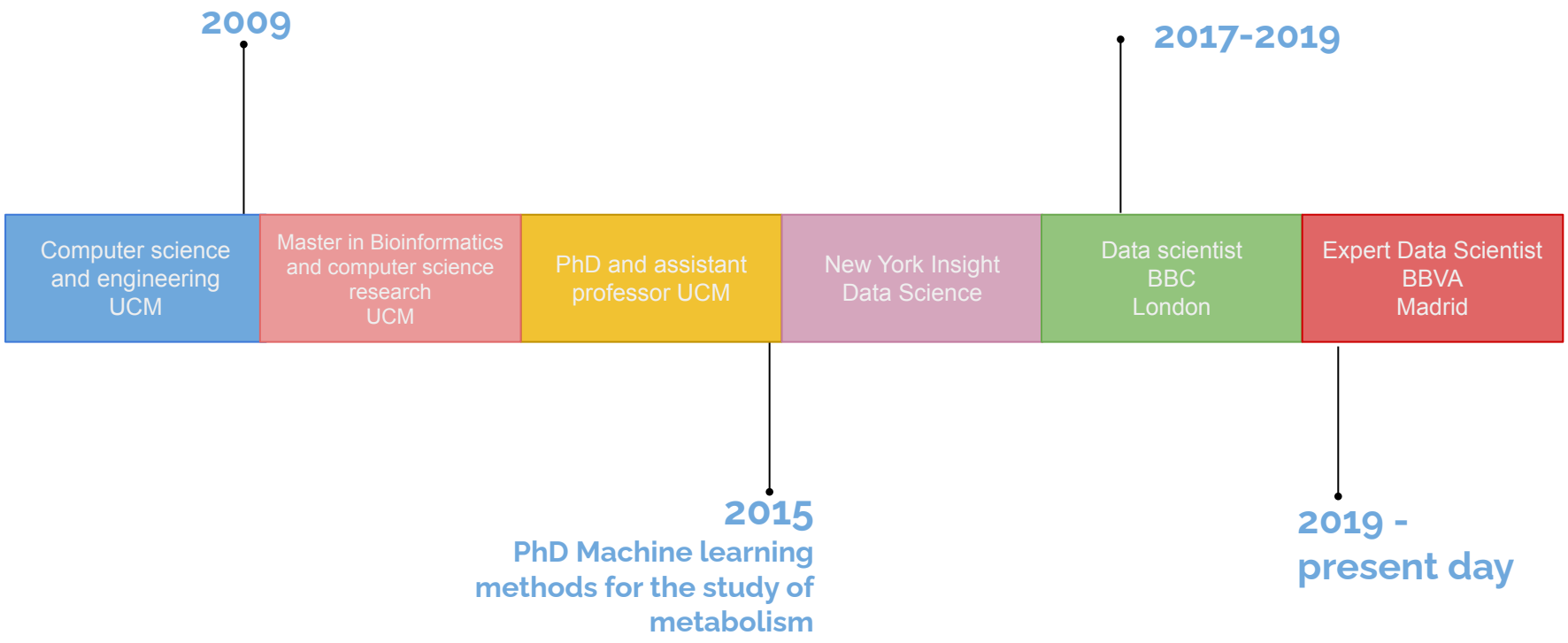
Introduction to Natural language processing

Session 1

Clara Higuera Cabañes, PhD
Barcelona Technology School, May 2021

About me


2009
2017-2019
Computer science and engineering UCM
Master in Bioinformatics and computer science research UCM
PhD and assistant professor UCM
New York Insight Data Science
Data scientist BBC London
Expert Data Scientist BBVA Madrid
2015
PhD Machine learning methods for the study of metabolism
2019 - present day

Class philosophy

Index

1. What is NLP?
2. NLP pipeline
    a. Data Acquisition
    b. Cleaning and preprocessing
    c. Text representation / Feature engineering

Break (9:50, 10min)

3. Hands on / live showcase nlp in action (20-30min)
4. NLP pipeline (30 min)
    a. Modeling
    b. Evaluation of supervised learning problems

Break 10 mins

Quiz (15 min, 10-15 mins review collaboratively)

5. Hands on / live showcase nlp in action (30 min)
6. NLP real use case
    a. Manual annotation

What is Natural Language Processing (NLP)?

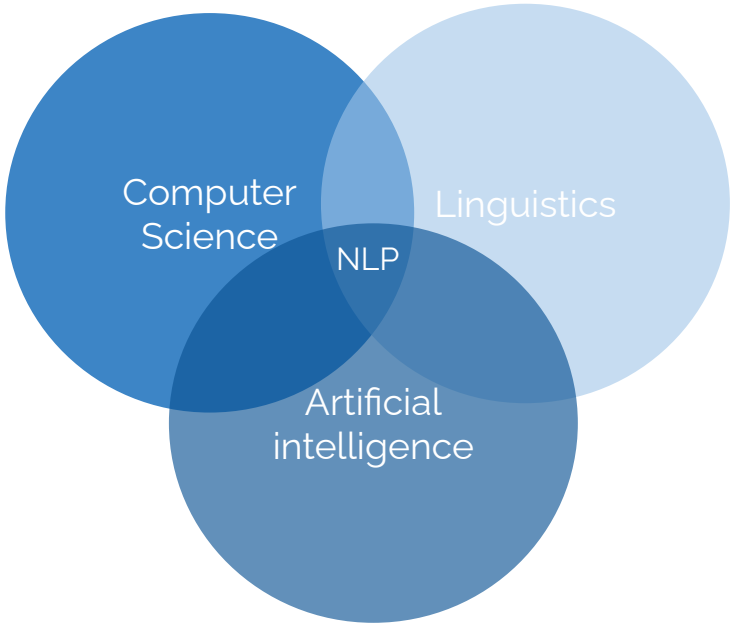
John: “How is the weather today?”

Digital assistant: “It is 37 degrees centigrade outside with no rain today.”

John: “What does my schedule look like?”

Digital assistant: “You have a strategy meeting at 4 p.m. and an all-hands at 5:30 p.m. Based on today’s traffic situation, it is recommended you leave for the office by 8:15 a.m.”

What is Natural Language Processing?



It concerns building systems that can process and understand human language.

Sectors: retail, healthcare, finance, media, law, marketing, human resources, and many more.

Applications of NLP

Customer service

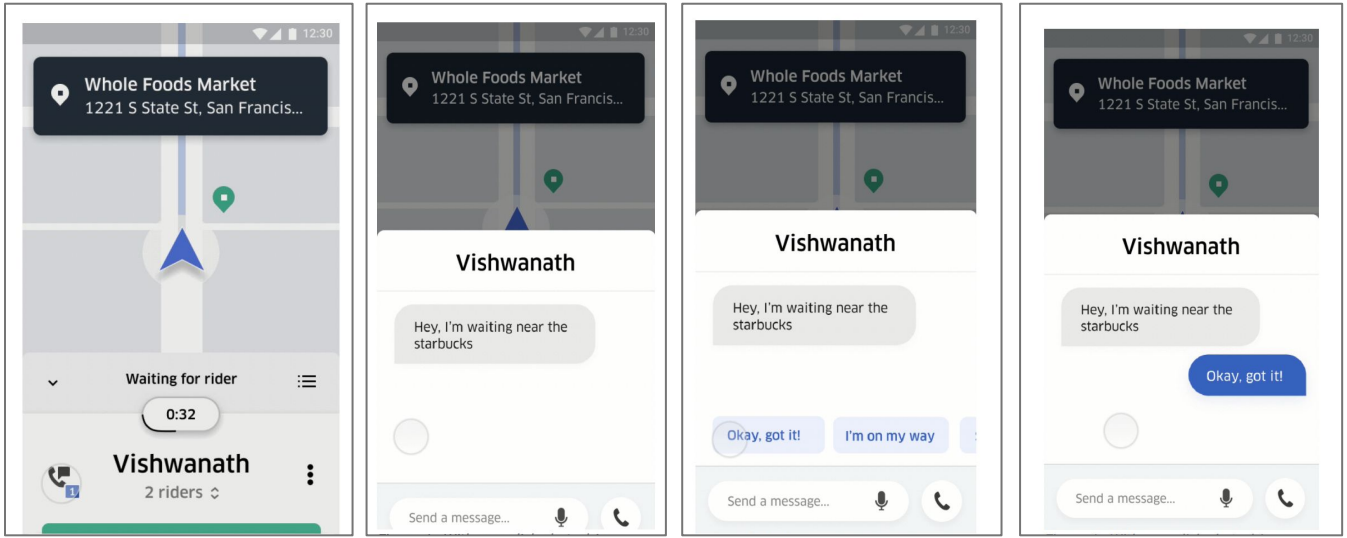
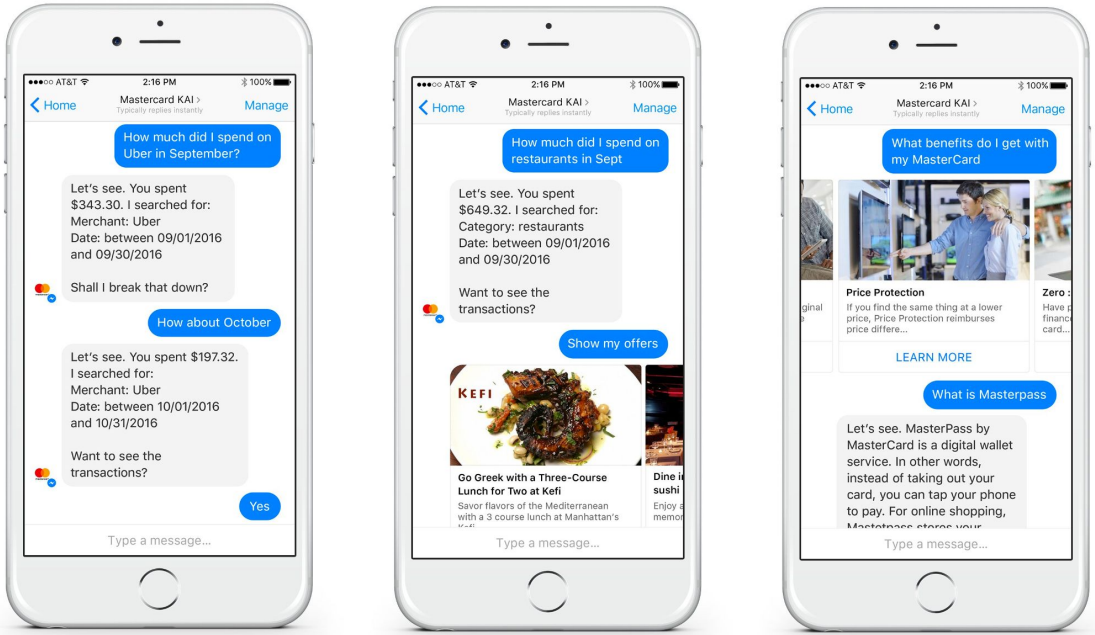


Figure 1: With one-click chat, driver-partners can more easily respond to rider messages.

