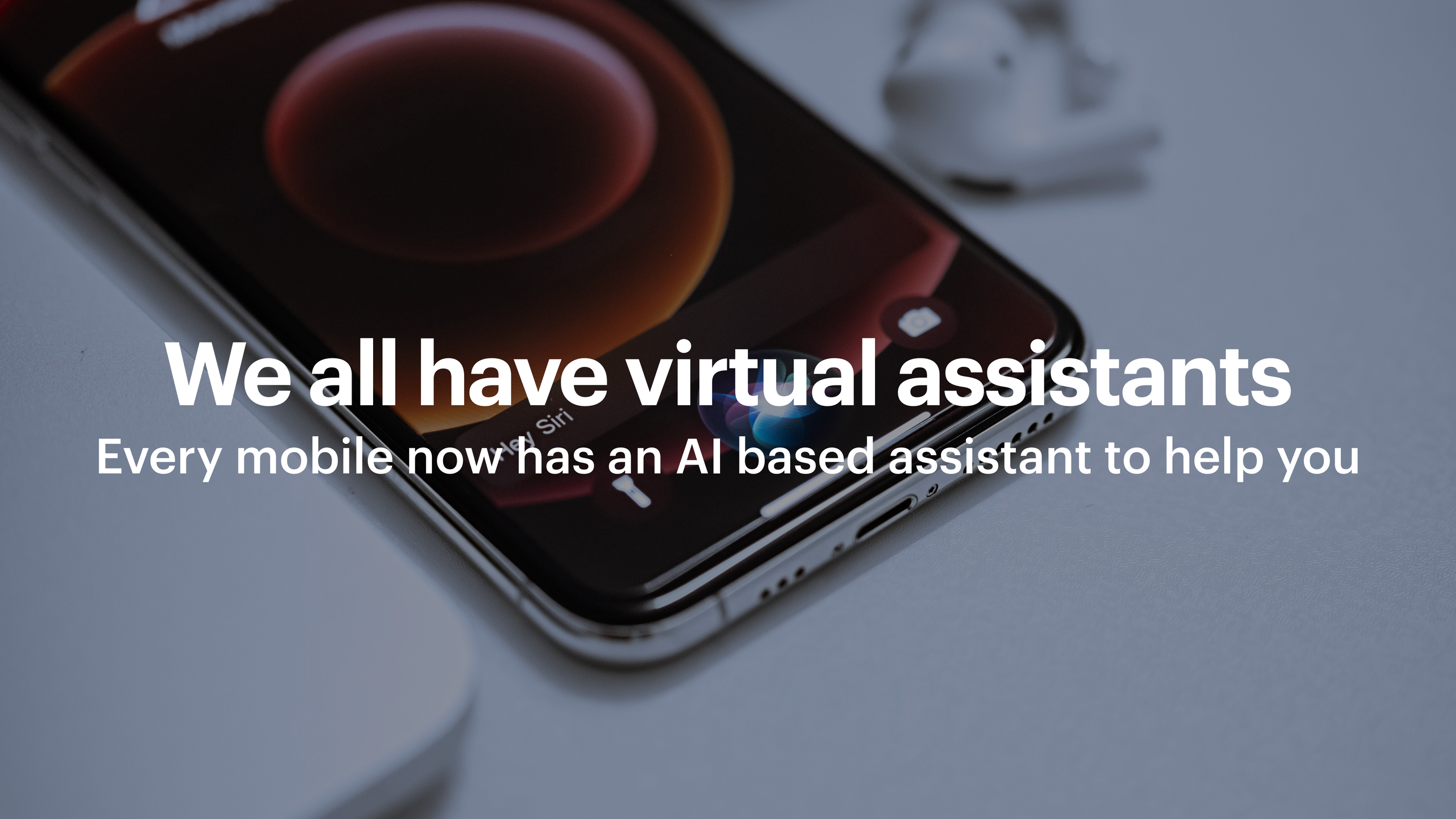
We make use of NLP almost every second !

