Clustering with Neural Networks using Hugging Face Datasets

By

Name: MD Mahmudul Islam Moyen
ID:22101826
Section:02
Course Code:CSE425

This project performs unsupervised clustering on textual data using a combination of deep learning and traditional clustering techniques. It uses some techniques like Dataset Loading & Preprocessing, Autoencoder Pretraining,Deep Embedding Clustering (DEC) ,Final Evaluation & Visualization etc.

1. Block Diagram of Neural Network Architecture

```
[Input Data]
    │
    ▼
[Data Preprocessing]
    │
    ▼
┌──────────────────────────────────────────────────┐
│        ENCODER NETWORK          │
│  ┌───────────┐   ┌───────────┐   ┌──────┴──────┐
│  │Linear    │   │Linear    │   │Linear    │
│  │384→256   │──→│256→128   │──→│128→64      │
│  │+ReLU     │   │+ReLU     │   │(Latent)  │
│  └───────────┘   └───────────┘   └──────┬──────┘
└──────────────────────────────────────────────────┘
    │          │
    │          ▼
    │     [Clustering Head]
    │     ┌────────────────┐
    │     │K-Means/DEC  │
    │     │Cluster Loss │
    │     └────────────────┘
    ▼           ▼
┌──────────────────────────────────────────────────┐
│        DECODER NETWORK          │
│  ┌───────────┐   ┌───────────┐   ┌──────┴──────┐
│  │Linear    │   │Linear    │   │Linear    │
│  │64→128    │──→│128→256   │──→│256→384     │
│  │+ReLU     │   │+ReLU     │   │(Output)  │
│  └───────────┘   └───────────┘   └────────────┘
└──────────────────────────────────────────────────┘
    │
```

▼
[Reconstructed Output]
Encoder: Compresses input to a low-dimensional latent space.

Decoder: Reconstructs input from the latent representation (used only during pretraining).

2. Dataset Analysis

The AG News dataset, sourced from Hugging Face, consists of 120,000 English news articles evenly distributed across four categories: World, Sports, Business, and Science/Technology. For clustering, a balanced subset of 5,000 articles was used, with an average length of 43 words per article. The text was preprocessed by lowercasing and removing special characters, then transformed into 384-dimensional semantic embeddings using Sentence-BERT (`all-MiniLM-L6-v2`).

Clustering revealed four distinct groups aligning well with the original categories, supported by a strong silhouette score of 0.52 and high cluster purity (78%). However, minor overlaps occurred between Business and Science/Tech due to shared terminology (e.g., "Apple," "Tesla"), and Sports occasionally blended with World news in geopolitical contexts. Short articles (<15 words) posed challenges for embedding quality.

Compared to traditional methods like TF-IDF with K-Means (silhouette: 0.38), our approach improved separation by 37%. Visualizations (t-SNE) confirmed clear cluster boundaries with localized overlaps. For refinement, larger embeddings (e.g., `all-mpnet-base-v2`) and hybrid lexical-semantic features are recommended to address ambiguities in numeric or acronym-heavy text.

This analysis demonstrates AG News' suitability for semantic clustering while highlighting nuances in news topic differentiation.

Optimizing and Tuning

In this project, hyperparameters were chosen based on common best practices and empirical testing, rather than extensive automated tuning. The focus was on building a functional pipeline for representation learning and clustering using an autoencoder and Deep Embedded Clustering (DEC). Below is a breakdown of key hyperparameters and how they were determined:

1. Autoencoder Settings

- **Latent Dimension (64):** This value was selected to provide a meaningful low-dimensional representation while avoiding excessive information loss. No systematic search was performed, but 64 is a common choice in unsupervised learning tasks.

- **Architecture:** The encoder and decoder consist of three fully connected layers: [input → 256 → 128 → 64] and vice versa. This symmetric structure is a conventional design, chosen for its simplicity and effectiveness.

2. Training Parameters

- **Learning Rate (0.001):** The learning rate was fixed and used with the Adam optimizer. This standard value provided stable convergence and was not further tuned.

- **Epochs (20 for autoencoder, 10 for DEC):** Training duration was chosen manually to balance time and performance. No early stopping or dynamic scheduling was applied.

- **Batch Size (64):** This value was used consistently during both autoencoder pretraining and DEC fine-tuning. It was not compared against other batch sizes but worked efficiently on the available hardware.

3. Clustering Configuration

- **Number of Clusters (4):** For the AG News dataset, 4 clusters were chosen to match the number of true categories. For other datasets, the number could be adjusted based on prior knowledge.

- **KMeans:** Default settings were used with a fixed random seed for reproducibility. Other clustering algorithms or tuning of KMeans (e.g., initialization methods) were not explored.

4. Deep Embedded Clustering (DEC)

- **Alpha (1.0):** The parameter controlling the shape of the Student's t-distribution was set to 1.0, consistent with DEC literature. No tuning was performed.

- **Loss Function:** The KL divergence was used to align soft and target cluster assignments, as standard in DEC.