Uber one-click-chat

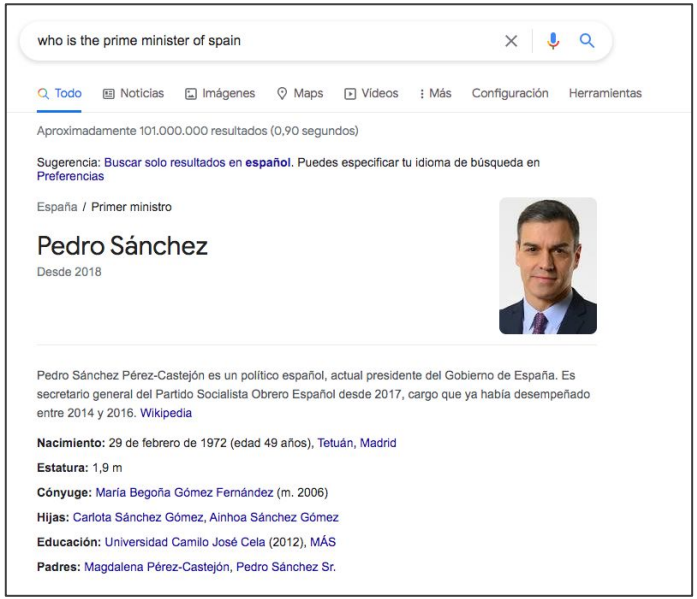
Applications of NLP

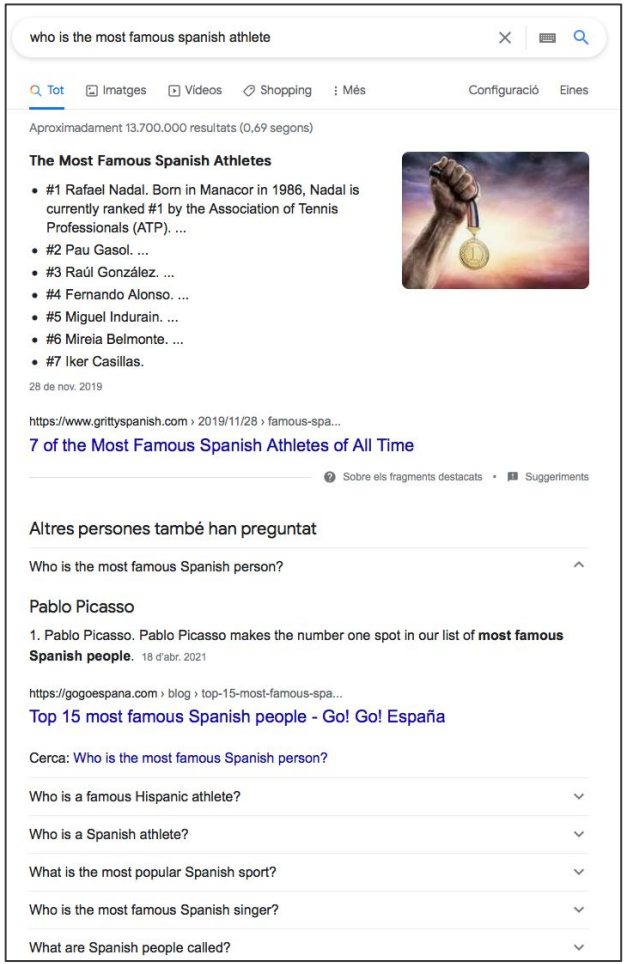
Customer service



Applications of NLP

Question answering

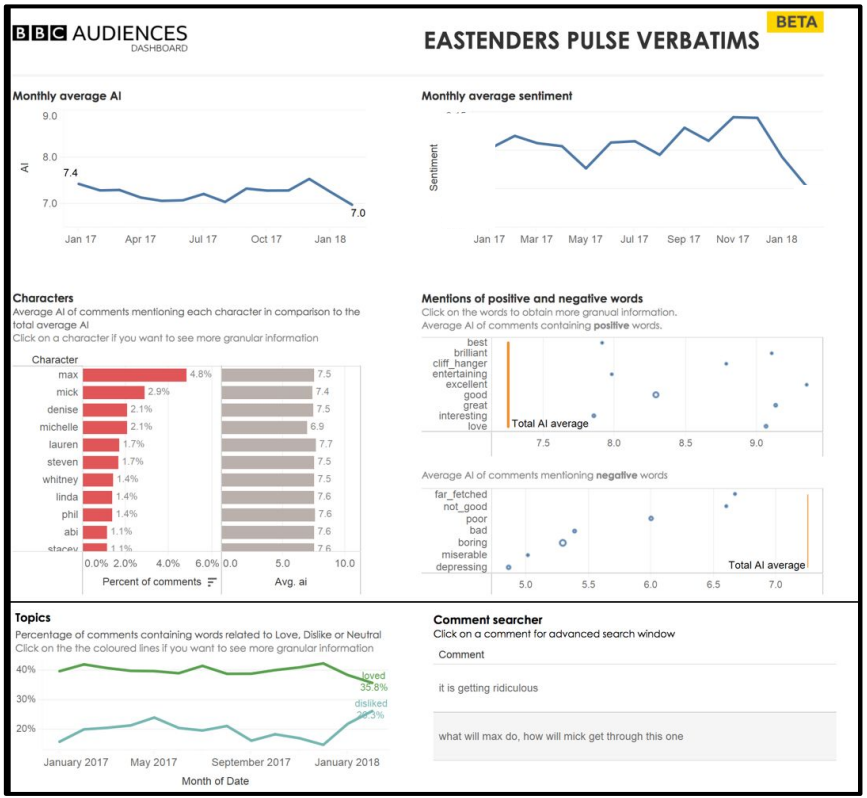




Applications of NLP

Media

BBC



Applications of NLP

Media

You might be interested as well in:

Content similarity recommenders

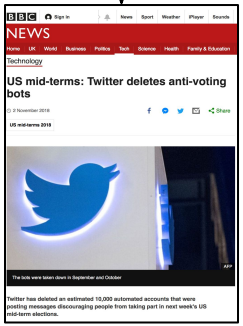
Applications of NLP

Media

You might be interested as well in:

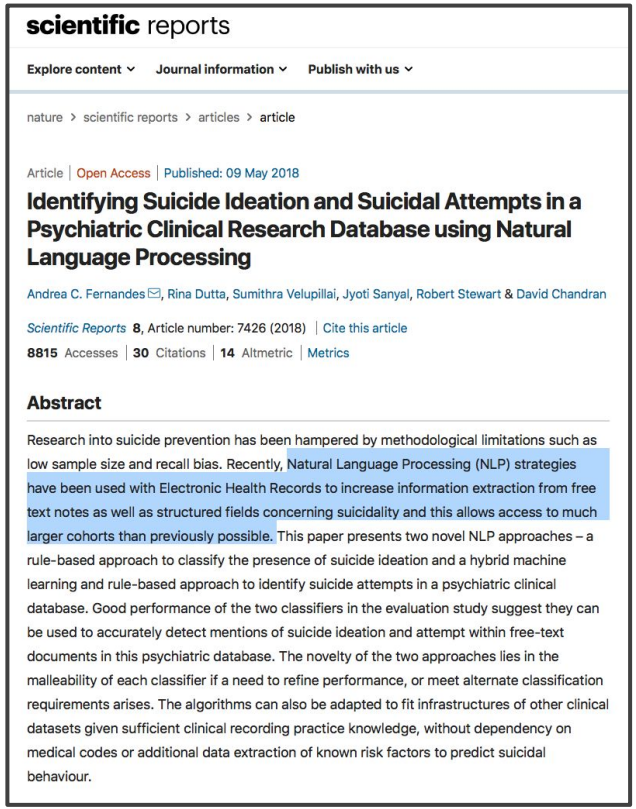
Content similarity recommenders

Which article is more similar?



Applications of NLP

Medicine and social science



scientific reports

Explore content | Journal information | Publish with us

nature › scientific reports › articles › article

Article | Open Access | Published: 09 May 2018

Identifying Suicide Ideation and Suicidal Attempts in a Psychiatric Clinical Research Database using Natural Language Processing

Andrea C. Fernandes, Rina Dutta, Sumithra Velupillai, Jyoti Sanyal, Robert Stewart & David Chandran

Scientific Reports **8**, Article number: 7426 (2018) | Cite this article

8815 Accesses | **30** Citations | **14** Altmetric | Metrics

Abstract

Research into suicide prevention has been hampered by methodological limitations such as low sample size and recall bias. Recently, Natural Language Processing (NLP) strategies have been used with Electronic Health Records to increase information extraction from free text notes as well as structured fields concerning suicidality and this allows access to much larger cohorts than previously possible. This paper presents two novel NLP approaches – a rule-based approach to classify the presence of suicide ideation and a hybrid machine learning and rule-based approach to identify suicide attempts in a psychiatric clinical database. Good performance of the two classifiers in the evaluation study suggest they can be used to accurately detect mentions of suicide ideation and attempt within free-text documents in this psychiatric database. The novelty of the two approaches lies in the malleability of each classifier if a need to refine performance, or meet alternate classification requirements arises. The algorithms can also be adapted to fit infrastructures of other clinical datasets given sufficient clinical recording practice knowledge, without dependency on medical codes or additional data extraction of known risk factors to predict suicidal behaviour.

Applications of NLP

Reducing response times to citizen legal questions across Africa

barefootlaw

Data Science for Social Good. Chicago University

Understanding Text Data to Help Disadvantaged Families

Understanding Text Data to Help Disadvantaged Families

DataKind UK

https://www.meetup.com/DataForGoodBCN/

Government

Using AI to unleash the power of unstructured government data

Improving forensics investigations

A growing number of government agencies are using NLP-based solutions to improve investigations in critical areas such as law enforcement, defense, and intelligence. The DoD's DEFT program referenced above uses NLP to uncover connections implicit in large text documents. Its objective is to improve the efficiency of defense analysts who investigate multiple documents to detect anomalies and causal relationships.[36]

The European Union's Horizon 2020 program launched an initiative called RED (Real-time Early Detection) Alert, aimed at countering terrorism by using NLP to monitor and analyze social media conversations. RED Alert is designed to provide early alerts of potential propaganda and signs of warfare by identifying online content posted by extremists. To comply with the General Data Protection Regulation (GDPR), this analysis uses homomorphic encryption, a method that allows mathematical operations to be performed on encrypted text without disturbing the original encryption.[37]

Enhancing policy analysis

The World Bank's Poverty and Equity Global Practice Group used LDA topic modeling to measure changes in policy priorities by examining presidential speeches in 10 Latin American countries and Spain from 1819 to 2016. Using LDA, the authors could identify the main topics for each document and indicate the variation in their significance across countries and over time. In Peru, for instance, topics on infrastructure and public services diminished in importance over time. With the help of topic modeling, the authors were able to establish, for each nation, a negative correlation between policy volatility and long-term growth.[41]