Google it !

Google uses NLP to process every query


We all make mistakes
Your keyboard keep on learning from your typing and correcting you.

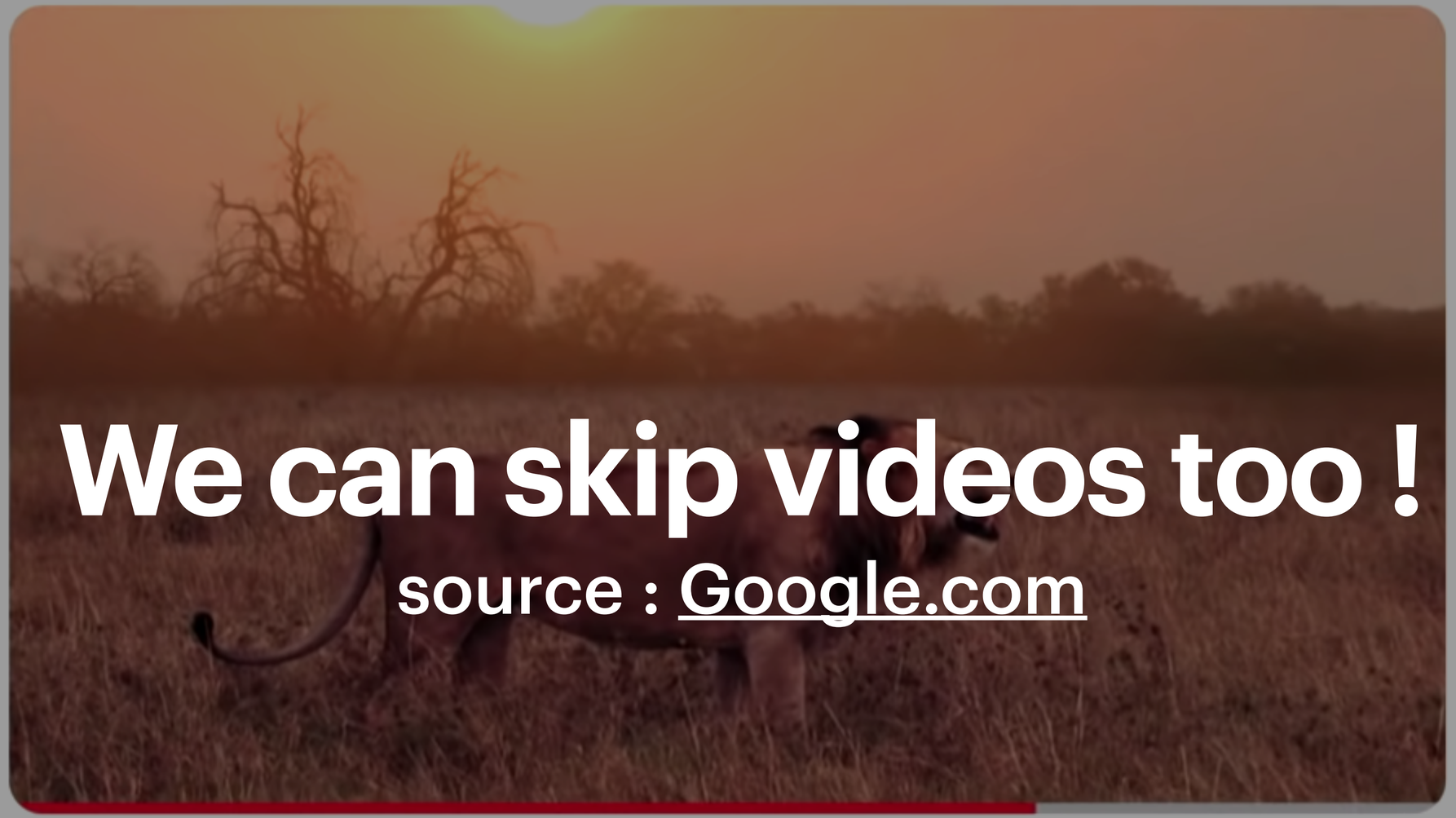

English
The train to
Barcelona is here
Spanish
El tren que va a
Barcelona está aquí
What language do you speak !
AI enables better translation.



