Embeddings in NLP

Vector Representations of Text and Meaning

1. Introduction

Embeddings are numerical vector representations of text, images, or other data that capture semantic meaning. In NLP, embeddings allow algorithms to measure similarity between words, sentences, or documents in a high-dimensional space.

2. Why Embeddings Are Needed

Text in its raw form (strings) cannot be processed directly by mathematical models. Embeddings convert text into vectors where *semantically similar items are close together*, enabling tasks like:

- Semantic search
- Clustering and classification
- Recommendation systems
- Context-aware chatbots

3. Types of Embeddings

3.1. Word Embeddings

- **Word2Vec:** Predicts words from context (CBOW/Skip-gram).
- **GloVe:** Matrix factorization on word co-occurrence statistics.
- **FastText:** Uses subword information for rare words.

3.2. Sentence / Document Embeddings

- Converts sentences or documents into a single vector.
- Examples: `Sentence-BERT`, `Universal Sentence Encoder`.

3.3. Contextual Embeddings (Transformers)

- Each token embedding depends on context in the sentence.
- Examples: BERT, GPT embeddings, RoBERTa.

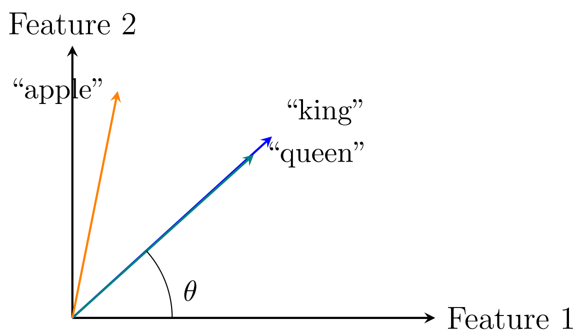
4. How Embeddings Work

1. Tokenize text into words or subwords.
2. Convert tokens to vectors using embedding models.
3. Contextual models adjust vectors based on surrounding words.
4. Resulting vectors are used for similarity calculations or downstream tasks.

5. Vector Similarity

The similarity between two embeddings can be computed using:

- **Cosine Similarity:** Measures angle between vectors.
- **Euclidean Distance:** Measures straight-line distance.



6. Example: Word2Vec Embeddings

Words: "king", "queen", "man", "woman"
Word vectors satisfy relationships like:

$$\vec{\text{king}} - \vec{\text{man}} + \vec{\text{woman}} \approx \vec{\text{queen}}$$

This demonstrates semantic understanding captured by embeddings.

7. Applications of Embeddings

- Semantic search and RAG systems.
- Text classification and clustering.
- Recommendation systems and personalization.
- Chatbots and conversational AI.
- Sentiment analysis and topic modeling.

8. Python-style Implementation Example

```
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity

model = SentenceTransformer('all-MiniLM-L6-v2')
sentences = ["I love machine learning.",
             "Deep learning is amazing."]

embeddings = model.encode(sentences)
similarity = cosine_similarity([embeddings[0]], [embeddings[1]])

print("Similarity:", similarity[0][0])
```

Summary

Embeddings convert text into numerical vectors capturing semantic meaning. They are the foundation of modern NLP applications, including semantic search, RAG systems, chatbots, and recommendation engines. Understanding embeddings is essential for building intelligent, context-aware AI systems.