Improving predictions to aid decision-making

One of the most striking characteristics of NLP is its ability to facilitate better predictions, which can help agencies design preemptive measures. The police department of Durham, North Carolina, uses NLP in crimefighting by enabling the police to observe patterns and interrelations in criminal activities and identify pockets with a high incidence of crime, thus allowing for quicker interventions. This contributed to a 39 percent drop in violent crime in Durham from 2007 to 2014.[38]

NLP also is being used to combat child trafficking. About 75 percent of child trafficking involves online advertisements. DARPA, in collaboration with commercial technology experts, has developed a platform that monitors and draws connections among the dubious content of online advertisements. Virginia's Fairfax County Police Department and New Orleans's Homeland Security investigations both use this advanced software to identify high-risk web advertisements and detect code words used by traffickers.[39]

NLP is developing quickly

- **Widely available and easy-to-use NLP tools,** techniques, and APIs are now all-pervading in the industry. There has never been a better time to build quick NLP solutions.
- **Development of more interpretable and generalized approaches** has improved the baseline performance for even complex NLP tasks, such as open-domain conversational tasks and question answering, which were not practically feasible before.
- More and **more organizations**, including Google, Microsoft, and Amazon, **are investing heavily in more interactive consumer products, where language** is used as the primary medium of communication.
- **Increased availability of useful open source datasets**, along with standard benchmarks on them, has acted as a catalyst in this revolution, as opposed to being impeded by proprietary datasets only available to limited organizations and individuals.
- **The viability of NLP has moved beyond English or other major languages.** Datasets and language-specific models are being created for the less-frequently digitized languages too. A fruitful product that came out this effort was a near-perfect automatic machine translation tool available to all individuals with a smartphone.
- The **amount of data available plus the advances in deep learning** is 'opening' a new era in NLP

NLP capabilities

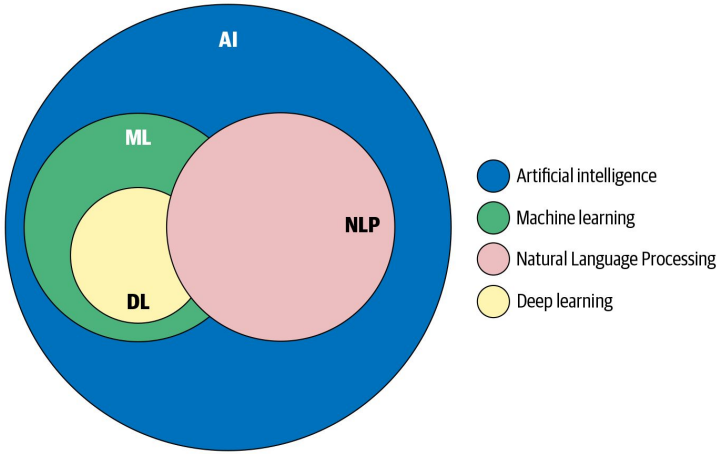
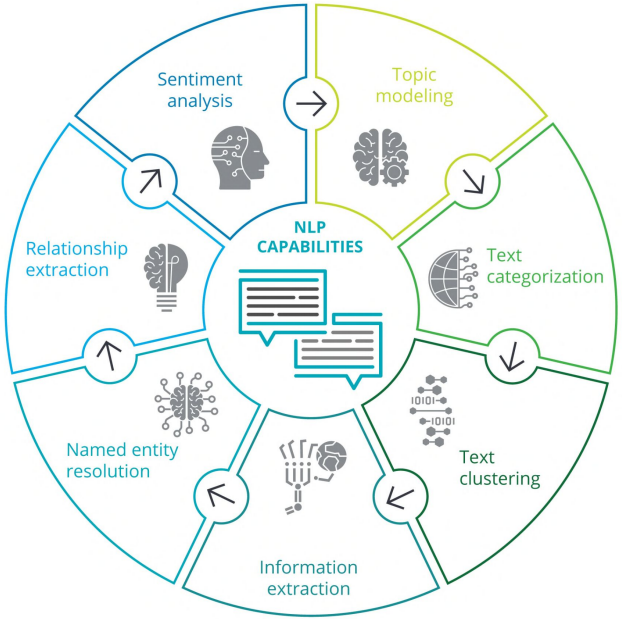


FIGURE 2

Key NLP capabilities



Source: Deloitte analysis.

Natural Language Generation

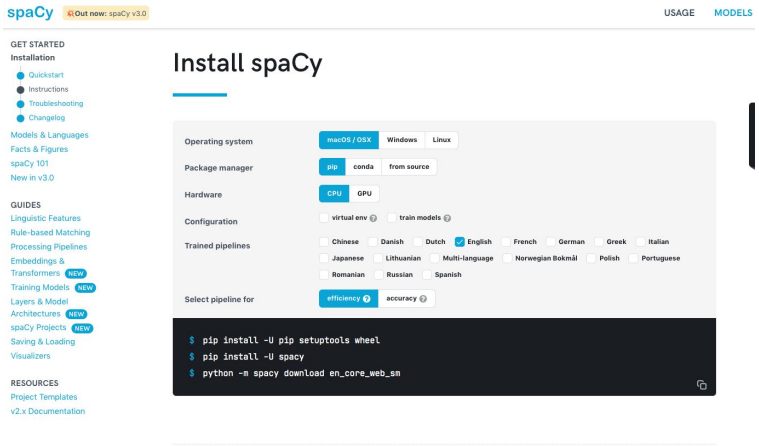
Question answering

Text Summarization

Machine Translation

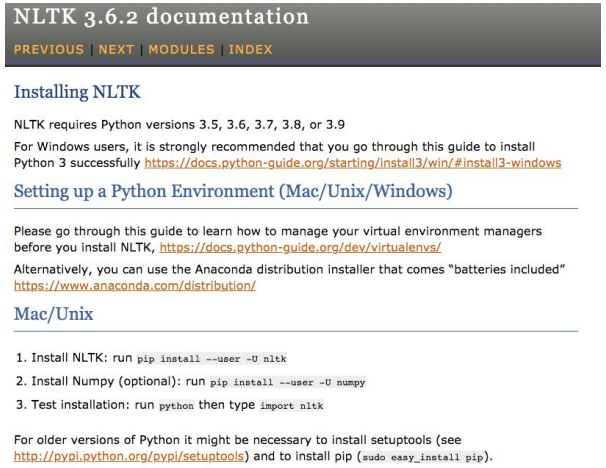
NLP Tools and Libraries

spaCy



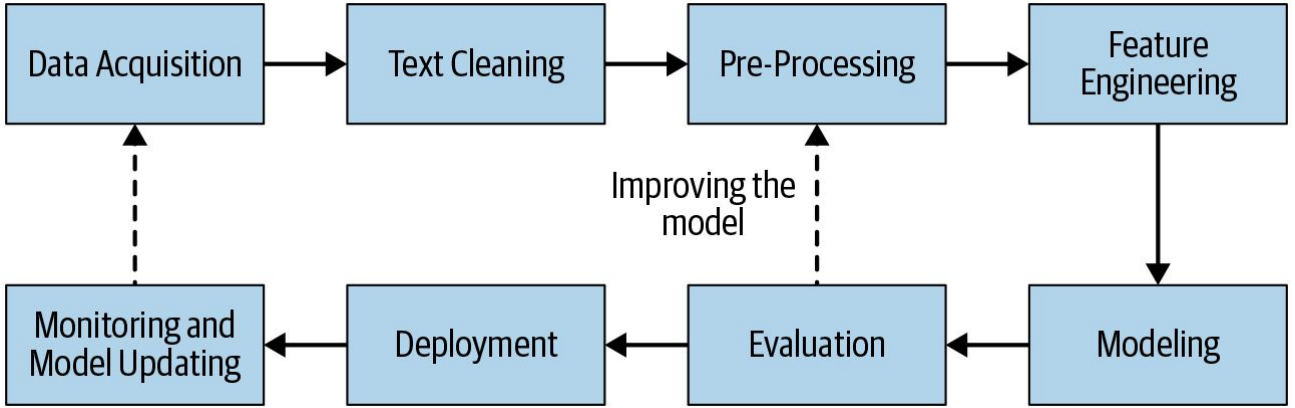
https://spacy.io/usage

NLTK



https://www.nltk.org/install.html

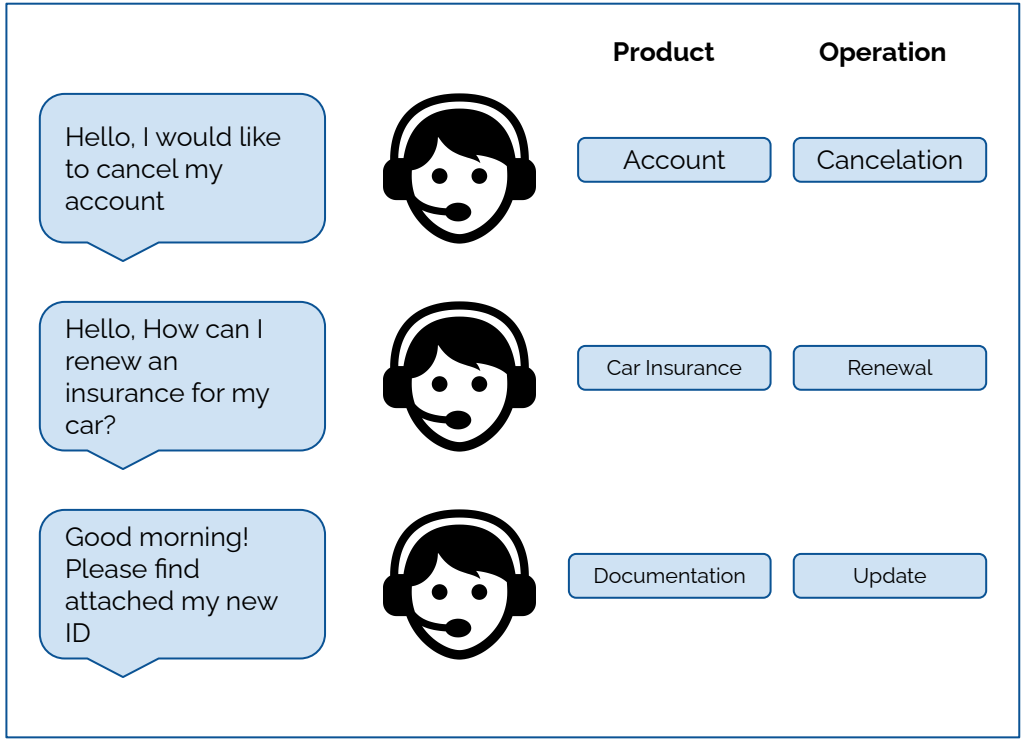
NLP Pipeline



NLP Pipeline

Data acquisition

- Use a public dataset
- Scrape data
- Product intervention / sensorization
- Data Augmentation -> refer to paper from EMNLP!
    - Back translation
    - Active learning



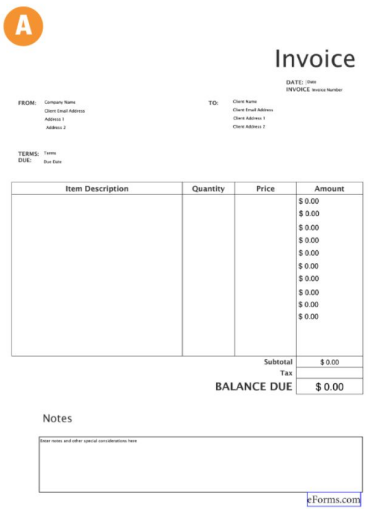
NLP Pipeline

Text extraction and cleaning

- Remove markdown (html)
- Extract text embedded with images
- Remove special characters

unidecode!

