Embedding Module: From Fundamentals to Deep Representations

DL & GenAI Project [BSDA2001P]
Indian Institute of Technology Madras

INDRANIL BHATTACHARYYA

DATA SCIENTIST, RENAULT NISSAN

Agenda

- Setting the Stage: Why GPU for NLP
- From TF-IDF to Deep Embeddings
- Accelerating Embedding Computation with GPUs
- Matryoshka Representation Learning (MRL)
- Wrap-Up & Takeaways

Why GPU for NLP?

Method	Hardware	Time (per 1k sentences)
TF-IDF	CPU	~2.1s
BERT (base)	CPU	~80s
BERT (base)	GPU	~4.5s

- Modern NLP models → billions of parameters → need parallel tensor operations.
- GPUs accelerate:
  - Matrix multiplications
  - Batch processing
  - Embedding generation for large corpora

Activity – GPU In Kaggle

- Enable GPU in Kaggle
- Verify GPU availability through code

From TF-IDF to Deep Embeddings: The Evolution of Text Representations

Traditional Representations: Sparse and Static

$$\text{TF-IDF}(t,d) = \text{tf}(t,d) \times \log \frac{N}{df(t)}$$

- Conceptual Foundations
  - Bag of Words (BoW): Counts term occurrences — ignores order & semantics.
  - TF-IDF: Weights rare words higher but still **independent of context**.
- Limitations:
  - High dimensional & sparse vectors
  - No notion of **semantic similarity**
  - Fails on polysemous words (e.g., *bank* → *river* / *finance*)

Dense Embeddings

- The Transition Phase:
  - **Word2Vec / GloVe:** Capture *co-occurrence statistics* via shallow neural nets.
  - Learn **dense**, low-dimensional embeddings (~300D).
  - Each word → a **single fixed vector** representing global meaning.
- *Properties:*
  - Enables vector arithmetic → *king – man + woman ≈ queen*
  - Still static → cannot disambiguate "apple" (fruit vs company)

Deep Contextual Representations

Contextual Embeddings with Transformers

ELMo, BERT, RoBERTa: Represent words in *context* using self-attention.

Embedding of a word depends on *surrounding tokens* — **dynamic meaning**.

Multi-layer representations capture hierarchy:

- Lower layers → syntax
- Middle → semantics
- Upper → task-specific nuances

Activity: Semantic Similarity

- We will use Cosine Similarity to measure the similarity between two sentences.
- Will compare:
  - Tf-IDF Embedding
  - Word2Vec
  - Transformer-based embedding

Comparative Insights

Deep embeddings compress semantics → fewer dimensions, richer relationships.

Representation	Contextual	Dimensionality	Training	Use-case
BoW / TF-IDF	❌	10k+ (Sparse)	None	Simple baselines
Word2Vec	Partial	~300	Self-supervised	Lightweight NLP
BERT / SBERT	✅	384–1024	Pre-trained Transformers	Semantic tasks, Sentiment, QA

Demo: Visualization Insight

- t-SNE / UMAP:
  - TF-IDF clusters by *keywords*
  - BERT clusters by *meaning*

Matryoshka Representation Learning (MRL)

The Problem — Embedding Efficiency at Scale

- **Context:**
  - Modern sentence embeddings (e.g., 768–1024D) are **computationally expensive**.
  - Real-world NLP tasks (e.g., retrieval, clustering, QA) don't always need *full precision embeddings*.
  - Need for **compact**, *multi-resolution* embeddings — without retraining for every size.
- **Challenge:**
  - Can we build one embedding space that performs well at multiple dimensionalities?

What is Matryoshka Representation Learning?

Formal Intuition:

If $f(x) \in \mathbb{R}^d$ is the full embedding,

then $f_k(x) = f(x)[:k]$ (the first *k* dimensions)

should maintain meaningful representation quality.

- Core Idea:
  - Like Russian nesting dolls 🪆 — **embeddings contain smaller embeddings** within them.
  - A single model is trained so that **progressively truncated embeddings** (e.g., first 256D of 1024D) **still perform well** on downstream tasks.
  - *Key Property*: Each prefix of the vector is itself a valid embedding.

Why this? (Training & Use Case)

- Multi-Scale Training:
  - The model produces a *hierarchical embedding vector*.
  - During training, multiple truncated versions are supervised to align with the full embedding space.
  - Output embedding: same meaning, smaller footprint.
- Deployment Flexibility:
  - Choose embedding size based on resource constraints:
    - Server → 768D
    - Edge device → 256D

Questions?