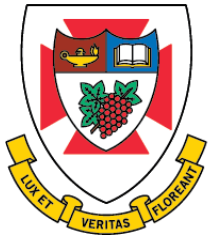
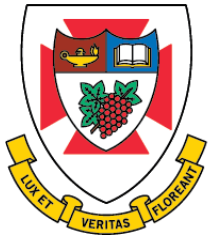


THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

INTRODUCTION TO MACHINE LEARNING

Working with Text Data



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

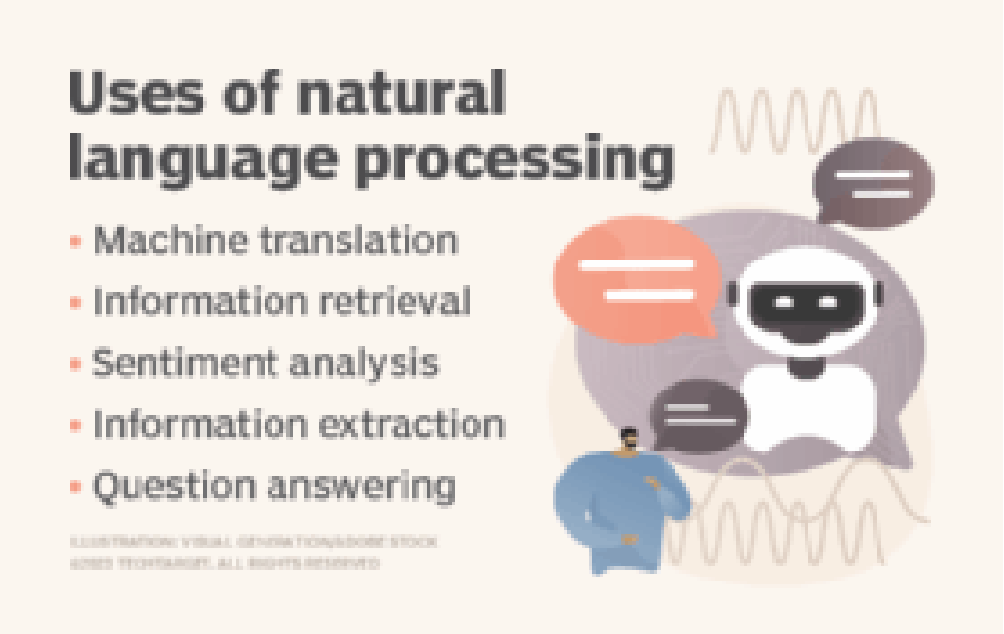
Natural Language Processing (NLP)
A Gentle Introduction

What is NLP?

- Natural language processing (NLP) is a component of artificial intelligence (AI)
- It is the ability of a computer program to understand human language as it's spoken and written -- referred to as natural language.

Why is NLP Important?

- Businesses use large amounts of unstructured, text-heavy data and need a way to efficiently process it.
- Much of the information created online and stored in databases is natural human language, and until recently, businesses couldn't effectively analyze this data.
- This is where natural language processing is useful.