A



B

```
<p style=" text-align: justify;">
The book will be around 350 pages. It will be accompanied by a code
repository containing several Jupyter notebooks for all the chapters to
give a walk-through and explain the code in detail. The code base is in
Python and various machine learning and natural language processing
libraries. The book assumes that the readers have a good grasp of
programming but no theoretical and practical knowledge of NLP.
</p>

    <p><br>
    </p>


<a
href="https://www.oreilly.com/library/view/practical-natural-language/97
81492054047/">
<img src="book-link-1.png" alt="Book Link">
</a>

<h1 style="color: #e74c3c;">Commonly Asked Questions</h1>
<ul>
<li> Can I contribute to the book?
<p>The book is accompanied by open source Jupyter notebooks and demo
applications. If you are a great ML or front-end engineer looking to
build something meaningful you can apply by filling <a
href="https://goo.gl/forms/do6NcW251iX26ajk1">this form</a>. Also refer
to the next question.
```

```
from bs4 import BeautifulSoup
from urllib.request import urlopen
myurl = "https://stackoverflow.com/questions/415511/   \
  how-to-get-the-current-time-in-python"
html = urlopen(myurl).read()
soupified = BeautifulSoup(html, "html.parser")
question = soupified.find("div", {"class": "question"})
questiontext = question.find("div", {"class": "post-text"})
print("Question: \n", questiontext.get_text().strip())
answer = soupified.find("div", {"class": "answer"})
answertext = answer.find("div", {"class": "post-text"})


    print("Best answer: \n", answertext.get_text().strip())
```

NLP Pipeline

Text preprocessing

Depends highly on the use case

- Language detection (polyglot)
- Text normalization
  - Convert all text to lowercase or uppercase
  - Convert digits to text
  - Normalize date format
- Remove punctuation, digits
- **Remove stop words**
- Sentence segmentation
- Stemming / lemmatization
- Word tokenization
- Bigram / trigram..
- Advanced processing
  - POS tagging, entity recognition

Sentence segmentation

```
text = u"This is first sentence. Second sentence. Third sentence."
text_sentences = nlp(text)
for sentence in text_sentences.sents:
   print(sentence.text)
```

```
This is first sentence.
Second sentence.
Third sentence.
```

NLP Pipeline

Text preprocessing

Depends highly on the use case

- Language detection (polyglot)
- Text normalization
  - Convert all text to lowercase or uppercase
  - Convert digits to text
  - Normalize date format
- Remove punctuation, digits
- **Remove stop words**
- Sentence segmentation
- Stemming / lemmatization
- Word tokenization
- Bigram / trigram..
- Advanced processing
  - POS tagging, entity recognition

Stop words

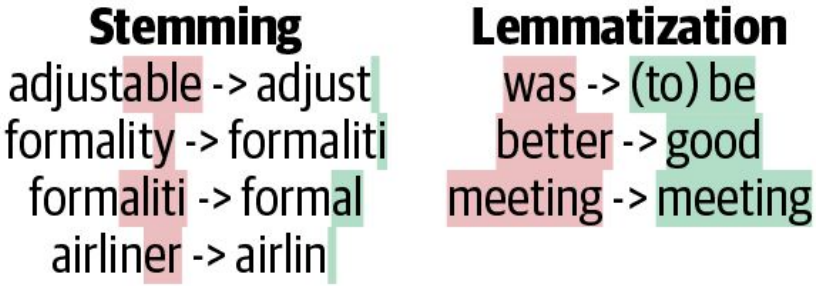
> *Generally, the most common words used in a text are "the", "is", "in", "for", "where", "when", "to", "at" etc.*

NLP Pipeline

Text preprocessing

Depends highly on the use case

- Language detection (polyglot)
- Text normalization
  - Convert all text to lowercase or uppercase
  - Convert digits to text
  - Normalize date format
- Remove punctuation, digits
- Remove stop words
- Sentence segmentation
- **Stemming / lemmatization**
- Word tokenization
- Bigram / trigram..
- Advanced processing
  - POS tagging, entity recognition




```
nlp = spacy.load( 'en_core_web_sm' )
doc = nlp(u 'I am Clara and this is the NLP class' )
#for token in doc:
#    print(token.text, token.lemma_, token.is_stop)

for token in doc:
    print(token.text, token.lemma_)
```

```
(u'I', u'-PRON-', True)
(u'am', u'be', True)
(u'Clara', u'Clara', False)
(u'and', u'and', True)
(u'this', u'this', True)
(u'is', u'be', True)
(u'the', u'the', True)
(u'NLP', u'NLP', False)
(u'class', u'class', False)
```

NLP Pipeline

Text preprocessing

Depends highly on the use case

- Language detection (polyglot)
- Text normalization
  - Convert all text to lowercase or uppercase
  - Convert digits to text
  - Normalize date format
- Remove punctuation, digits
- Remove stop words
- Sentence segmentation
- Stemming / lemmatization
- Word tokenization
- Bigram / trigram..
- **Advanced processing**
  - POS tagging, entity recognition

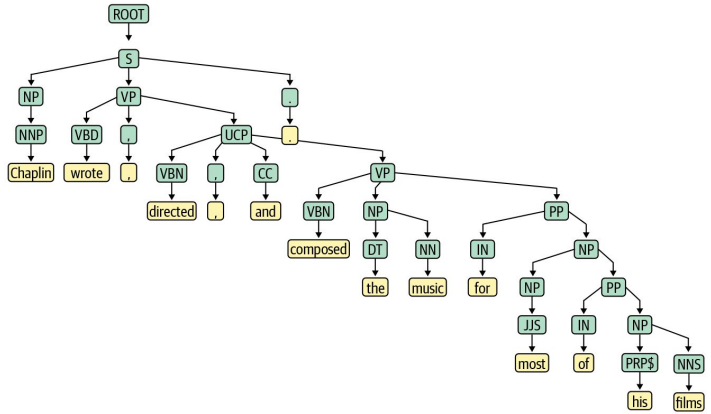
Input

Chaplin wrote, directed, and composed the music for most of his films.

Tokenization with Lemmatization

POS Tagging

Parse Tree



NLP Pipeline

Text preprocessing

Depends highly on the use case

- Language detection (polyglot)
- Text normalization
  - Convert all text to lowercase or uppercase
  - Convert digits to text
  - Normalize date format
- Remove punctuation, digits
- Remove stop words
- Sentence segmentation
- Stemming / lemmatization
- Word tokenization
- Bigram / trigram..
- **Advanced processing**
  - POS tagging, entity recognition

```
nlp = spacy.load('en_core_web_sm')

doc = nlp(u'I am Clara and this is the NLP class')

for token in doc:
    print(token.text, token.lemma_, token.is_stop, token.pos_)
```

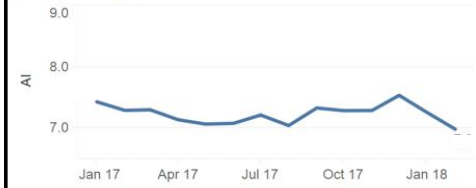
```
(u'I', u'-PRON-', True, u'PRON')
(u'am', u'be', True, u'VERB')
(u'Clara', u'Clara', False, u'PROPN')
(u'and', u'and', True, u'CCONJ')
(u'this', u'this', True, u'DET')
(u'is', u'be', True, u'VERB')
(u'the', u'the', True, u'DET')
(u'NLP', u'NLP', False, u'PROPN')
(u'class', u'class', False, u'NOUN')
```

BBC AUDIENCES DASHBOARD

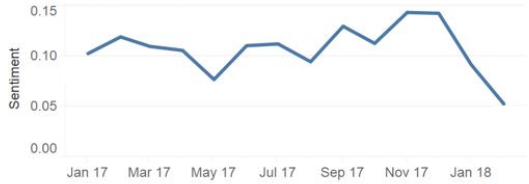
EASTENDERS PULSE VERBATIMS

BETA

Monthly average AI



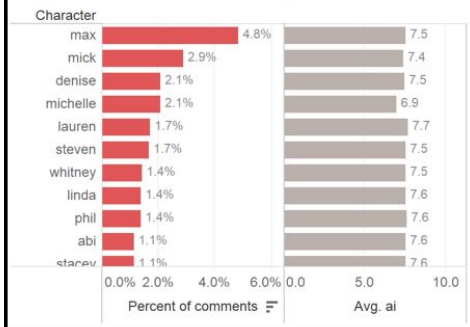
Monthly average sentiment



Characters

Average AI of comments mentioning each character in comparison to the total average AI

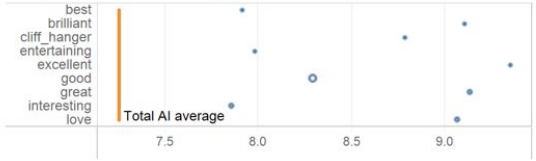
Click on a character if you want to see more granular information



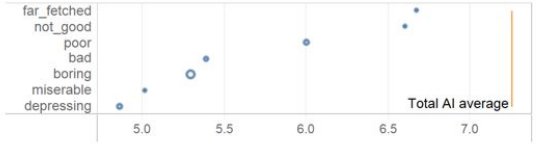
Mentions of positive and negative words

Click on the words to obtain more granual information.

Average AI of comments containing **positive** words.