We all have virtual assistants

Every mobile now has an AI based assistant to help you

SOURCE: GOOGLE



Show me the part where
the lion roars at sunset

Just to name a few ...

NLP Tasks

Some of the most common NLP tasks

- Text Classification
- Question Answering
- Part-Of-Speech tagging
- Named entity recognition
- Machine Translation
- Language modeling
- and more ...

Notebook - 1

Regular Expressions

Notebook - 2

Edit Distances

Notebook - 3

Text Normalization

Questions ?

Regex, Edit distance and text normalization

Text Representation

All texts are numbers

- We can't feed characters directly to a model
- Text is usually represented in form of numbers like bow or word embeddings
- These numbers are the features that describe the text and we use them to train our model

Bag of Words

- Count the occurrences of each word in the corpus
- Doesn't care about order
- Easy to generate and understand

	cat	dog	eats	food	red	the
the red dog	0	1	0	0	1	1
cat eats dog	1	1	1	0	0	0
dog eats food	0	1	1	1	0	0
red cat eats	1	0	1	0	1	0

Bag of Words with N-grams

- To account for context we can use N-grams
- N-Grams consider multiple (N) words
- N-Grams increase the number of features

	cat	cat eats	dog	dog eats	eats	eats dog	eats food	food	red	red cat	red dog	the	the red
the red dog	0	0	1	0	0	0	0	0	1	0	1	1	1
cat eats dog	1	1	1	0	1	1	0	0	0	0	0	0	0
dog eats food	0	0	1	1	1	0	1	1	0	0	0	0	0
red cat eats	1	1	0	0	1	0	0	0	1	1	0	0	0

Bag of Words with Char N-grams

- We can use characters group instead of words
- This method helps overcome new words by looking up characters
- Feature space can grow exponentially

	ca	cat	cat	do	dog	dog	ea	eat	eats	fo	...	od	og	ood	ood	ot	red	red	the	the	ts
the red dog	0	0	0	1	1	1	0	0	0	0	...	0	1	0	0	0	1	1	1	1	0
cat eats dog	1	1	1	1	1	1	1	1	1	0	...	0	1	0	0	0	0	0	0	0	1
dog eats food	0	0	0	1	1	1	1	1	1	1	...	1	1	1	1	0	0	0	0	0	1
red cat eats	1	1	1	0	0	0	1	1	1	0	...	0	0	0	0	0	1	1	0	0	1
the hot dog	0	0	0	1	1	1	0	0	0	0	...	0	1	0	0	1	0	0	1	1	0

TF-IDF

Text Frequency - Inverse Document Frequency

- Focus on unique words in the corpus
- Give words/n-gram a weight based on it's frequency
- very useful in many domains

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

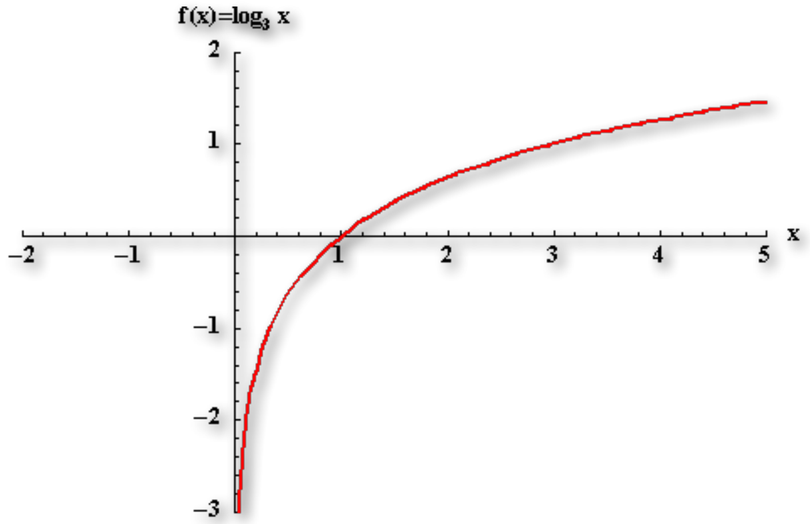
TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents