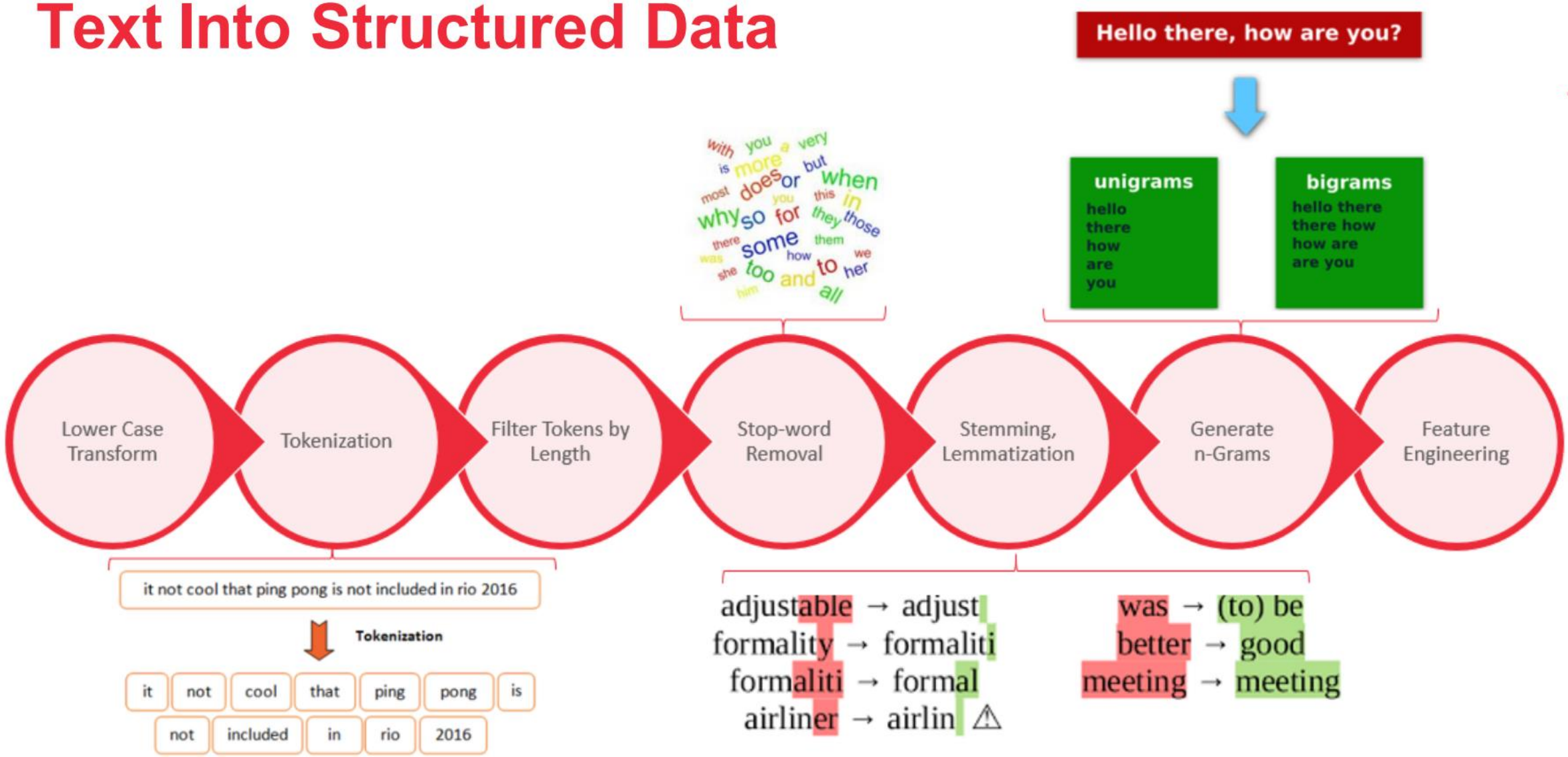


https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP

Working with Text Data

- A problem with modeling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed-length inputs and outputs.
- Machine learning algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers.
- In language processing, the vectors X are derived from textual data, in order to reflect various linguistic properties of the text.
- This is called feature extraction / feature encoding / feature engineering.

Text Into Structured Data



Feature Extraction

- Bag of Words
- Scoring Words
  - Count
  - Frequency
- TF-IDF

Bag of Words (BOW)

- A way of extracting features from text for use in modeling with machine learning algorithms.
- The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents.
- A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:
  1. A vocabulary of known words.
  2. A measure of the presence of known words.

BOW Model

- Step 1: Collect Data
- Step 2: Design the Vocabulary
- Step 3: Create Document Vectors

BOW Model Example

Step 1: Collect Data

- Below is a snippet of the first few lines of text from the book “A Tale of Two Cities” by Charles Dickens, taken from Project Gutenberg.