Average AI of comments mentioning **negative** words



Topics

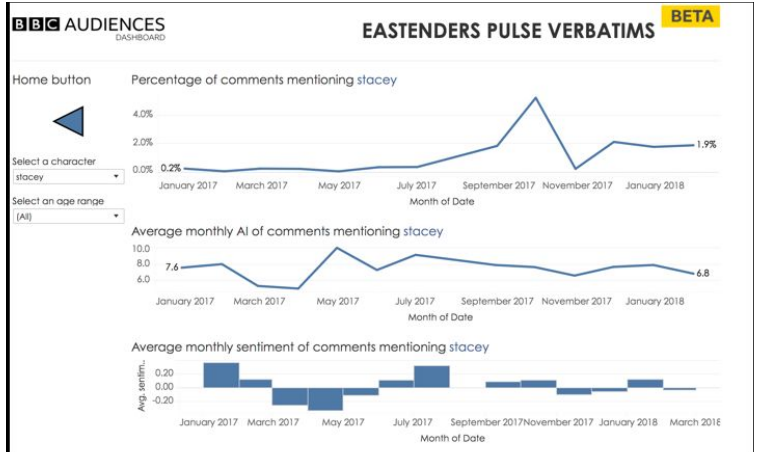
Percentage of comments containing words related to Love, Dislike or Neutral

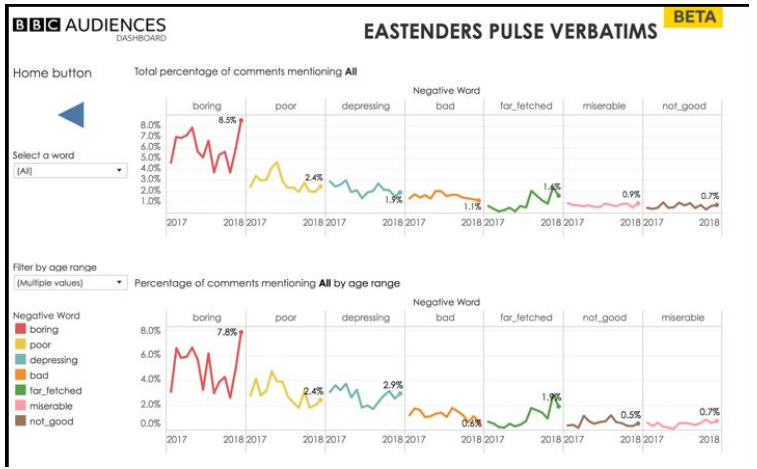
Click on the the coloured lines if you want to see more granular information

Comment searcher

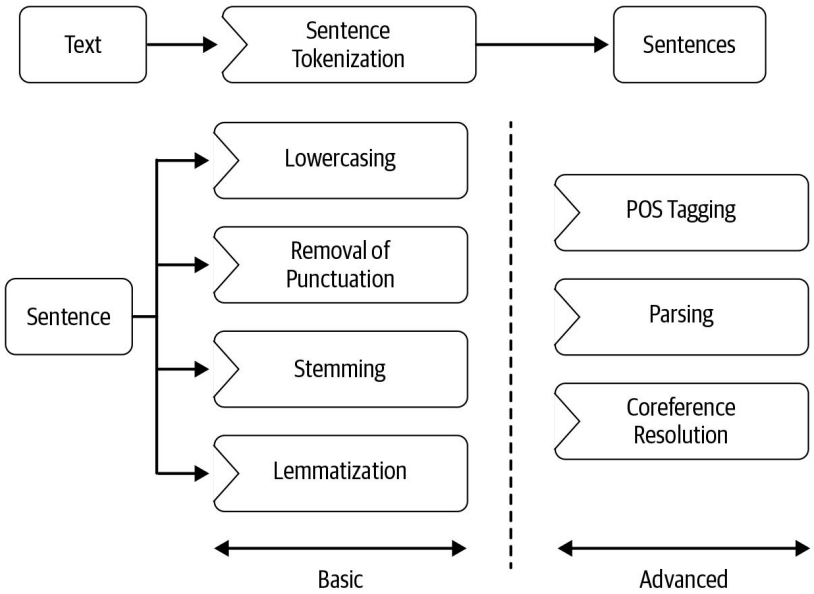
Click on a comment for advanced search window

Comment
it is getting ridiculous
what will max do, how will mick get through this one



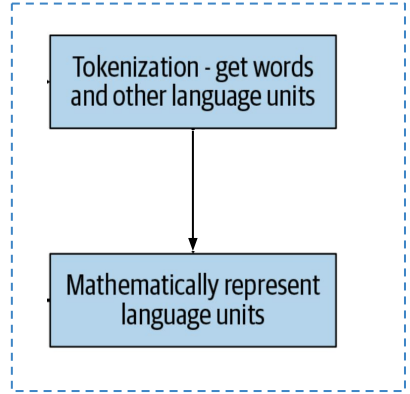


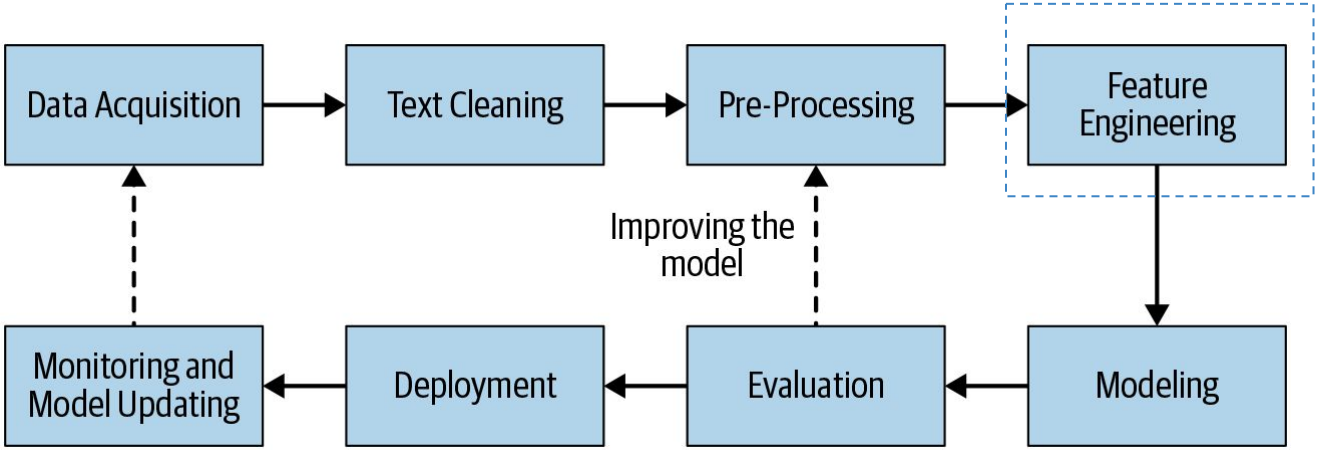
NLP Pipeline



NLP Pipeline

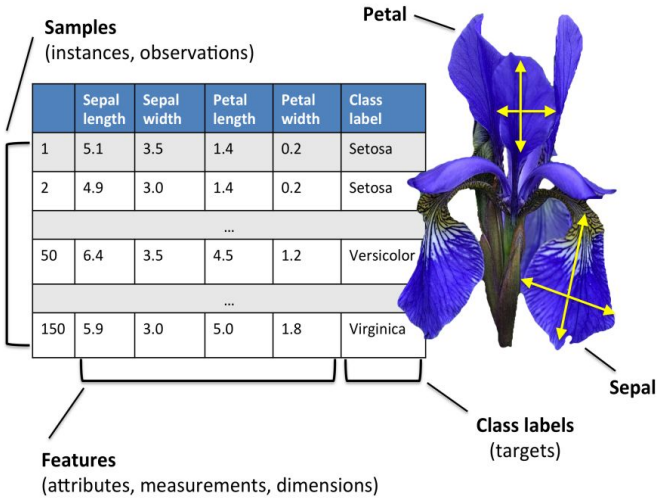
Feature engineering or text representation





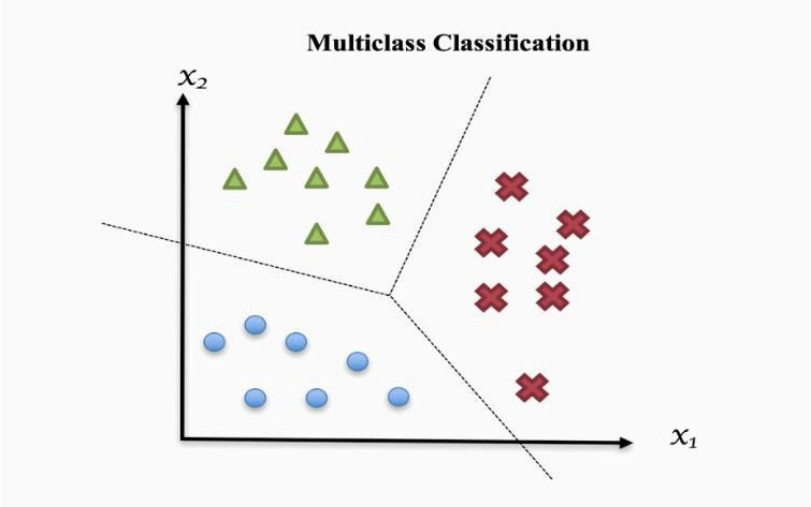
NLP Pipeline

Feature engineering or text representation

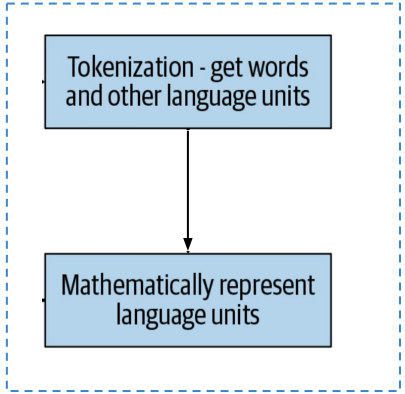


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Classical input dataset for a ML algorithm



How do we train a model with text?
How do we translate text into a mathematical representation?



Bag of words approach (BoW)

John likes to watch movies. Mary likes movies too.

Mary also likes to watch football games.


```
{"John":1,"likes":3,"to":2,"watch":2,"movies":2,"Mary":2,"too":1,"also":1,"football":1,"games":1}
```


Revisar
binary=true

	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

→ Vocabulary

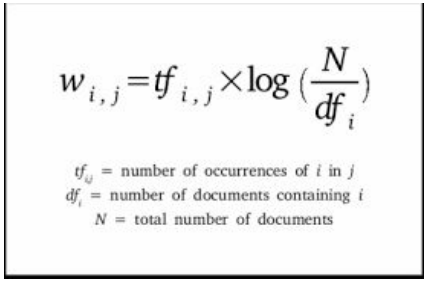
```
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer(binary=True)
bow_rep_bin = count_vect.fit_transform(processed_docs)
temp = count_vect.transform(["the red dog"])


print("Bow representation for 'the red dog':",
temp.toarray())
```

Hands on exercise...

How do we train a model with text?
How do we translate text into a mathematical representation?