sources : medium

TF-IDF

Text Frequency - Inverse Document Frequency

- Focus on unique words in the corpus
- Give words/n-gram a weight based on it's frequency
- very useful in many domains

	cat	dog	eats	food	hot	red	the
the red dog	0.000000	0.442740	0.000000	0.000000	0.000000	0.634027	0.634027
cat eats dog	0.677803	0.473309	0.562638	0.000000	0.000000	0.000000	0.000000
dog eats food	0.000000	0.423954	0.503968	0.752515	0.000000	0.000000	0.000000
red cat eats	0.609818	0.000000	0.506204	0.000000	0.000000	0.609818	0.000000
the hot dog	0.000000	0.401565	0.000000	0.000000	0.712775	0.000000	0.575063

Notebook - 4

Feature extraction

Questions ?

Feature extraction

Text Classification

The hello world of NLP

Text Classification

Sentiment analysis

- Carefully investigate the data for
  - imbalance classes
  - imbalance distribution of features/samples
  - integrity between training, validation, testing and production
- Evaluate your model using cross folds
- Be pessimistic about your model

Text Classification

Sentiment analysis

- Will be working with this dataset
- It contains 58K Arabic tweets (47K training, 11K test) tweets annotated in positive and negative labels. The dataset is balanced and collected using positive and negative emojis lexicon.

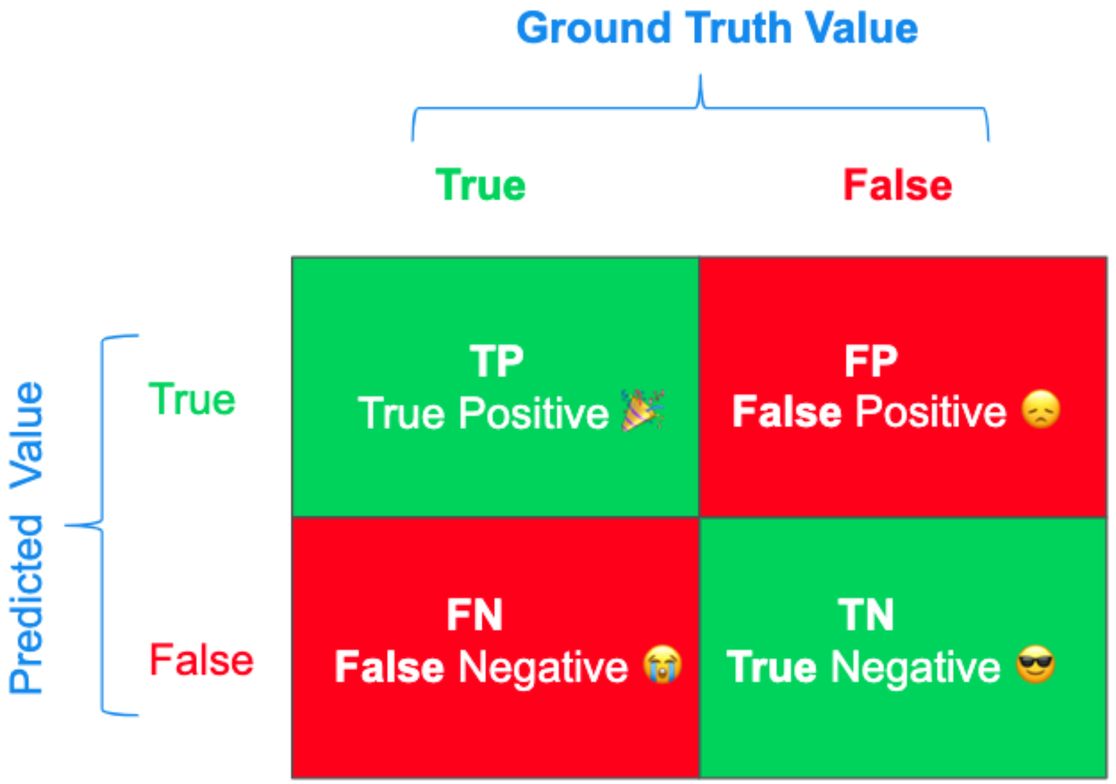
	label	tweet
17985	pos	💙 احب فانتاجيو
25139	neg	😡 شنو ذنب هالطفل لي ماله شغل باجر الناس تكرها ويسمونه ولد حرام
594	pos	♥.. أغلبنا ينظرون للأشياء من منظور واحد ولا نراها من زواية أخرى التي من الممكن ان تقلب مفهومنا للمغزى من وراء الحدث تماما 👌 '#صباح_الخير
31358	neg	💔،،وش_تعلمت_من_الدنياء لم أعد أبالي لشيأ، اختلق لنفسي أعذاراً بأن الحياه جميلة، لكي أعيش قدر المستطاع#
16394	pos	🤣 كنا نسويها قبل بس مو كذا