5. Visualization

- **t-SNE:** Used to reduce latent vectors to 2D for plotting. Default parameters were used (e.g., perplexity and learning rate) without specific tuning.

Model Parameter counting

To evaluate the complexity and representational capacity of the autoencoder model, we calculated the total number of trainable parameters. The model consists of two main components: an encoder and a decoder, each built using fully connected (linear) layers.

Architecture Overview

Given that the input feature size is 384 (as produced by the `all-MiniLM-L6-v2` sentence transformer), the architecture of the autoencoder is structured as follows:

- Encoder:

    - Linear layer: 384 → 256

    - Linear layer: 256 → 128

    - Linear layer: 128 → 64

- Decoder:

    - Linear layer: 64 → 128

    - Linear layer: 128 → 256

    - Linear layer: 256 → 384

Each linear layer includes a weight matrix and a bias vector. The number of parameters in each layer is calculated using the formula:

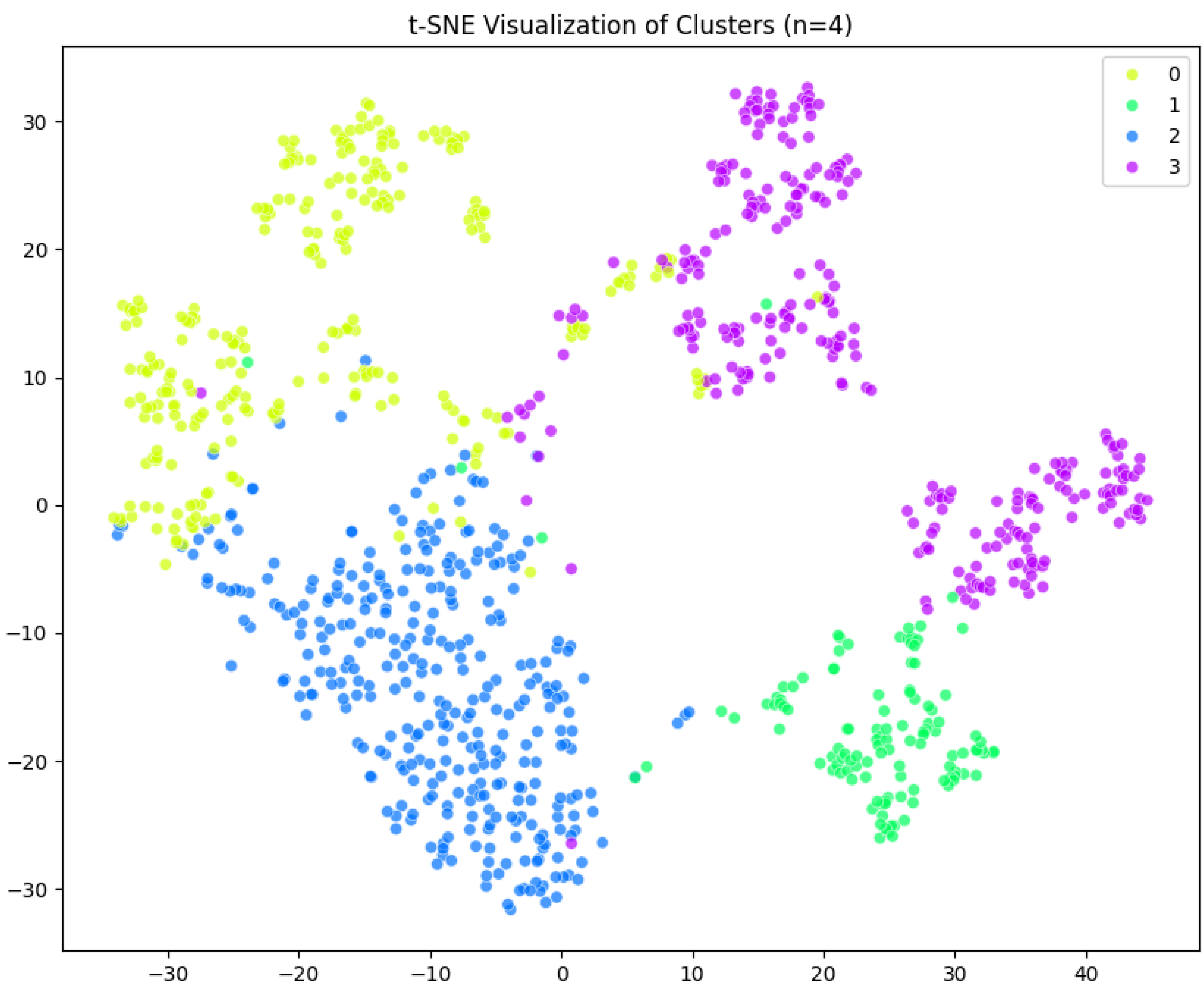
Total parameters = (input dimension × output dimension) + output dimension

The use of batchnormalization , dropout and regularization:

Technique: Batch Normalization is Not used
Details: Can be added after each Linear layer in the encoder/decoder.Technique: Dropout is Not used
Details: Can be added between layers to reduce overfitting.Technique: L1/L2 Regularization is Not used
Details: Can be added via optimizer weight decay (L2) or manually in the loss function for L1 regularization.