Bag of words approach (BoW) and **TF-IDF**



- Vectors instead of binary values have real values
- Penalises very frequent words

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
bow_rep_tfidf = tfidf.fit_transform(processed_docs)
print(tfidf.idf_) #IDF for all words in the vocabulary
print(tfidf.get_feature_names()) #All words in the vocabulary.

temp = tfidf.transform(["dog and man are friends"])


print("Tfidf representation for 'dog and man are friends'\n",
temp.toarray())
```

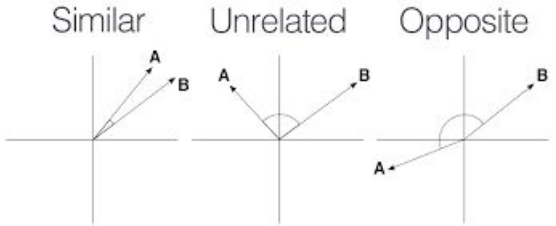
How do we train a model with text?
How do we translate text into a mathematical representation?

Bag of words approach (BoW) and **TF-IDF**

Vectorizing texts allows to make mathematical calculations with it

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Cosine similarity



How do we train a model with text?
How do we translate text into a mathematical representation?

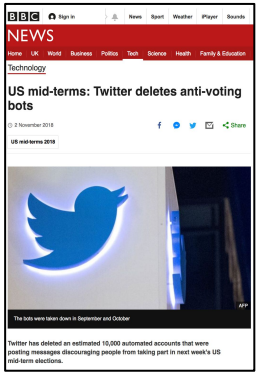
Bag of words approach (BoW) and **TF-IDF**

Vectorizing texts allows to make mathematical calculations with it

Database of articles

Article read
$(a_1,a_2a_3,a_4,..,a_n)$

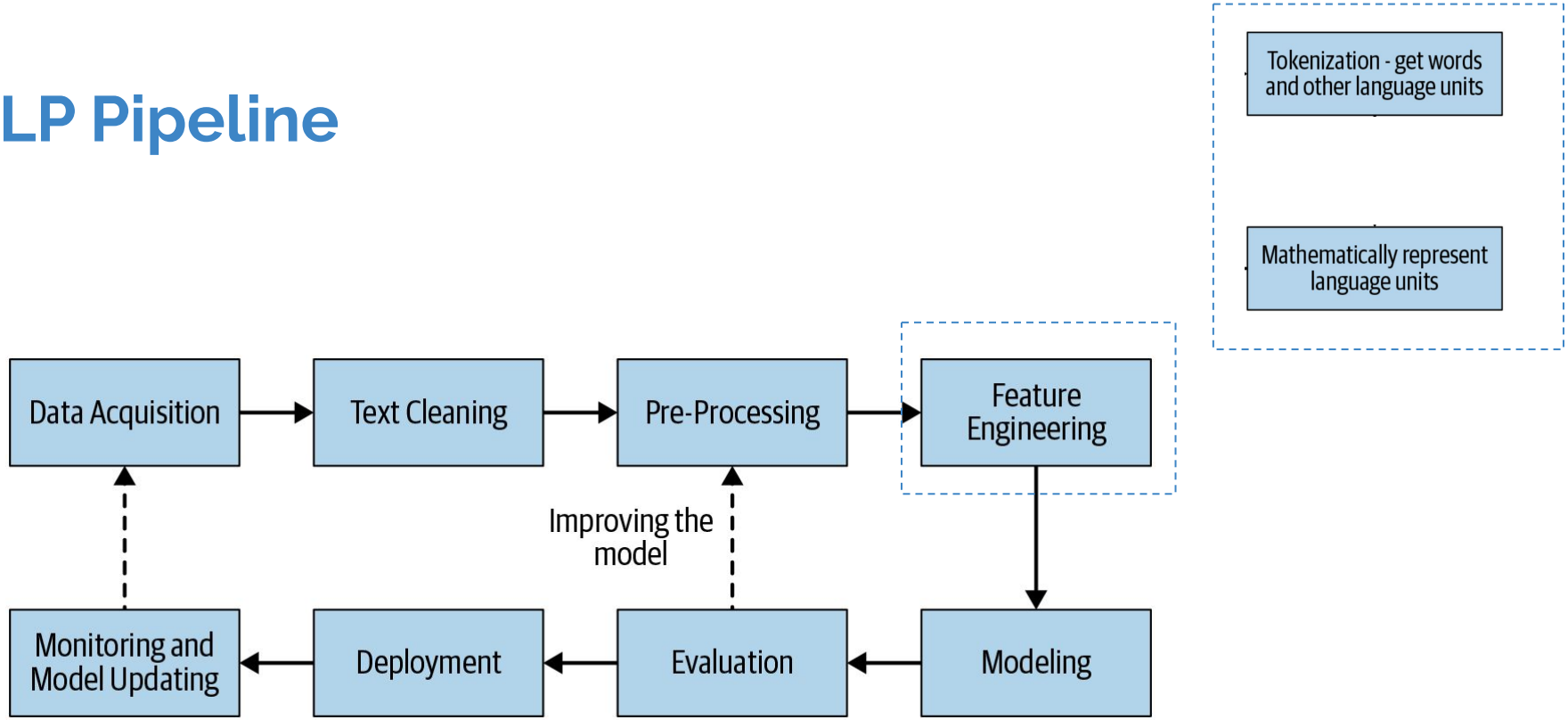
$(b_1,b_2b_3,b_4,..,b_n)$



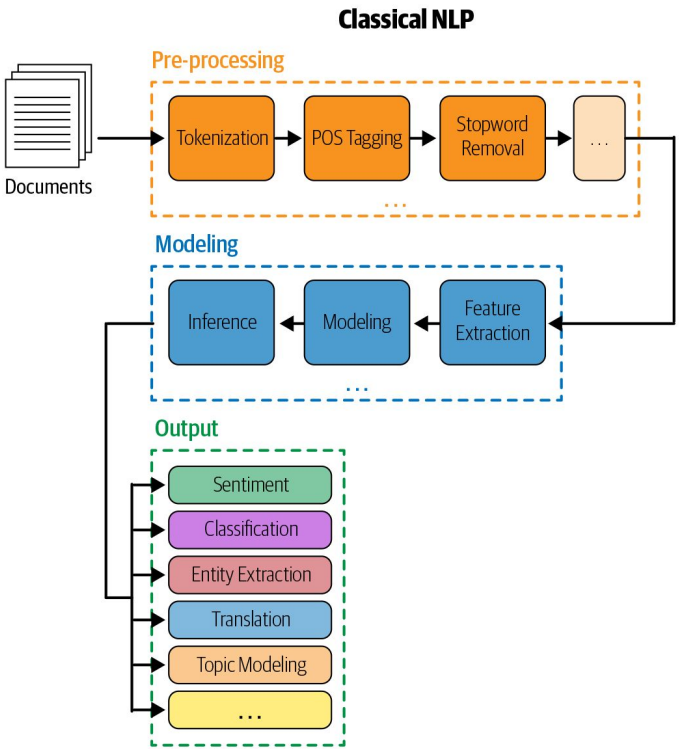
$(c_1,c_2c_3,c_4,..,c_n)$

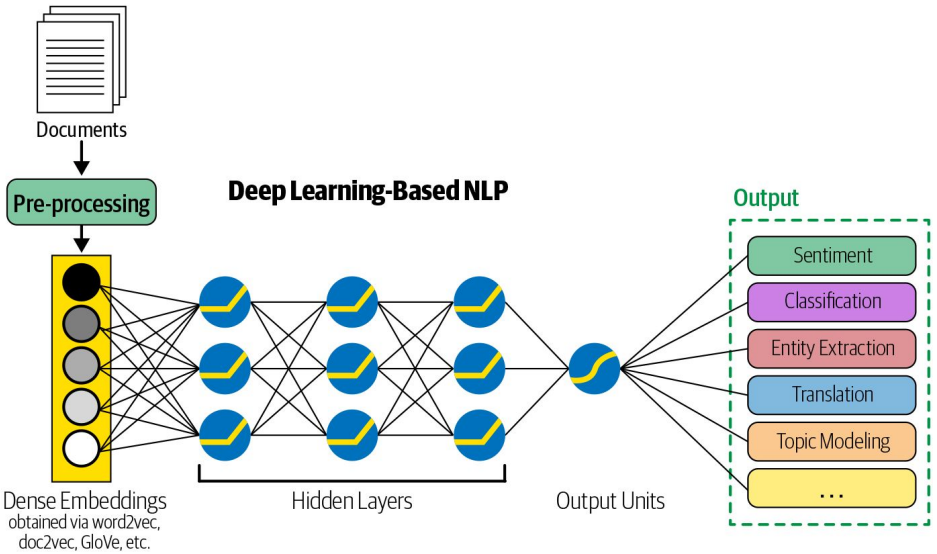
$(d_1,d_2d_3,d_4,..,d_n)$

NLP Pipeline



NLP Pipeline



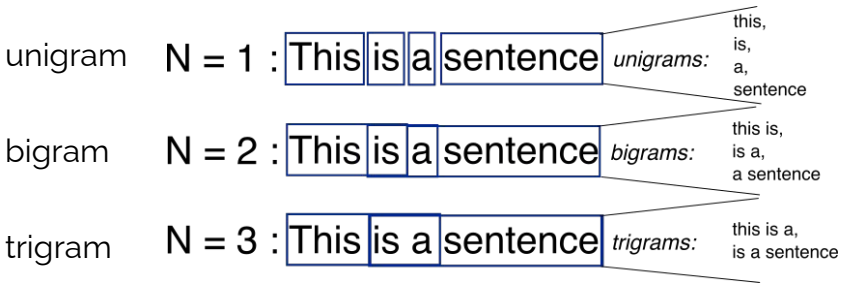


How do we train a model with text?
How do we translate text into a mathematical representation?

Bag of words approach (BoW)

+ Transforms text into vectors
+ Simplicity
- Ignores order and context
- Size of the vector increases with the size of the vocabulary
- Does not capture the similarity between words that mean the same (i.e. run and ran)

We can also use bigrams, trigrams,...



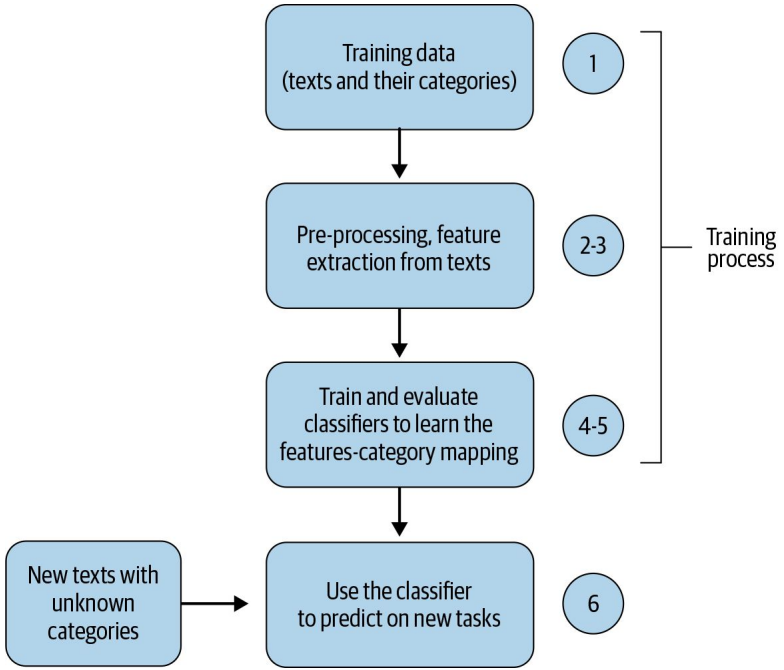

