DAY – 7

6 August 2025

Tokenization

Tokenization is the foundational step of breaking down a continuous stream of text into smaller, meaningful units called tokens. These tokens can be words, subwords, or characters, and serve as the basic building blocks for further analysis.

Types of Tokenization

- **Word-Level Tokenization:** Splits text into individual words, typically using whitespace and punctuation as delimiters. This is straightforward for languages like English but can struggle with contractions or compound words.
- **Character-Level Tokenization:** Breaks text into individual characters. This is useful for languages without clear word boundaries (like Chinese) or for tasks like spelling correction.
- **Subword Tokenization:** A hybrid approach that divides words into smaller units (morphemes or frequent character sequences). This method manages vocabulary size effectively and handles out-of-vocabulary (OOV) words gracefully.
- **Byte Pair Encoding (BPE):** Learns merge rules to combine the most frequent pairs of tokens, starting from individual characters. Used in models like GPT.
- **WordPiece:** Similar to BPE, it merges token pairs based on a frequency-weighted score. Used in models like BERT.
- **SentencePiece:** A language-independent method that treats the entire input as a raw stream and segments it into subwords.
- **Sentence Tokenization:** Splits a large text corpus into individual sentences, which can then be further tokenized at the word or character level.

Normalization

Normalization is the process of standardizing data to a consistent format or scale, which helps reduce variations and inconsistencies. In NLP, it ensures that different forms of a word (e.g., "running", "runs", "ran") are treated the same way.

Types of Normalization

- **Text Normalization (NLP):** Involves several pre-processing steps:
- **Lowercasing:** Converting all text to lowercase to ensure consistency (e.g., "The" and "the" are treated as the same token).
- **Stemming:** Reducing words to their root or stem form (e.g., "running" becomes "run"). Stems are often not actual words.
- **Lemmatization:** Reducing words to their base or lemma (dictionary form) (e.g., "running" becomes "run", "am" becomes "be"). This is generally more sophisticated than stemming as it considers context.
- **Removing noise**: Eliminating irrelevant elements like punctuation, special characters, HTML tags, or stop words (common words like "a", "an", "the").
- **Numerical Data Normalization (Machine Learning):** Scales numerical features to a similar range to prevent features with large magnitudes from dominating those with smaller magnitudes during model training.
- **Linear Scaling (Min-Max Scaling):** Rescales the data to a fixed range, usually [0, 1].
- **Z-score Scaling (Standardization):** Transforms the data to have a mean of zero and a standard deviation of one, useful for normally distributed data.
- **L1/L2 Normalization:** Used specifically for vectors, these methods scale the vector so it has a magnitude (length) of one (L2 normalization) or the sum of its absolute components is one (L1 normalization), which is useful for distance calculations like cosine similarity.

Vectorization

Vectorization (or text representation/embedding) is the process of converting data, especially text, into a numerical vector format that machine learning models can understand and process.

Types of Vectorization

- **Frequency-Based Methods**:
- **Bag-of-Words (BoW):** Represents text as an unordered collection of words, typically by counting word occurrences in a document. It results in a sparse matrix.

- **TF-IDF (Term Frequency-Inverse Document Frequency):** A more advanced frequency method that weighs words by their importance. It gives high scores to words that are frequent in a specific document but rare across the entire corpus, down-weighting common words like "the".
- **Word Embeddings:** Modern techniques that learn dense vector representations where words with similar meanings are located close to each other in a high-dimensional space.
- **Word2Vec:** Uses a shallow neural network with two main architectures, Continuous Bag-of-Words (CBOW) and Skip-gram, to learn word representations from context.
- **GloVe (Global Vectors for Word Representation):** Creates word vectors by analyzing global word co-occurrence statistics across the entire corpus.
- **Transformer-Based Embeddings (BERT, GPT, etc.):** These state-of-the-art models generate contextual embeddings, where the vector representation of a word changes based on the surrounding words in the sentence.

Streaming

Streaming refers to a data processing paradigm where data is processed continuously as it flows through a system, rather than being processed in large batches (batch processing).

- **Real-time Streaming:** Data is processed immediately as it arrives, suitable for applications requiring low latency responses, such as fraud detection, live analytics, or real-time NLP interactions (e.g., a chatbot responding instantly).
- **NLP and Streaming:** In an NLP pipeline, streaming allows models to analyze incoming text data sequentially (e.g., processing a continuous feed of social media posts for sentiment analysis) without waiting for an entire dataset to accumulate. This ensures timely insights and efficient resource usage.

Embedding

In machine learning, an embedding is a numerical, vector representation of complex data (like words, images, or audio) in a lower-dimensional space. The primary goal is to map semantically or functionally similar items to points that are close to one another in this continuous vector space.

Core Concepts

- **Numerical Representation:** Computers can only process numbers. Embeddings translate unstructured data into a list of floating-point numbers (a vector) that machine learning models can understand and use efficiently.
- **Dimensionality Reduction:** Embeddings convert high-dimensional data (e.g., a vocabulary of thousands of words) into a dense, lower-dimensional space, which drastically reduces computational costs and memory requirements compared to traditional methods like one-hot encoding.
- **Semantic Meaning:** The key feature of embeddings is that the geometric distance between points in the vector space is meaningful. For example, the vector for "king" is mathematically close to "queen," and the relationship between "man" and "woman" is mirrored in the relationship between "king" and "queen".
- **Similarity Search:** Because similar items are close together, embeddings allow for efficient "similarity searches". This powers applications like finding related documents, recommending similar products, or identifying the most relevant answer to a query.

Applications

Embeddings are foundational to modern AI systems, particularly Large Language Models (LLMs). Common uses include:

- **Natural Language Processing (NLP):** Used for sentiment analysis, machine translation, text classification, and powering advanced search engines.
- **Computer Vision**: Image embeddings capture visual features, enabling tasks like image similarity search and facial recognition.
- **Recommender Systems:** Embeddings learn representations for both users and items (e.g., movies, products), allowing models to predict user preferences and provide personalized recommendations.
- **Audio Processing and Graph Analysis:** They can also represent audio signals or nodes in a network (like a social graph), enabling analysis and pattern detection in various domains.