Lecture 7
Introduction to Text Analysis

Pierre Biscaye

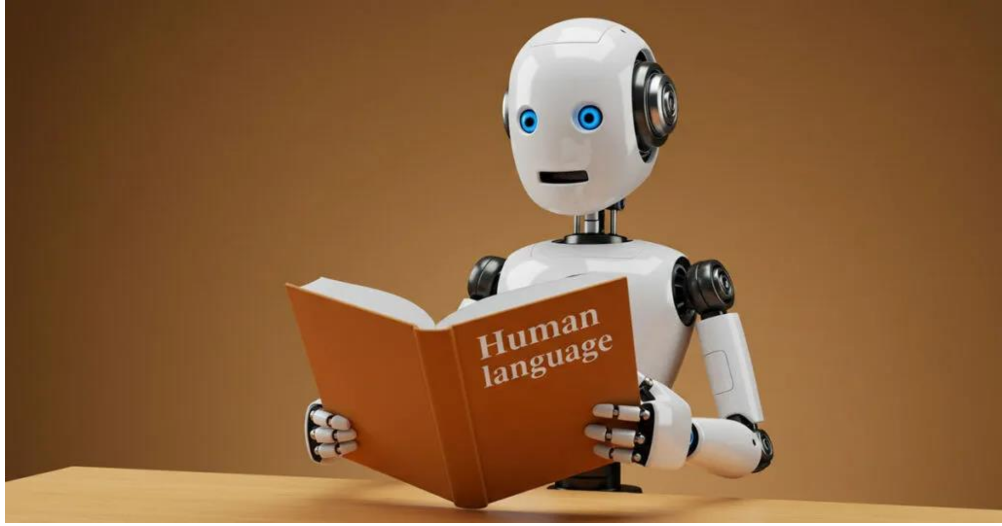
Université Clermont Auvergne

Data Science for Economics

Why text? The potential of unstructured data

- Motivation: Over 80% of the world’s data is unstructured text (emails, social media, news, medical records, legal docs).
- Opportunity: Traditional econometrics uses "hard" numbers (GDP, price, rainfall). NLP allows us to turn "soft" signals (public mood, political rhetoric, uncertainty) extracted from text into quantitative variables.
  - Natural Language Processing (NLP): The field of AI focused on enabling computers to understand, interpret, and generate human language.
- Examples: How does the sentiment of social media posts in a region correlate with measures of conflict or protest? Can we identify areas experiencing floods in a low-income country from local new coverage?