```
#n-gram vectorization example with count
vectorizer and uni, bi, trigrams

count_vect = CountVectorizer(ngram_range=(1,3))
```

NLP pipeline

Modeling

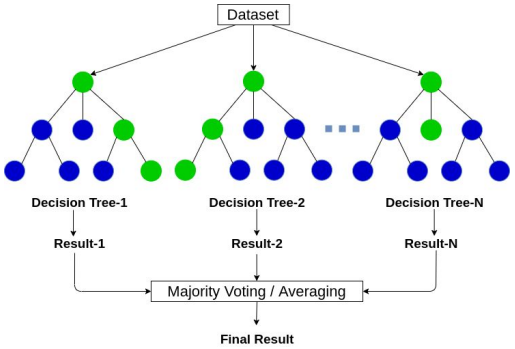
- **Types of problems**
  - Supervised learning problems
  - Unsupervised learning problems
  - Language generation problems
- Evaluation (metrics)
  - Supervised learning problems
  - Unsupervised learning problems
  - Generated language problems



NLP pipeline

Modeling

- **Types of problems**
  - Supervised learning problems - **classification**

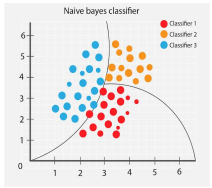


Random Forest

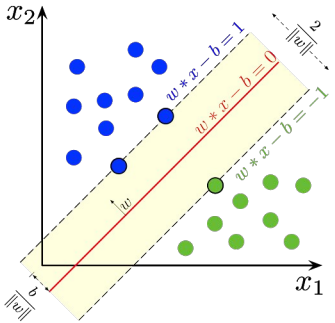
$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$



Naive Bayes



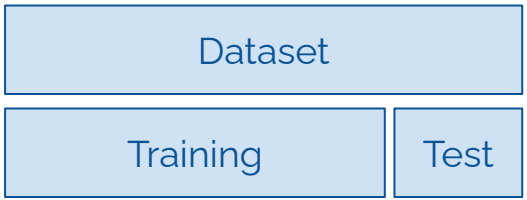
Support vector machines

Same models as in non-text problems

NLP Pipeline

Modeling

- **Evaluation**
  - Supervised learning problems



Regression	Classification
• Mean Absolute Error (MAE) • Root Mean Squared Error (RMSE) • R-Squared and Adjusted R-Squared	• Recall • Precision • F1-Score • Accuracy • Area Under the Curve (AUC)

Break 10 mins
Quiz (15 min, 10-15 mins review collaboratively)

Hands on exercise...

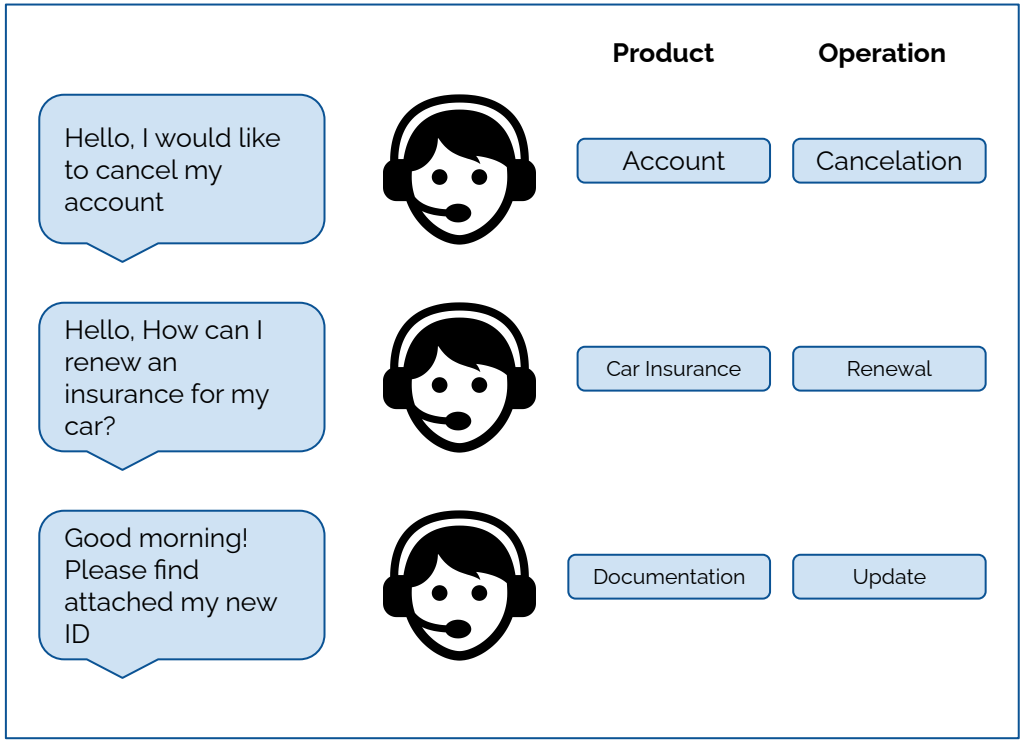
NLP Pipeline

Post-modeling phases

- Deployment
- Monitoring
- Model updating

Real use case

Conversations classifier



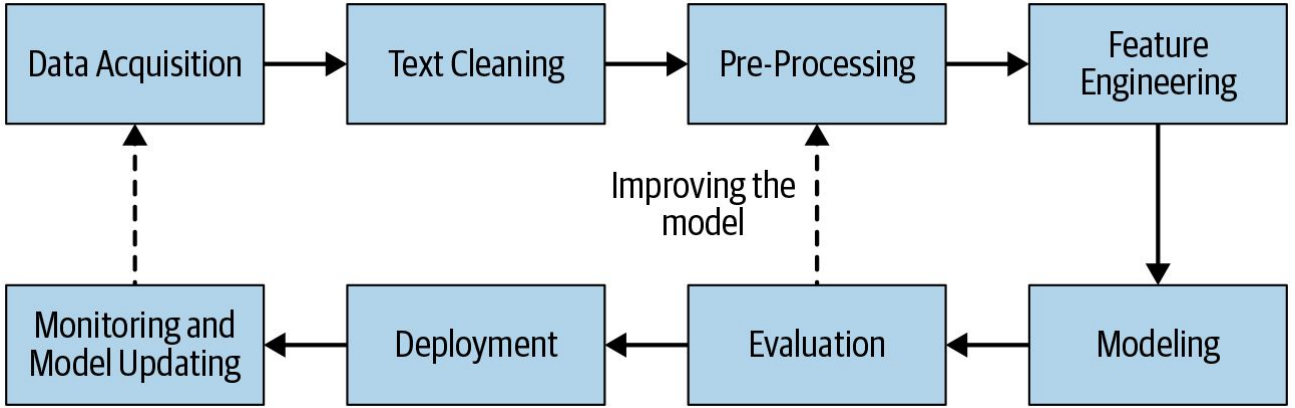
BBVA

Introduction to Natural language processing

Session 2

Clara Higuera Cabañes, PhD
Barcelona Technology School, May 2021

NLP pipeline



Index

1. Unsupervised learning in NLP
    a. Clustering
    b. Topic modeling
    c. Distance metrics
2. Hands on / live showcase nlp in action
3. Practical case
    a. Recommender system for BBC News with topic modeling
    b. Reading profile of users for segmentation
4. Word2vec - Embeddings
5. Deep learning architectures in nlp
6. Explainability in NLP
7. Bias in NLP

Explainability methods (after modeling)

- Eli5
- Lime

What happens with deep learning

Preprocessing
Text representation
- Embeddings

Unsupervised learning

- Topic modeling
- Content similarity recommender, cosine similarity and other distance metrics
- User reading profiles at BBC!
- Knowledge gaps study

Annotation

Snorkel [7, 8, 52]
This is a system for building training data automatically, without manual labeling. Using Snorkel, a large training dataset can be "created"—without manual labeling—using heuristics and creating synthetic data by transforming existing data and creating new data samples. This approach was shown to work well at Google in the recent past [9]

Active learning [13]
This is a specialized paradigm of ML where the learning algorithm can interactively query a data point and get its label. It is used in scenarios where there is an abundance of unlabeled data but manually labeling is expensive. In such cases, the question becomes: for which data points should we ask for labels to maximize learning while keeping the labeling cost low?

1. Intro

Choose one approach to grab the audience's attention right from the start: unexpected, emotional, or simple.

➔ **Unexpected**
Highlight what's new, unusual, or surprising.

➔ **Emotional**
Give people a reason to care.

➔ **Simple**
Provide a simple unifying message for what is to come

How many languages do you need to know to communicate with the rest of the world?

Tip

In this example, we're leading off with something **unexpected.**

While the audience is trying to come up with a number, we'll surprise them with the next slide.

Just one! Your own.

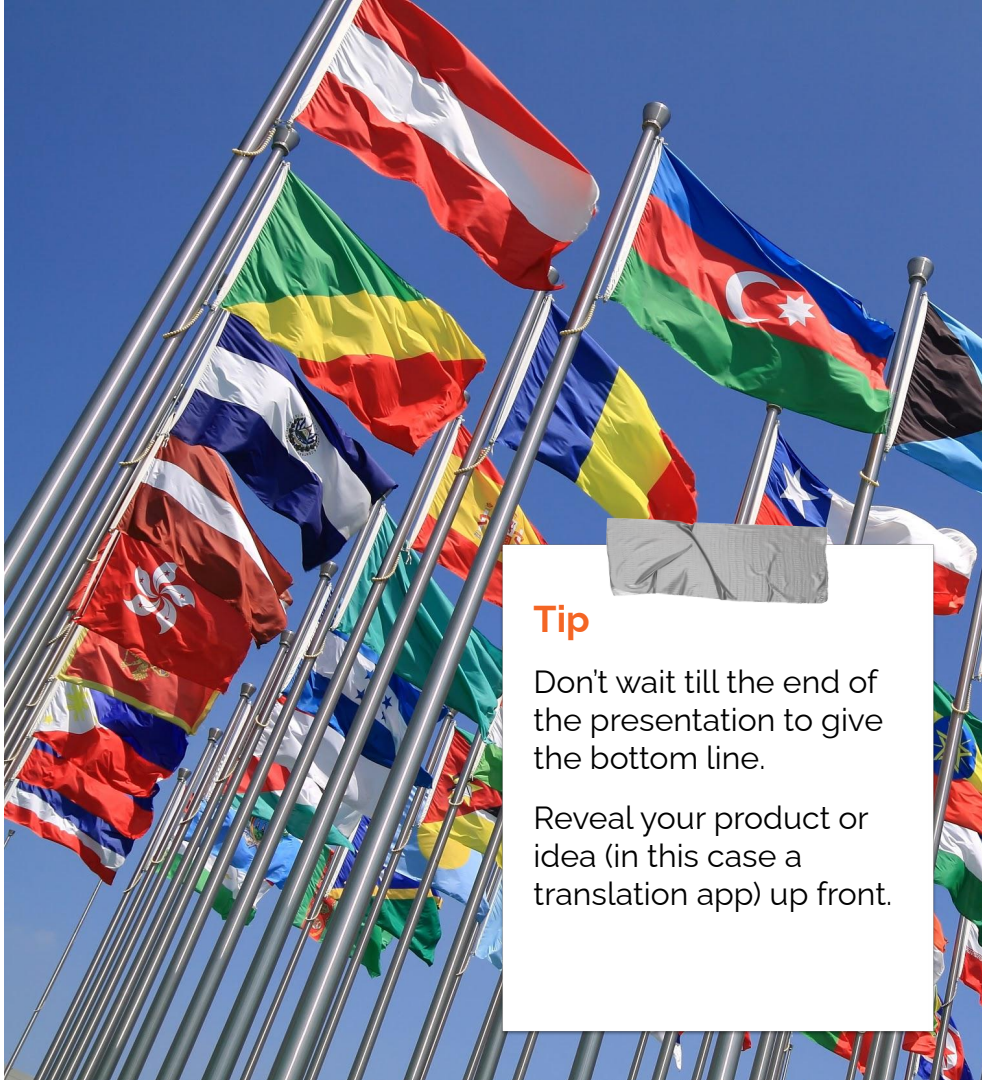
(With a little help from your smart phone)

Tip

Remember. If something sounds like common sense, people will ignore it.

Highlight what is unexpected about your topic.

The Google Translate app can repeat anything you say in up to **NINETY LANGUAGES** from German and Japanese to Czech and Zulu



Tip

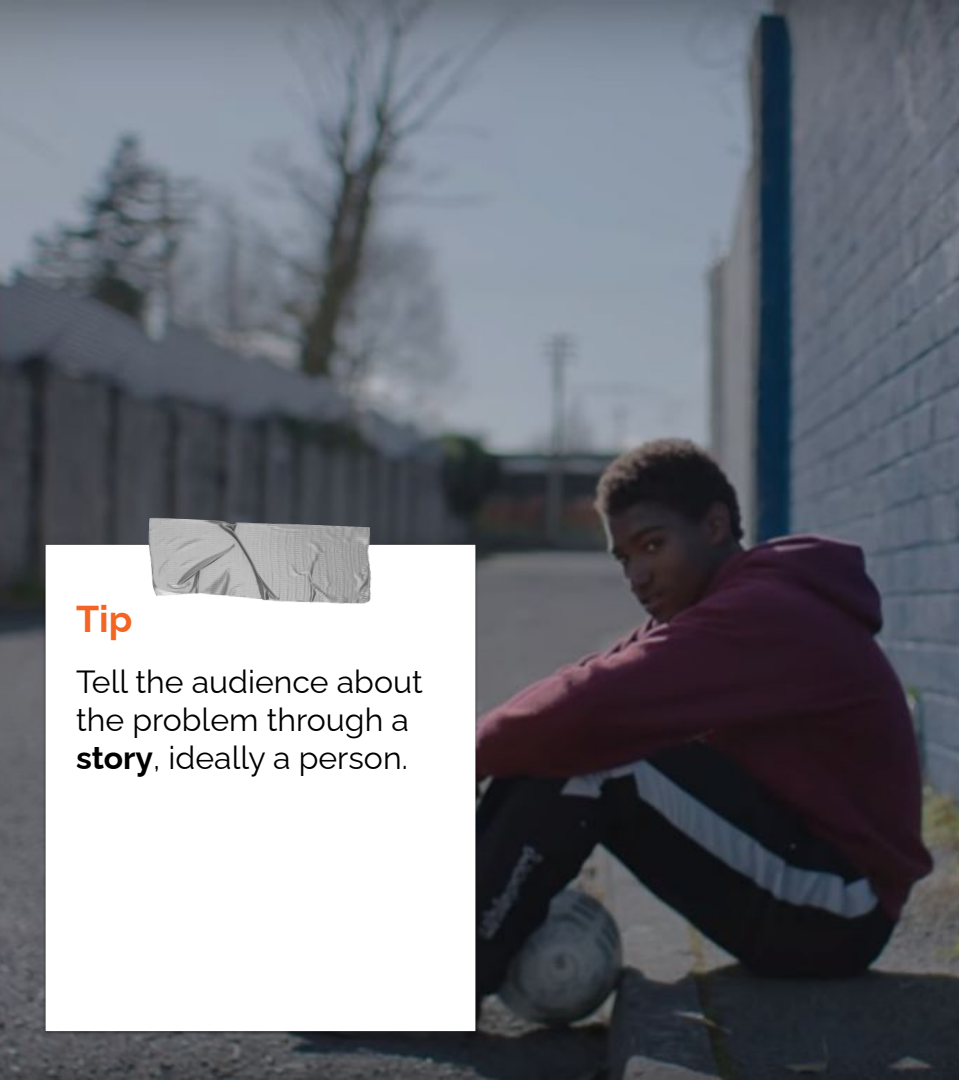
Don't wait till the end of the presentation to give the bottom line.

