AI/ML Accelerator

Natural Language Processing

Session 4 - Week 1

Learning Outcomes

- Practical knowledge of natural language processing (NLP) specific model training and applications
- Be comfortable talking with NLP Terminology

Use Case Summary

For this course, we'll apply sentiment analysis to product reviews from a travel accessories retailer, Oceanwave15 Retails. By analyzing customer reviews, we aim to identify patterns in customer satisfaction, highlight popular products, and detect potential issues with product quality or usability. This analysis can guide strategic decisions in marketing, product improvement, and customer support.

Review	Sentiment
1. "This water bottle keeps drinks cold for hours and is perfect for long hikes."	Positive
2. "The suitcase is a bit heavy and not very easy to carry, especially when full."	Negative
3. "The seat cushion was comfortable but didn't offer much back support over long trips."	Neutral

Question 1

How to Solve this Problem in Least Effort in Scalable of Million Review Records in Secured Environment - Cost Efficiently - Accuracy doesn’t matter a lot

a **Open AI 4o ChatGPT**

b **Gemini**

c **Open AI 3.5 ChatGPT**

d **Simple NLP Technique**

NLP Sessions Overview - Week 1

Day 1 - BootCamp

- Introduction to NLP and Applications Text Preprocessing (Stemming, Stop Words)
- Basic Text Representations (BoW, TF-IDF)
- N-Gram Language Modeling and Probabilities
- Hands-On Demo: Building Bigram/Trigram Model
- Tokenization and Embedding

Day 2 - Office Hours

- Recap and Discussion of NLP Basics & N-Gram Modeling
- Review Kaggle Exercises

Machine Learning with Text Data

ML models need **well-defined numerical data.**

Text data	Text preprocessing (Cleaning and formatting)	Vectorization (Convert to numbers)	Train ML Model using numerical data
	Stop words removal, Stemming, Lemmatization	Bag of Words	K Nearest Neighbors (KNN), Neural Network, etc.

Stop Words Removal

What is Stop Words Removal in NLP?

- Stop words are common words in a language (e.g., "and," "the," "is") that:
- Appear frequently in text but add little meaning.
- Contribute minimal value to NLP tasks like classification or topic modeling.
- Are removed to reduce noise and improve processing efficiency.

Why Do We Need Stop Words Removal?

- Reduces Dimensionality: By filtering out common words, the model processes a smaller, more relevant set of words.
- Enhances Model Performance: Focusing on significant words can lead to better results in tasks like text classification.
- Speeds Up Processing: Fewer words mean quicker analysis and computation.
- Customizable: Stop words can be tailored for specific tasks to include or exclude certain words based on the context.

Quick Demo -1

- Stop Word Removal

Stemming

What is Stop Words Removal in NLP?

- Stemming is the process of reducing words to their base or root form by removing suffixes and prefixes. The root form obtained may not always be a valid word but serves as a representation of the original word group.

- Original: ["cats", "running", "jumping", "happily"]
- Stemmed: ["cat", "run", "jump", "happi"]

Use Cases:

- Search Engines: Stemming helps match queries with relevant documents by matching different forms of a word.
- Text Classification: It reduces vocabulary size and helps algorithms identify related terms as the same.
- Sentiment Analysis: Ensures consistency when analyzing words like "happy," "happily," and "happiest."

Challenges

- **Over-Stemming:** This occurs when words are reduced too much, resulting in unrelated terms being matched (e.g., "universe" and "university" could be reduced to "univers").
- **Under-Stemming:** This happens when words are not sufficiently reduced, and similar words remain separate.

Quick Demo -2

- Stemming

Bag of Words (BoW)

Simple example using **word counts**:

	a	cat	dog	is	it	my	not	old	wolf
“It is a dog.”	1	0	1	1	1	0	0	0	0
“my cat is old”	0	1	0	1	0	1	0	1	0
“It is not a dog, it a is wolf.”	2	0	1	2	2	0	1	0	1

Term Frequency (TF)

Term frequency (TF): Increases the weight for **common** words in a document.