  - It was the best of times,
  - it was the worst of times,
  - it was the age of wisdom,
  - it was the age of foolishness,

- For this small example, let’s treat each line as a separate “document” and the 4 lines as our entire corpus of documents.

BOW Model Example

Step 2: Design the Vocabulary

- A list of all of the words in our model vocabulary.

- The unique words here (ignoring case and punctuation) are:

  “it” “was” “the” “best” “of” “times”

  “worst” “age” “wisdom” “foolishness”

- That is a vocabulary of 10 words from a corpus containing 24 words.

BOW Model Example

Step 3: Create Document Vectors

- The next step is to score the words in each document.

1. “it was the best of times” = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
2. "it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
3. "it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
4. "it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

Term Frequency – Inverse Document Frequency

- TF-IDF is an approach to rescale the frequency of words by how often they appear in all documents
  - not all words are equally important
  - frequent words across all documents are penalized
  - highlight words that are distinct (contain useful information) in a given document

- Two steps in TF-IDF
  - **Term Frequency**: is a scoring of the frequency of the word in the current document.
  - **Inverse Document Frequency**: is a scoring of how rare the word is across documents.

TF-IDF

- Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval.
- It measures how important a term is within a document relative to a collection of documents (also known as corpus).
- As its name implies, TF-IDF vectorizes/scores a word by multiplying the word's Term Frequency (TF) with the Inverse Document Frequency (IDF).

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$
$$= \text{TF} \times \log(N/DF)$$

Where,

TF is the term frequency of a word in a document, N is the total number of documents in the corpus, and DF is the number of documents that contain the word

https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/

Motivation

- TF-IDF basically measures how relevant a word is with respect to the document that contains it, and the collection of documents.
- Term frequency (TF) is how often a word appears in a document, divided by how many words there are.
- Inverse document frequency (IDF) is how unique or rare a word is.
  - If a word appears in many documents, its IDF score will be lower, (*less unique).*
  - If a word appears in fewer documents, its IDF score will be higher (*more unique).*
- A High TF-IDF score is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents.

- https://medium.com/towards-data-science/text-summarization-using-tf-idf-e64a0644ace3

Example

- Given a corpus of five documents:

  Doc1: The quick brown fox jumps over the lazy dog
  Doc2: The lazy dog likes to sleep all day
  Doc3: The brown fox prefers to eat cheese
  Doc4: The red fox jumps over the brown fox
  Doc5: The brown dog chases the fox

- Step 1: Term Frequency (TF) of the word “fox”

  TF = (Number of times word appears in the document) / (Total number of words in the document)
  Doc1: 1 / 9
  Doc2: 0 / 8
  Doc3: 1 / 7
  Doc4: 2 / 8
  Doc5: 1 / 6

- Step 2: Document Frequency (DF)

  DF = 4 (Doc1, Doc3, Doc4 and Doc5)

https://medium.com/@er.iit.pradeep09/understanding-tf-idf-in-nlp-a-comprehensive-guide-26707db0cec5

Example contd.

- Step 3: Inverse Document Frequency (IDF)

  IDF = ln(5/4) = 0.2231

- Step 4: TF-IDF Score for the word “fox”

  TF-IDF = TF * IDF
  Doc1: 1/9 * 0.2231 = 0.0247
  Doc2: 0/8 * 0.2231 = 0
  Doc3: 1/7 * 0.2231 = 0.0318
  Doc4: 2/8 * 0.2231 = 0.0557
  Doc5: 1/6 * 0.2231 = 0.0372

- TF-IDF score for the word “fox” is highest in Doc4 indicating that this word is relatively important in this document compared to the rest of the corpus.
- TF-IDF score is zero in Doc2, indicating that the word “fox” is not relevant in this document

https://medium.com/@er.iit.pradeep09/understanding-tf-idf-in-nlp-a-comprehensive-guide-26707db0cec5

BOW Model in Python scikit-learn

- https://www.kaggle.com/code/vipulgandhi/bag-of-words-model-for-beginners
- https://scikit-learn.org/stable/modules/feature_extraction.html

TF-IDF in Python scikit-learn

- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html