Reveal your product or idea (in this case a translation app) up front.

2. Examples

By the end of this section, your audience should be able to visualize:

- **What**
  What is the pain you cure with your solution?
- **Who**
  Show them a specific person who would benefit from your solution.



Tip

Tell the audience about the problem through a **story**, ideally a person.

Meet Alberto.

He recently moved from Spain to a small town in Northern Ireland.

He loved soccer, but feared he had no way to talk to a coach or teammates.

Meet Marcos.

He recently opened a camera shop near the Louvre in Paris.

Visitors to his store, mostly tourists, speak many different languages making anything beyond a simple transaction a challenge.

Story for illustration purposes only

Tip

If one example isn't sufficient to help people understand the breadth of your idea, pick a couple of examples.

A translation barrier left Alberto feeling lonely and hurt Marco's business.

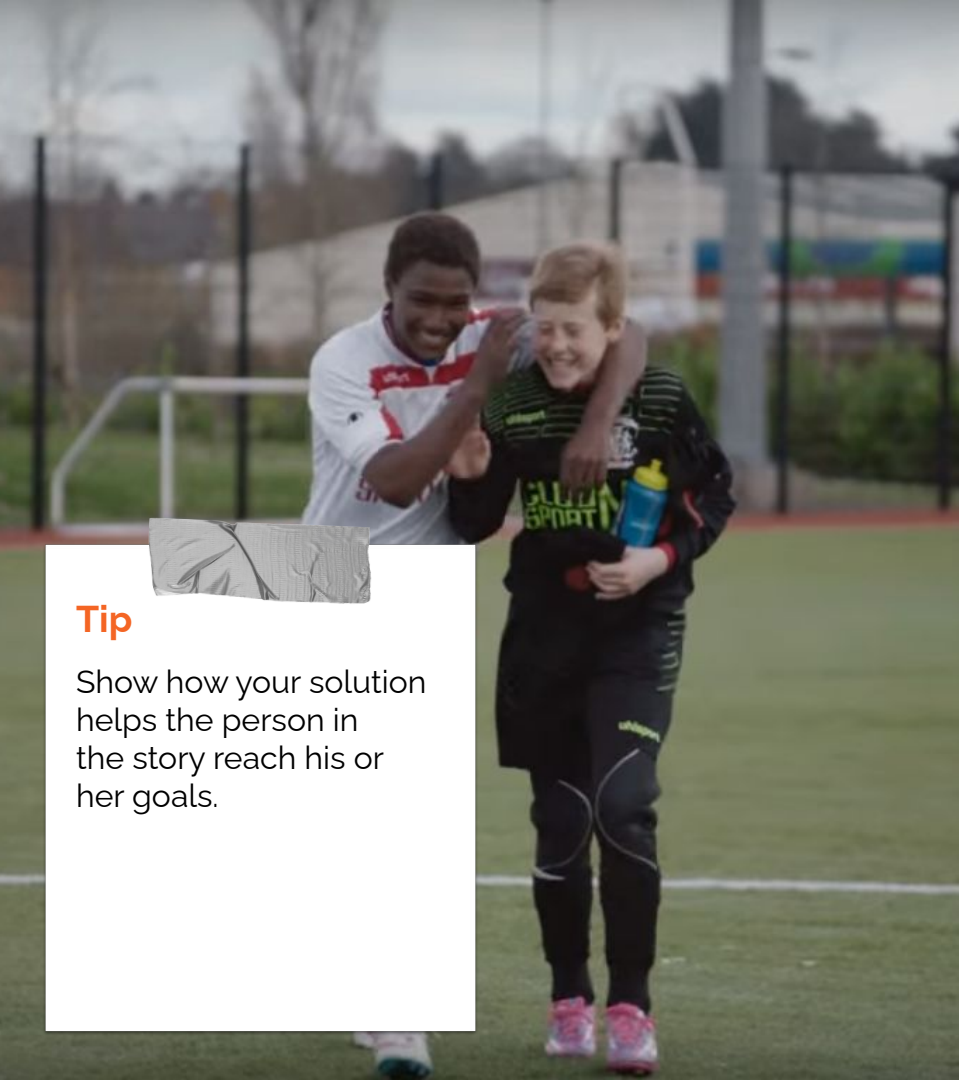
Tip

Ideally, speak of people in very different situations, but where each could benefit from your solution.

Then, Marcos discovered Google Translate

He has his visiting customers speak their camera issues into the app.

He’s able to give them a friendly, personalized experience by understanding exactly what they need.



Tip

Show how your solution helps the person in the story reach his or her goals.

A simple gesture

Coaches Gary and Glen knew no Spanish.

They used Google Translate to invite Alberto to join in... “Do you want to play?”... “Can you defend the left side?”

From outsider to star

Alberto scored 30 goals in 21 games. He is now being scouted by several professional clubs in the Premier League. And he's a favorite of the other boys on the team.

See a short video on Alberto's story

Tip

Stories become more credible when they use concrete details such as the specific complex moves Alberto learned through Translate and his 30 goals in 21 games performance stats.

3. Examples

People need to understand how rare or frequent your examples are.

Pick 1 or 2 statistics and make them as concrete as possible. Stats are generally not sticky, but here are a few tactics:

- **Relate**
  Deliver data within the context of a story you've already told
- **Compare**
  Make big numbers digestible by putting them in the context of something familiar

It's no surprise Marcos uses Google Translate in his shop regularly.

There are 23 officially recognized languages in the EU.

Source: theguardian.com

Tip

Don't let data stand alone. Always relate it back to a story you've already told, in this case, Marco's shop.

More than 50 million Americans travelled abroad in 2015

THAT'S MORE THAN THE POPULATION OF CALIFORNIA AND TEXAS COMBINED

Source: travel.trade.gov

Tip

When a number is too large or too small to easily comprehend, clarify it with a comparison to something familiar.

4. Closing

Build confidence around your product or idea by including at least one of the these slides:

- **Milestones**
  What has been accomplished and what might be left to tackle?
- **Testimonials**
  Who supports your idea (or doesn't)?
- **What's next?**
  How can the audience get involved or find out more?

What people are saying

With this app, I'm confident to plan a trip to rural Vietnam

Wendy Writer, CA

Visual translation feels like magic

Ronny Reader, NYC

Translate has officially inspired me to learn French

Abby Author, NYC

Quotes for illustration purposes only

Know a 2nd language? Make Google Translate even better by joining the community.

Tip

Inspire your audience to act on the information they just learned.

Depending on your idea, this can be anything from downloading an app to joining an organization.

Good luck!

We hope you'll use these tips to go out and deliver a memorable pitch for your product or service!

For more (free) presentation tips relevant to other types of messages, go to heathbrothers.com/presentations

For more about making your ideas stick with others, check out our book!