Human language is hard

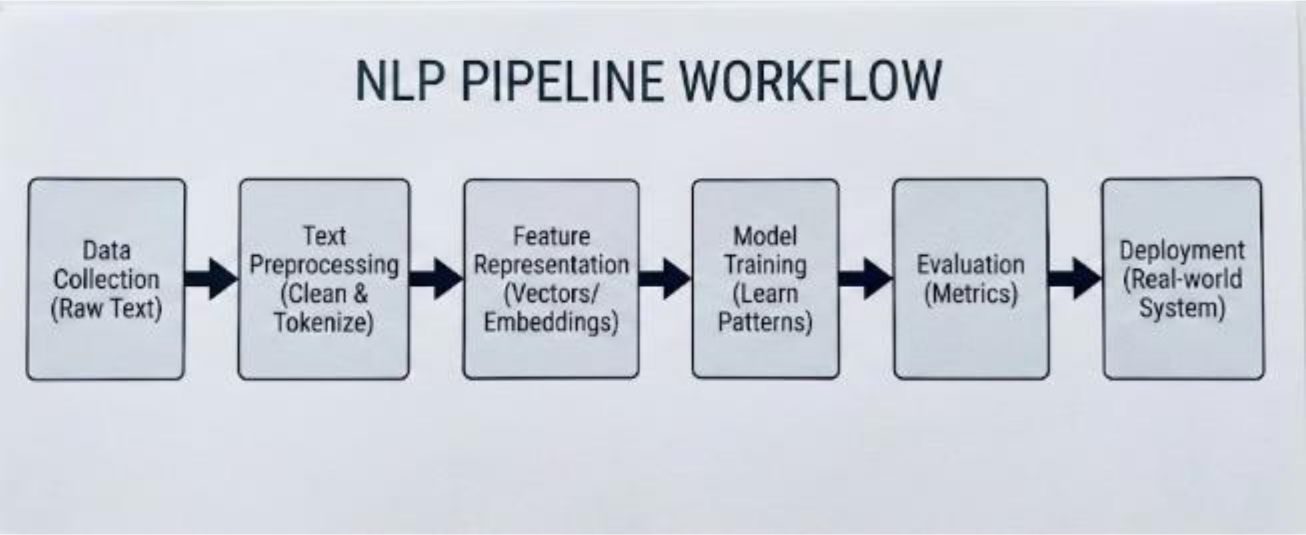


Source

- Ambiguity:
  - "The tank was full" (Military vs. Water)
  - "It's down by the bank" (Finance vs. Water)
- Context/Sarcasm: "Oh great, another delayed flight." (A positive word used for a negative sentiment).
- Sparsity: There are millions of unique words, but most appear only once or twice in a dataset.
- The Goal: We need to move from Words (Human logic) to Vectors (Machine logic)

High-level NLP Pipeline

- Preprocessing: Scrub the data clean.
- Tokenization: Break text into manageable "atoms.“
- Exploratory Data Analysis (EDA): Identify patterns, word counts, and distributions.
- Vectorization: Convert atoms into numbers (BoW, TF-IDF, Embeddings).
- Modeling: Predict sentiment or similarity using those numbers.


NLP PIPELINE WORKFLOW
Data Collection (Raw Text)
Text Preprocessing (Clean & Tokenize)
Feature Representation (Vectors/ Embeddings)
Model Training (Learn Patterns)
Evaluation (Metrics)
Deployment (Real-world System)

Source

Key terms/concepts

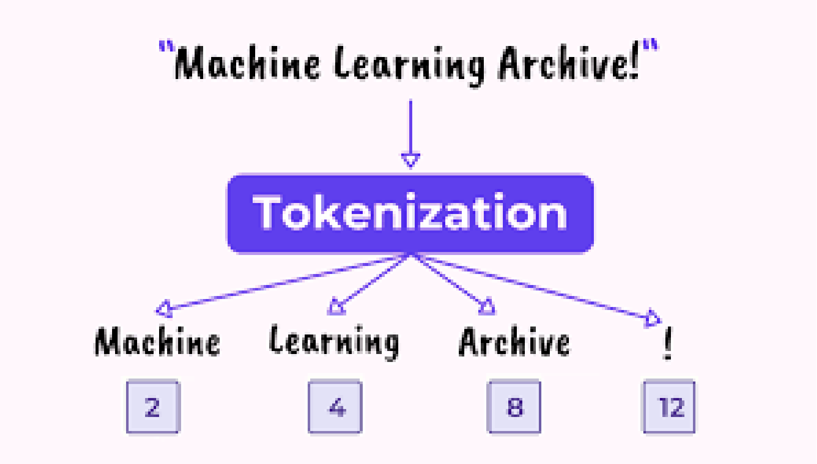
- **Corpus** (pl. Corpora): A large, structured collection of machine-readable texts.
- **Document**: Individual texts (e.g., a single Tweet in a collection).
- **Stemming**: Removing suffixes/prefixes to get word “stems”.
- **Token**: A single “unit” of text, usually a word, character, or subword.
- **Vocabulary**: All unique tokens in a corpus.
- **Embedding**: A dense, continuous numerical representation of a word where the position in space captures its semantic meaning (e.g., “dog” and “puppy” are close together).

Preprocessing text

- Standard Steps:
  - Lowercasing: "Apple" and "apple" should be the same.
  - Punctuation: Removing ! and ? (unless needed for sentiment).
  - Stop-words: Dropping frequent but low-info words (a, an, the, is).
- Normalization:
  - Stemming: Truncating words (e.g., "Universal," "University," "Universe" → "Univers").
  - Lemmatization: Finding the root dictionary form (e.g., "Better" → "Good").
- Corpus-Specific Issues:
  - E.g., Twitter: @usernames, #hashtags, and URLs.