Evaluation

For the model not you, don't worry

Precision
Model Evaluation

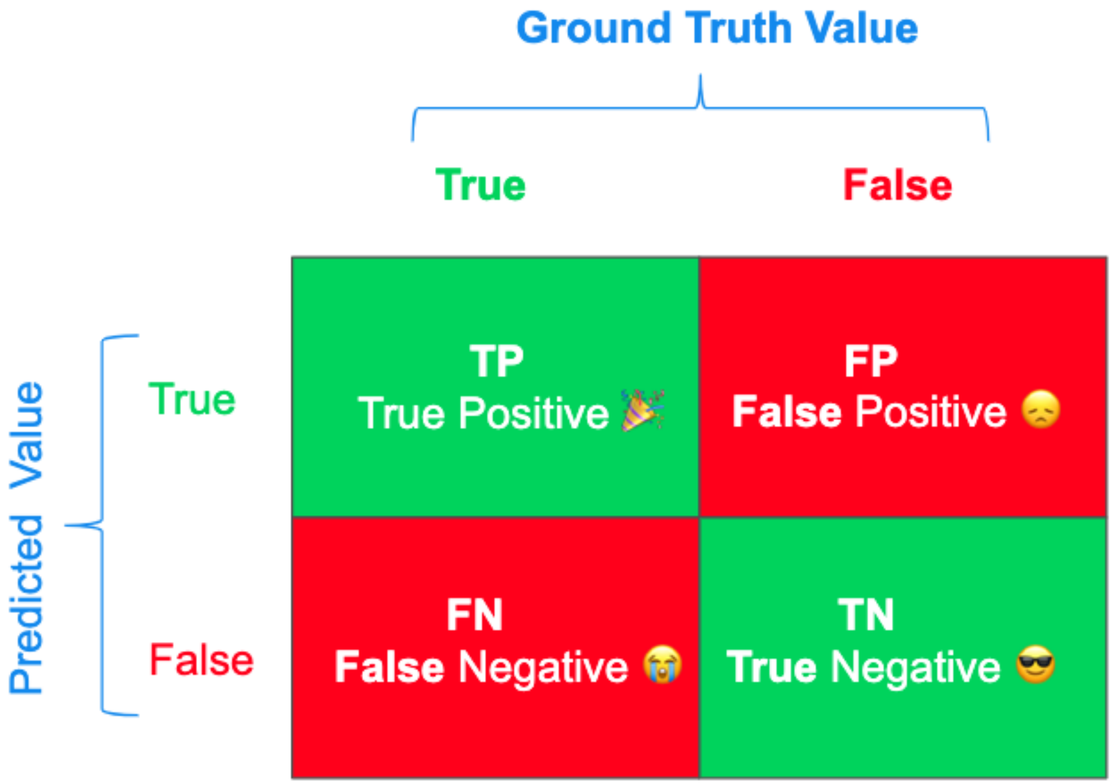
- The precision is the ratio **tp / (tp + fp)** where **tp** is the number of true positives and **fp** the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.
- The best value is 1 and the worst value is 0.