=

	a	cat	dog	is	it	my	not	old	wolf
“It is a dog.”	**0.25**	0	**0.25**	**0.25**	**0.25**	0	0	0	0
“my cat is old”	0	**0.25**	0	**0.25**	0	**0.25**	0	**0.25**	0
“It is not a dog, it a is wolf.”	**0.22**	0	**0.11**	**0.22**	**0.22**	0	**0.11**	0	**0.11**

Inverse Document Frequency (IDF)

term	idf
a	log(3/3)+1=**1**
cat	log(3/2)+1=**1.18**
dog	log(3/3)+1=**1**
is	log(3/4)+1=**0.87**
it	log(3/3)+1=**1**
my	log(3/2)+1=**1.18**
not	log(3/2)+1=**1.18**
old	log(3/2)+1=**1.18**
wolf	log(3/2)+1=**1.18**

Inverse document frequency (IDF): Decreases the weights for **commonly** used words and **increases** weights for **rare** words in the vocabulary.

Term Freq.-Inverse Doc. Freq (TF-IDF)

Term Freq. Inverse Doc. Freq (TF-IDF): Combines term frequency and inverse document frequency.

	a	cat	dog	is	it	my	not	old	wolf
“It is a dog.”	**0.25**	0	**0.25**	**0.22**	**0.25**	0	0	0	0
“my cat is old”	0	**0.3**	0	**0.22**	0	**0.3**	0	**0.3**	0
“It is not a dog, it a is wolf.”	**0.22**	0	**0.11**	**0.19**	**0.22**	0	**0.13**	0	**0.13**

N-gram

- An n-gram is a sequence of n tokens from a given sample of text or speech.
- We can include n-grams in our term frequencies too.

Sentence	1-gram (uni-gram):	2-gram (bi-gram):
It is not a dog, it is a wolf	“it”, “is”, “not”, “a”, “dog”, “it”, “is”, “a”, “wolf”	“it is”, “is not”, “not a”, “a dog”, “dog it”, “it is”, “is a”, “a wolf”

Bag-of-Words (BoW) and Term Freq.-Inverse Doc. Freq (TF-IDF)

	Bag of Words (BoW)	TF-IDF (Term Frequency-Inverse Document Frequency)
Definition	Counts occurrences of each word in a document, creating a sparse matrix.	Weighs words based on frequency in a document and uniqueness across documents.
Purpose	Captures word presence and frequency, useful for basic text representation.	Emphasizes unique words, reducing the impact of commonly used ones.
Use Case	**Sentiment Analysis:** Determines positivity/negativity by word frequency. **Topic Modeling:** Finds common themes by identifying frequent words.	**Document Similarity:** Measures similarity between documents based on unique terms. **Information Retrieval:** Helps match queries with relevant documents based on term importance.
Calculation	Counts number of times each word appears (frequency).	Calculates TF-IDF weight as TF (Term Frequency) x IDF (Inverse Document Frequency).

Bag-of-Words (BoW) and Term Freq.-Inverse Doc. Freq (TF-IDF)

	Bag of Words (BoW)	TF-IDF (Term Frequency-Inverse Document Frequency)
Handling Common Words	Cannot differentiate importance of common vs. unique words; frequent words dominate.	Reduces weight of common words, giving unique terms greater importance.
Challenges	• Leads to high-dimensional, sparse matrix with many zeros. • High memory usage, especially for large datasets. • Often overemphasizes common words without distinguishing relevance.	• Can ignore context by isolating term importance per document. • Sensitive to dataset size; rare words may be underweighted. • High computational cost for calculating IDF on large corpora.
Benefits	• Simple and easy to implement. • Effective for tasks where high frequency signals importance.	• Provides a balanced representation by highlighting unique terms. • Improves accuracy in tasks requiring term specificity.