Tokenization: breaking text into "chunks"

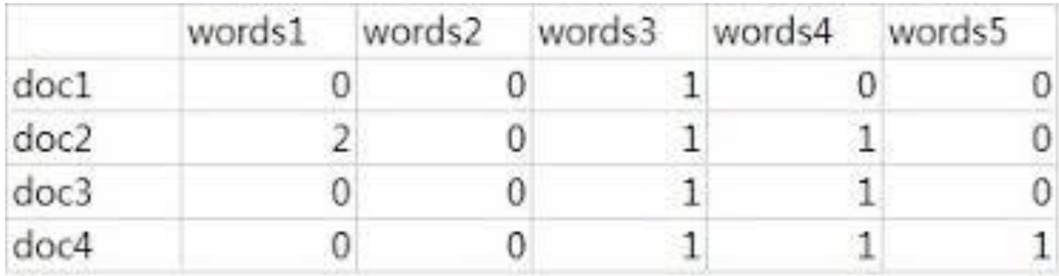
- Rule-Based (NLTK/spaCy): Splits by spaces or punctuation.
  - Simple, but fails on "don't" (is it one token or two?).
- Modern Subword Tokenization (BERT):
  - Breaks words into meaningful chunks: "unhappiness" → un, happi, ness.
  - The Benefit: Never encounter an "Unknown" word again. If the model hasn't seen a word, it just breaks it into smaller pieces it has seen.



Source

Bag-of-Words: Counting what matters

- The Concept: Ignore order, just count.
    - "I love Python" and "Python love I" look identical to the model.
- The Document-Term Matrix (DTM):
    - Each row is a document in your corpus (e.g., tweet, blog post, report).
    - Each column is a word in your entire vocabulary.
    - The cell value is the count of that word in that tweet.
    - Issue: Long documents get higher scores just for being long.



	words1	words2	words3	words4	words5
doc1	0	0	1	0	0
doc2	2	0	1	1	0
doc3	0	0	1	1	0
doc4	0	0	1	1	1

Source

Scaling the signal

- Term Frequency (TF): Does this word appear often here?
- Inverse Document Frequency (IDF): Is this word rare across the whole dataset?
- The Logic: If "inflation" appears 5 times in one tweet, but it appears in every tweet in your dataset, it’s not a good signal.
- TF-IDF penalizes common words and rewards specific ones.

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

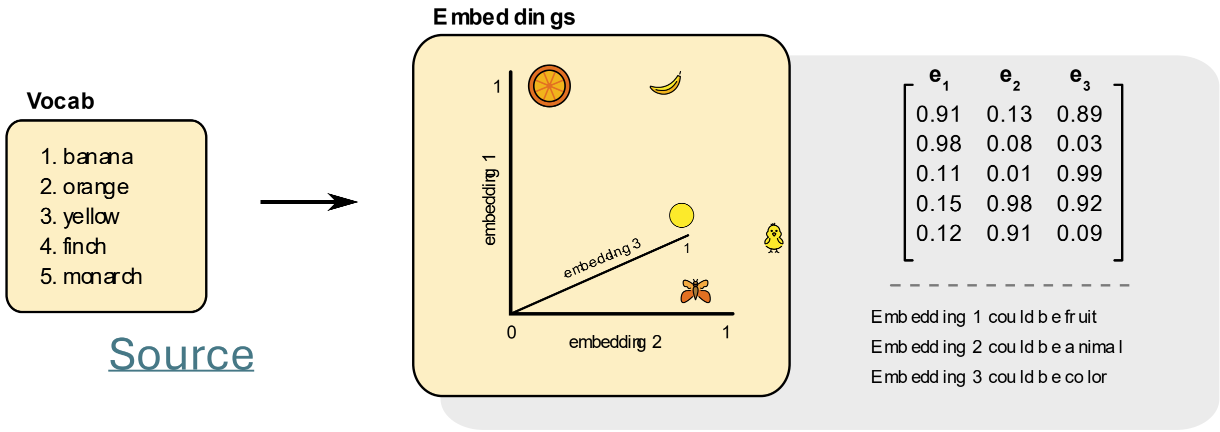
$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

Source

Measuring meaning

- The Shift: Bag-of-Words treats words as isolated islands. Embeddings treat them as coordinates in space.
- Word2Vec/GloVe: "You shall know a word by the company it keeps.“
- Intuition: Words that appear in similar contexts (e.g., "Coffee" and "Tea") will have similar coordinates. We can measure the Cosine Similarity (angle) between these vectors.