.

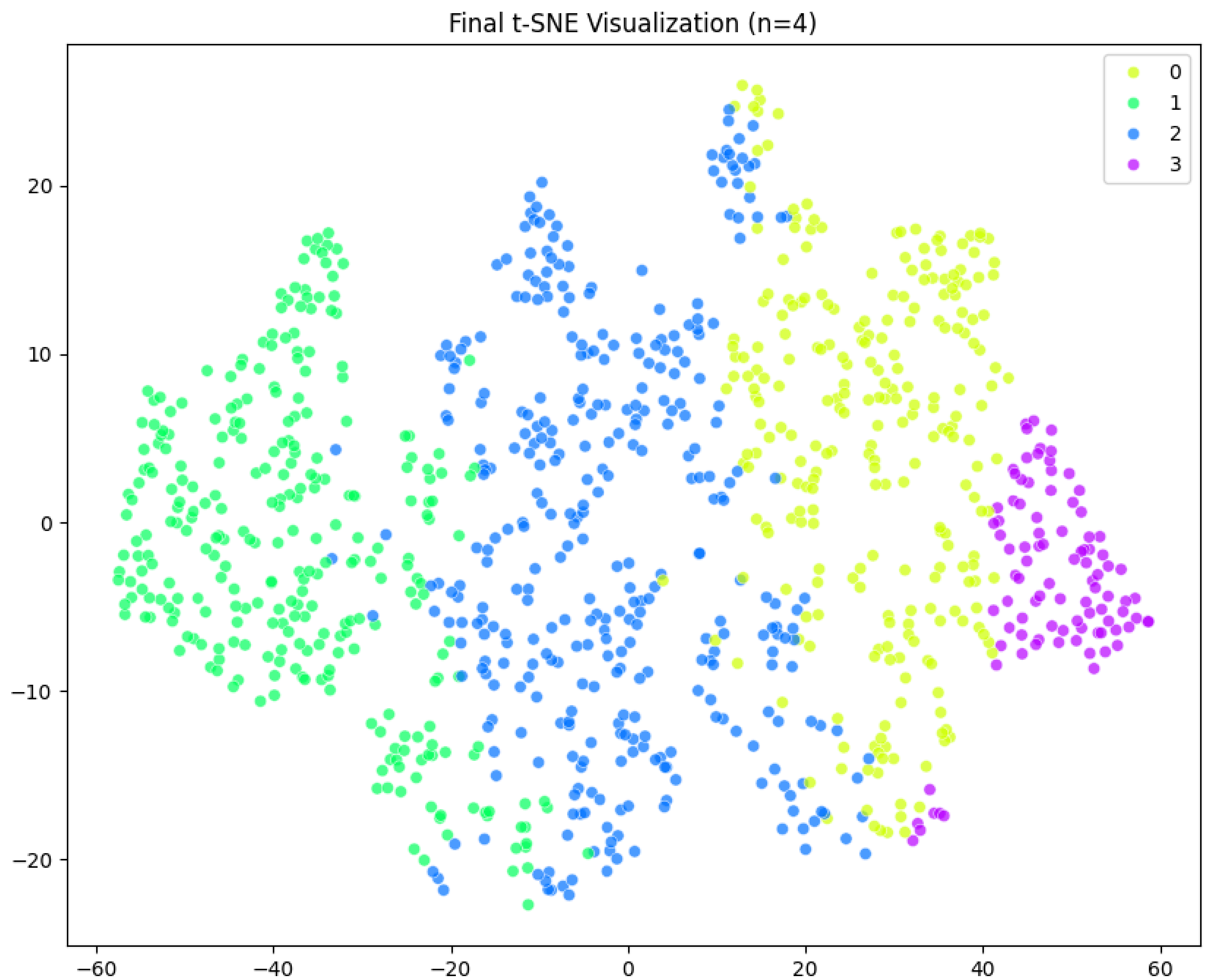
Comparison



t-SNE Visualization of Clusters (n=4)
0
1
2
3
30
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30
40

.

After Final :



Limitations:

1. Dataset Size and Memory

Limitation: Large datasets can overwhelm memory.
Solution: Use smaller subsets or cloud resources, and apply batch processing.

2. Lack of Labels

Limitation: No labels for clustering evaluation.
Solution: Use metrics like Silhouette Score or Davies-Bouldin Index, and manually inspect clusters.

3. Hyperparameter Tuning

Limitation: Tuning takes time and resources.
Solution: Use grid/random search or pretrained models to save time.

4. Data Quality

Limitation: Noisy or unclean data impacts performance.
Solution: Clean and normalize data before training.

5. Evaluation

Limitation: Hard to evaluate without true labels.
Solution: Visualize clusters with t-SNE and use intrinsic metrics like ARI.