Natural Language Processing (NLP)

Vector Representation

Equipping You with Research Depth and Industry Skills

By:

Dr. Zohair Ahmed

www.youtube.com/@ZohairAI Subscribe

www.begindiscovery.com

NLP (Natural Language Processing)

- **NLU + NLG ⊂ NLP**
- **NLP (Natural Language Processing)**
- The **big umbrella field**.
- Any computer method that deals with human language → text or speech.
- Includes tasks like tokenization, sentiment analysis, translation, summarization, chatbots, etc.
- **NLU (Natural Language Understanding)**
- A **subfield of NLP**.
- Focus: **understand the meaning** of language.
- Examples:
  - Intent detection (*"Book me a flight" → intent = flight booking*)
  - Named Entity Recognition (*"Paris" → location*)
  - Sentiment analysis (*positive/negative*)
- **NLG (Natural Language Generation)**
- Another **subfield of NLP**.
- Focus: **produce language** that sounds natural.
- Examples:
  - Chatbot replies
  - Machine translation (English → Arabic)
  - Text summarization

What are Features?

- **ML definition**: Attributes describing data
- Example: *House Price Prediction*
  - Area, Location, Facilities, Age → Features
- Example: *Image Classification*
  - Ears, Nose, Eyes, Whiskers → Features
- Features = characteristics used by models to make decisions

From Images to Text

- Images → Ears, Nose, Eyes = Features
- Property Price → Area, Facilities = Features
- Text → ??? (needs conversion → numbers)

Why Convert Text to Numbers?

- ML models **cannot process text directly**
- Require **numeric representation**
- Text → Feature Vectors
- Enables **math operations** (e.g., cosine similarity)

Example

- Text: **Afridi, Cummins, Australia**
- Afridi → Person = 1, Location = 0
- Cummins → Person = 1, Location = 0
- Australia → Person = 0, Location = 1
- Handcrafted Features → **Feature Vector**

Benefits of Vector Representation

- Similar words → similar vectors
  - **Afridi** ≈ *Cummins*
  - *Bad* ≈ *Worse*
- Helps in tasks like **Sentiment Analysis**, **Text Classification**

Benefits of Vector Representation

- Similar words → similar vectors
  - **Afridi** ≈ *Cummins*
  - *Bad* ≈ *Worse*
- Helps in tasks like **Sentiment Analysis**, **Text Classification**

Vector Space Model

- Represent **words / phrases / sentences / paragraphs** as vectors
- Called **Vector Space Model**
- Core of **text representation in NLP**

Common Techniques

One-Hot Encoding

Bag of Words (BoW)

TF-IDF

Word Embeddings (Word2Vec, GloVe, FastText)

Contextual Embeddings (BERT, GPT)

Key Insight

- From Practical NLP book:
- “Feeding a good representation to an ordinary algorithm often beats applying a top-notch algorithm to poor text representation.”

One-Hot Encoding (earliest & simplest)

- Each word → unique position in a vector
- Example vocabulary: {cat, dog, ball}
  - cat = [1,0,0]
  - dog = [0,1,0]
  - ball = [0,0,1]
- Simple & clear
- Very sparse, no word similarity captured

Bag of Words (BoW)

- Represent text by **word counts**
- Example:
  - Sentence 1: “I love cats” → [1,1,0]
  - Sentence 2: “I love dogs” → [1,0,1]
- Easy to use
- Ignores word order, meaning

TF–IDF (Term Frequency – Inverse Document Frequency)

- Improves BoW by **down-weighting common words**
- Formula = (Word frequency) × (Inverse of document frequency)

$$\text{TF–IDF}(w, d) = \text{TF}(w, d) \times \text{IDF}(w)$$

  - N = total number of documents
  - $df(w)$ = number of documents containing the word

$$\text{IDF}(w) = \log\left(\frac{N}{df(w)}\right)$$

- Example:
  - “the, is, and” → low weight
  - “cancer, treatment” → higher weight
- Captures importance of words
- Still ignores word order & context

Sparse Representation

- **Definition**: Most entries in the vector are **0**.
- Example (One-Hot Encoding):

Word	cat	dog	ball	tree	run	play	…
"cat"	1	0	0	0	0	0	…

- If vocab = 10,000 words, each word vector = length 10,000 with **only 1 non-zero**.
- Simple to build
- Very **memory-heavy** and doesn’t capture meaning

Dense Representation

- **Definition**: Vectors have **few or no zeros**; values are spread across dimensions.
- Example (Word Embedding for "cat"):
- (vector length maybe 100–300, not 10,000)
- Each number encodes **semantic meaning**.
- Compact, efficient
- Captures similarity (cat ≈ dog)
- Harder to design manually (needs training/learning)