Vocab

1. banana
2. orange
3. yellow
4. finch
5. monarch

Embeddings

$$\begin{array}{ccc} e_1 & e_2 & e_3 \\ 0.91 & 0.13 & 0.89 \\ 0.98 & 0.08 & 0.03 \\ 0.11 & 0.01 & 0.99 \\ 0.15 & 0.98 & 0.92 \\ 0.12 & 0.91 & 0.09 \end{array}$$

Embedding 1 could be fruit

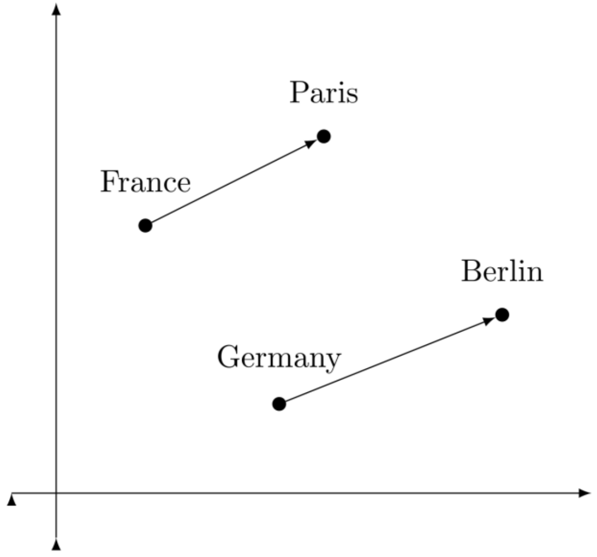
Embedding 2 could be animal

Embedding 3 could be color

Source

Semantic axes and associations

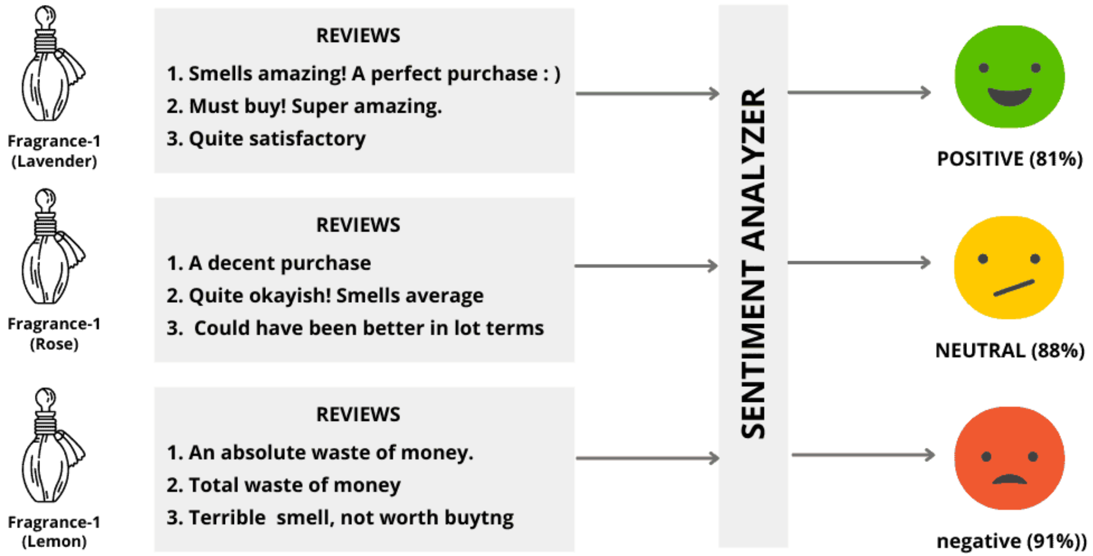
- Word Algebra: After embedding words as vectors in “vocabulary space”, we can perform math on them.
  - Example: Paris – France + Germany = Berlin
- Associations: Project words onto an axis that captures a particular meaning to see where they fall.
  - Example: Bias Detection: By projecting words onto a "Gender Axis" (Man → Woman), we can see if certain occupations or traits cluster closer to one end.



Source

Example application: Sentiment prediction

- Take your vectors (TF-IDF or Embeddings) and feed them into an ML classifier.
- Assign labels: Positive, Negative, or Neutral.
- Validation: How well does the model perform against human-labeled "Gold Standard" data?



REVIEWS

1. Smells amazing! A perfect purchase :)
2. Must buy! Super amazing.
3. Quite satisfactory

REVIEWS

1. A decent purchase
2. Quite okayish! Smells average
3. Could have been better in lot terms

REVIEWS

1. An absolute waste of money.
2. Total waste of money
3. Terrible smell, not worth buytng

Source