Quick Demo - 3

- Bag of Words
- TF-IDF

Some NLP Terms

- **Corpus**: Large collection of words or phrases - can come from different sources: documents, web sources, database
  - Common Crawl Corpus: web crawl data composed of over 5 billion web pages (541 TB)
  - Reddit Submission Corpus: publicly available Reddit submissions (42 GB)
  - Wikipedia XML Data: complete copy of all Wikimedia wikis, in the form of wikitext source and metadata embedded in XML. (500 GB)
  - **en_core_web_md** is a medium-sized pre-trained NLP model provided by spaCy, offering robust word vectors, tokenization, and linguistic features for tasks such as named entity recognition, part-of-speech tagging, and word embeddings.

Tokenization

What

Tokenization is the process of breaking down a larger body of text into smaller units called tokens. These tokens can be words, subwords, or sentences, serving as the basic building blocks for further text analysis

Types of Tokenization:

- Word Tokenization:Breaks down text into individual words.

  Example: “The travel backpack is spacious.” → ['The', 'travel', 'backpack', 'is', 'spacious']
- Sentence Tokenization: Splits text into sentences.

  Example: “The travel backpack is spacious. It fits all my essentials.” → ['The travel backpack is spacious.', 'It fits all my essentials.']

Purpose

- Text Processing: Simplifies and structures raw text to make it analyzable for NLP tasks.
- Feature Extraction: Enables feature creation for models by isolating words or phrases.
- Understanding Language: Helps capture the distribution and sequence of words.

Challenges

- Handling Punctuation: Deciding whether punctuation should be retained as separate tokens.
- Compound Words: Maintaining words like “New York” as a single token.
- Language Complexity: Adapting tokenization rules for languages without spaces (e.g., Chinese).

Quick Demo - 4

- Word Embedding
- Sentence Embedding

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple, non-parametric, and supervised learning algorithm used for both classification and regression tasks. KNN classifies new data points based on the "distance" from other points in the dataset.

Applications of KNN

- Sentiment Analysis: Classifying text data (e.g., reviews) into positive or negative sentiment.
- Recommender Systems: Finding similar items or users for personalized recommendations.
- Anomaly Detection: Identifying unusual patterns in data, such as fraud detection.

How KNN Works

1. Distance Calculation: Measures the similarity between data points using distance metrics (e.g., Euclidean, Manhattan).
2. Neighbor Selection: Selects the 'k' nearest neighbors of the data point.
3. Classification/Prediction:
    - Classification: The class with the majority among the k-neighbors is assigned to the data point.
    - Regression: Takes the average (or weighted average) of neighbors’ values to predict.

Quick Demo - 5 - KNN

Recap of the Day

Text data

Text preprocessing (Cleaning and formatting)

Stop words removal, Stemming, Lemmatization

Vectorization (Convert to numbers)

Bag of Words

Train ML Model using numerical data

K Nearest Neighbors (KNN), Neural Network, etc.

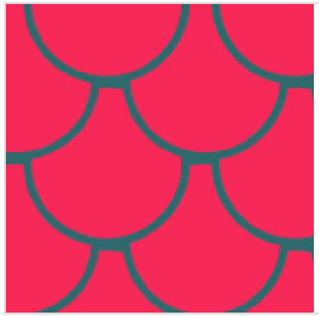
Office Hours Week 1
Discussion - Exercise - Day 2

171k product review with Sentiment Dataset

Customer Review on product with sentiment

kaggle.com

https://www.kaggle.com/datasets/mansithummar67/171k-product-review-with-sentiment-dataset?



E-Commerce Dataset for Practice

Kaggle is the world’s largest data science community with powerful tools and resources to help you achieve your data science goals.

kaggle.com

https://www.kaggle.com/datasets/shivrajguvi/e-commerce-dataset-for-practice

NLP Sessions Overview - Week 2

Session 2

- Parts of Speech (POS) Tagging and Demo
- Grammar, Syntax, and Parsing Techniques
- Parsing Demo: Dependency Parsing with Spacy
- Introduction to Encoder-Decoder Models
- Hands-On: Building a Simple Encoder-Decoder

Office Hours 2

- Recap and Discussion on Parsing and Encoder-Decoder Models
- Review Exercises (Hugging Face, Kaggle)