Cross-Domain Multilingual Text Clustering Analysis

Zhengze Yu
University of Chinese Academy of Sciences

1. Dataset Preparation

We use the English source corpus from the multi-domain translation dataset as the text to be clustered. The dataset used in this project is available at my **GitHub Repository (data folder)**.

The monolingual corpus covers five distinct domains:

- IT (Information Technology)
- Koran
- Law
- Medical
- Subtitles

Each domain contains thousands of sentence samples. We extract hidden-layer features from BERT and apply Principal Component Analysis (PCA) for two-dimensional visualization. As shown in Figure 1, the BERT-based representations exhibit clear domain separation, indicating that BERT embeddings capture sufficient semantic information for text clustering tasks.

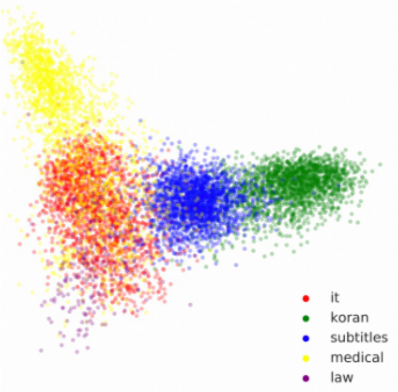

Figure 1. PCA visualization of BERT-based representations across domains.

2. K-Means Clustering

Given a set of n data points $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^d$, K-Means aims to partition them into K clusters $\{C_1, C_2, \ldots, C_K\}$ by minimizing the within-cluster sum of squares:

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

where μ_k is the centroid of cluster C_k:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

The optimization iterates between two steps:

1. **Assignment step:** Assign each data point to the nearest centroid

   $$C_k = \{x_i : \|x_i - \mu_k\|^2 \leq \|x_i - \mu_j\|^2, \forall j = 1, \ldots, K\}$$

2. **Update step:** Recompute each centroid as the mean of points in the cluster

   $$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

The algorithm repeats until convergence, i.e., cluster assignments do not change or the loss $\mathcal{L}$ does not decrease significantly.

Algorithm 1 K-Means Clustering

```
Initialize $K$ centroids randomly: $\{\mu_1, \dots, \mu_K\}$
repeat
    Assignment step: For each data point $x_i$, assign it to the nearest centroid:
        $c_i = \arg\min_k \|x_i - \mu_k\|^2$
    Update step: For each cluster $k$, update the centroid:
        $\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$
until centroids do not change or maximum iterations reached
```

3. Methodology

All sentences are first encoded using a pretrained BERT model. The tokenizer and model weights are loaded to convert raw text into token sequences, which are then fed into the BERT encoder. The average pooling of the final hidden layer is taken as the sentence-level representation. These feature vectors are subsequently clustered using the K-Means algorithm implemented in `scikit-learn`.

The training process is performed by calling the `fit()` function, which iteratively optimizes the cluster centroids. After training, the `predict()` function assigns each sentence to the nearest centroid to determine its cluster label.The code is also available at my **GitHub Repository (code folder)**.

4. Experimental Results

The clustering performance is evaluated using supervised accuracy as the metric. For each cluster, the majority domain label is assigned as its predicted label. For example, if Cluster 0 contains 1000 *IT*, 250 *Koran*, 200 *Law*, 100 *Medical*, and 50 *Subtitles* samples, its accuracy is calculated by dividing the number of correctly clustered samples by the total number of samples, that is, 1000 out of 1600, or 62.5%.

The overall average accuracy across all clusters reaches **85.6%**. The confusion matrix (Figure 2) shows that samples from *Medical* and *IT*, as well as from *Law* and *IT*, are more likely to be confused, reflecting semantic overlaps among these domains.

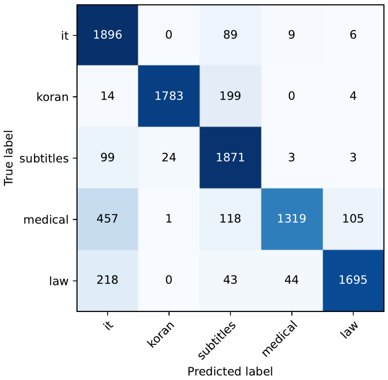

Figure 2. Confusion matrix of clustering results across five domains.