**Sarvam Research Fellow Assignment**

**-Ojas Jain**

**(**[**ojasjain94@gmail.com**](mailto:ojasjain94@gmail.com)**)**

The code implements a supervised cross-lingual word embedding alignment system using the Procrustes method, evaluates its performance, and conducts ablation studies. It also includes an optional unsupervised alignment method using CSLS and adversarial training.

## Code Structure

## **1. Data Preparation**

The code begins by loading pre-trained FastText embeddings for English and Hindi, limiting the vocabulary to the top 100,000 words for each language.

Key Steps:

* Load embeddings using gensim.models.KeyedVectors.
* Filter embeddings to include only the top 100,000 most frequent words.

## **2. Downloading and Loading Bilingual Dictionary**

The bilingual dictionary is downloaded from the MUSE dataset and processed to extract valid word pairs for supervised alignment.

## **Function: download\_muse\_dictionary**

Downloads bilingual dictionaries for English-Hindi from the MUSE dataset.

* Inputs: Source language (src\_lang), Target language (tgt\_lang).
* Outputs: Paths to downloaded dictionary files.

## **Function: load\_dictionary**

Processes dictionary files to create a list of valid word pairs.

* Inputs: Dictionary path (dict\_path), source vocabulary (src\_vocab), target vocabulary (tgt\_vocab).
* Outputs: List of valid word pairs.

## **3. Procrustes Alignment**

The Procrustes method is used to learn a linear mapping between source (English) and target (Hindi) embeddings using the bilingual dictionary.

## **Function: learn\_procrustes\_alignment**

Learns an orthogonal mapping matrix using Procrustes analysis.

* Inputs: Source embeddings (src\_embs), Target embeddings (tgt\_embs), Bilingual dictionary (bilingual\_dict).
* Outputs: Mapping matrix (W).

## **4. Evaluation**

The aligned embeddings are evaluated using translation accuracy metrics: Precision@1 and Precision@5.

## **Function: compute\_precision\_at\_k**

Computes translation accuracy for word pairs based on cosine similarity.

* Inputs: Source embeddings (src\_embs), Target embeddings (tgt\_embs), Test dictionary (test\_dict), Mapping matrix (mapping\_matrix), Precision level (k).
* Outputs: Precision score (p\_at\_k) and predicted translations.

## **5. Ablation Study**

An ablation study is conducted to assess the impact of bilingual lexicon size on alignment quality.

## **Function: ablation\_study**

Tests different dictionary sizes and evaluates translation accuracy.

* Inputs: Source embeddings (src\_embs), Target embeddings (tgt\_embs), Training dictionary (train\_dict), Test dictionary (test\_dict), List of dictionary sizes (dict\_sizes).
* Outputs: Results with Precision@1 and Precision@5 scores for each dictionary size.

## **6. Cosine Similarity Analysis**

Analyzes cosine similarities between word pairs after alignment.

## **Function: analyze\_cosine\_similarities**

Computes cosine similarity between aligned word pairs.

* Inputs: Source embeddings (src\_embs), Target embeddings (tgt\_embs), Test dictionary (test\_dict), Mapping matrix (mapping\_matrix).
* Outputs: List of cosine similarities between word pairs.

## **7. Visualization**

Plots results from the ablation study to visualize the impact of bilingual dictionary size on translation accuracy.

## **Function: plot\_ablation\_results**

Plots Precision@1 and Precision@5 scores against dictionary sizes.

* Inputs: Results from ablation study, Dictionary sizes.
* Outputs: Saves plot as ablation\_results.png.

## **8. Optional Unsupervised Alignment**

**Note-**The optional unsupervised method could not be executed due to computational constraints but is included in the code for reference.

Implements unsupervised alignment using Cross-Domain Similarity Local Scaling (CSLS) combined with adversarial training.

## **Function: unsupervised\_alignment**

Uses CSLS and adversarial training to refine alignment iteratively.

* Inputs: Source embeddings, Target embeddings, Number of iterations.
* Outputs: Refined mapping matrix.

## **Function: csls\_retrieval**

Retrieves nearest neighbors using CSLS for improved alignment quality.

* Inputs: Source embeddings, Target embeddings, Mapping matrix, Number of neighbors (k).
* Outputs: CSLS scores.

## 

## **9. Comparison of Methods**

Compares supervised Procrustes alignment with unsupervised CSLS-based alignment methods.

## **Function: compare\_methods**

Evaluates both supervised and unsupervised methods using Precision@1 and Precision@5 metrics.

* Inputs: Source embeddings, Target embeddings, Test dictionary.
* Outputs: Comparison results with visualization.

## **10. Main Function**

The main function orchestrates all steps:

1. Loads embeddings and dictionaries.
2. Performs Procrustes alignment.
3. Evaluates translation accuracy.
4. Conducts ablation study and cosine similarity analysis.
5. Visualizes results.

## 

## 

## 

## 

## **Function: main()**

## Key Outputs

1. Translation Accuracy Metrics:
   * Precision@1
   * Precision@5
2. Cosine Similarity Analysis:
   * Top 10 most similar word pairs
   * Bottom 10 least similar word pairs
3. Ablation Study Results:
   * Impact of bilingual dictionary size on translation accuracy
4. Visualization:
   * Plots saved as PNG files

**Notes**

The optional unsupervised method could not be executed due to computational constraints but is included in the code for reference.

**How to run the file**

To run this code just change the path name as mentioned below after downloading the required files

I have uploaded those things on my drive and accessing them from there