sources : sklearn, floydhub

Recall
Model Evaluation

- The recall is the ratio **tp / (tp + fn)** where **tp** is the number of true positives and **fn** the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.
- The best value is 1 and the worst value is 0.



sources : sklearn, floydhub

F1-score
Model Evaluation

- The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

sources : sklearn

Multi class Averaging
Model Evaluation

- Working with multiple classes will generate multiple precision/recall/f1
- To get an overall value for any metric we need to average these metrics across classes
- We will explore Micro, Macro and weighted averaging.

		True/Actual		
		Cat	Fish	Hen
Predicted	Cat	4	6	3
	Fish	1	2	0
	Hen	1	2	6

sources : medium

Macro Average

Multi class Averaging

- To calculate the macro average we take into account how many classes we have and average across them
- Macro-precision = (31% + 67% + 67%) / 3 = **54.7**%
- Macro-recall = (67% + 20% + 67%) / 3 = **51.1**%
- Macro-F1 = (42.1% + 30.8% + 66.7%) / 3 = **46.5**%

		True/Actual		
		Cat	Fish	Hen
Predicted	Cat	4	6	3
	Fish	1	2	0
	Hen	1	2	6

Class	Precision	Recall	F1-score
Cat	30.8%	66.7%	42.1%
Fish	66.7%	20.0%	30.8%
Hen	66.7%	66.7%	66.7%

sources : medium

Weighted Average

Multi class Averaging

- Here we weight each class by the number of samples it has, here we have 25 samples in total
- Weighted-precision = (6 × 30.8% + 10 × 66.7% + 9 × 66.7%)/25 = **58.1**%
- Weighted-recall = (6 × 66.7% + 10 × 20.0% + 9 × 66.7%) / 25 = **48.0**%
- Weighted-F1 = (6 × 42.1% + 10 × 30.8% + 9 × 66.7%) / 25 = **46.4**%

		True/Actual		
		Cat	Fish	Hen
Predicted	Cat	4	6	3
	Fish	1	2	0
	Hen	1	2	6

Class	Precision	Recall	F1-score
Cat	30.8%	66.7%	42.1%
Fish	66.7%	20.0%	30.8%
Hen	66.7%	66.7%	66.7%

sources : medium

Micro Average

Multi class Averaging

- To compute the micro average we look all samples together as if all were one class
- TP = 4 + 2 + 6 = 12
- FP = FN = 6 + 3 3 1 + 0 + 1 + 2=13
- Micro-precision = 12/(12+13) = **48.0**%
- Micro-recall = 12/(12+13) = **48.0**%
- Micro-F1 = Micro-precision = Micro-recall = **48.0**%

sources : medium

		True/Actual		
		Cat (🐱)	Fish (🐟)	Hen (🐔)
Predicted	Cat (🐱)	4	6	3
	Fish (🐟)	1	2	0
	Hen (🐔)	1	2	6

Class	Precision	Recall	F1-score
Cat	30.8%	66.7%	42.1%
Fish	66.7%	20.0%	30.8%
Hen	66.7%	66.7%	66.7%

Notebook - 5

Text Classification

Questions ?

Text Classification

Let's build a quick API

A sentiment analysis API


Thank you !