Université Claude Bernard Lyon 1

Natural Language Processing (NLP)

Individual Assignment 1

Author: Muhammad Safi Ullah ADAM

Academic Year 2025–2026

17 February 2026

Contents

1 Introduction

This report presents a comprehensive exploration of multilingual word embeddings, focusing on aligning English and French semantic representations into a shared vector space. The primary objective of this project is to investigate how words from different languages can be mapped into a common embedding space and to evaluate the linguistic and computational properties of the aligned representations.

The analysis is based on parallel English–French text data and a bilingual dictionary (MUSE) used to guide supervised alignment. Word embeddings are trained separately for each language and subsequently aligned to enable cross-lingual semantic comparison. This alignment makes it possible to analyze semantic relationships across languages and to evaluate the embeddings in a downstream task.

The project is divided into several tasks, each addressing a specific aspect of multilingual representation learning:

- **Embedding Training:** Word embeddings are generated independently for English and French corpora using distributional learning methods, capturing semantic relationships within each language.
- **Cross-Lingual Alignment:** Using a bilingual dictionary, the independently trained embeddings are aligned into a shared vector space to enable semantic comparison between languages.
- **Linguistic Analysis:** The aligned embeddings are analyzed to evaluate semantic properties such as synonym similarity, antonym separation, and cross-lingual word correspondence.
- **Common Word Evaluation:** Frequently used words across both languages are examined to assess whether semantically equivalent terms occupy similar regions in the shared embedding space.
- **Downstream Classification Task:** The effectiveness of the aligned embeddings is evaluated through a language identification task, demonstrating their utility in practical NLP applications.
- **Visualization and Interpretation:** Dimensionality reduction techniques are used to visualize embedding distributions, providing qualitative insight into cross-lingual semantic structure.

The input data consists of parallel English–French sentences and a bilingual lexicon for supervised alignment. It is assumed that the corpora provide sufficient semantic coverage for meaningful embedding learning. The implementation is conducted in Python using standard NLP and scientific computing libraries within a Google Colab environment to ensure reproducibility and computational efficiency.

2 Dataset

2.1 Data Sources and Structure

The analysis is based on a parallel English–French corpus and a bilingual lexicon used for supervised alignment. The parallel sentences were obtained from the Tatoeba project and stored in TSV format, containing sentence identifiers and corresponding translations in both languages. Each record includes an English sentence paired with its French equivalent, enabling cross-lingual semantic learning.

In addition to the parallel corpus, the MUSE English–French bilingual dictionary was used to provide translation pairs for alignment. This dictionary contains word-level correspondences between the two languages. Only word pairs present in both embedding vocabularies were retained to ensure reliable alignment.

To support embedding training and comparison, pretrained GloVe vectors (6B, 100-dimensional) were loaded in the working directory (google drive) to provide globally learned semantic representations.

2.2 Corpus Preprocessing

Before training embeddings, the text data was cleaned and normalized to ensure consistency and reduce noise. The preprocessing pipeline included:

- **Lowercasing:** All text was converted to lowercase to avoid duplicate representations caused by case variations.
- **Text Cleaning:** Punctuation and non-alphabetic characters were removed where necessary to standardize tokens.
- **Tokenization:** Sentences were split into word tokens to prepare the corpus for embedding training.

The final dataset consists of **429,371 parallel English–French sentence pairs**, providing substantial bilingual context for embedding learning. The MUSE dictionary contains **113,286 translation pairs**, all of which were retained as usable alignment pairs after vocabulary filtering. This preprocessing step ensures that semantically identical words share the same representation and improves embedding quality by reducing sparsity.

2.3 Vocabulary Preparation

Separate vocabularies were built for English and French based on the tokenized corpora. Words with sufficient frequency were retained for embedding training, while rare or unseen tokens were handled through subword modeling when using FastText.

The bilingual dictionary was filtered to keep only translation pairs present in both vocabularies. This filtering step ensured that the alignment process used valid vector correspondences and avoided errors caused by missing embeddings. Embeddings were trained with 100-dimensional vectors, and low-frequency tokens were filtered to reduce noise and improve alignment reliability. The resulting preprocessed corpus and filtered dictionary provide the foundation for training robust monolingual embeddings and learning a reliable cross-lingual mapping.

3 Word Embedding Representation

3.1 Definition

In Natural Language Processing (NLP), a *word embedding* is a dense vector representation of a word in a continuous space, designed such that words with similar meanings obtain similar vectors. Instead of treating words as discrete symbols, embeddings map each word w to a real-valued vector:

$$\phi(w) = \mathbf{v}_w \in \mathbb{R}^d \tag{1}$$

where d is the embedding dimension (in this project, $d = 100$).

These representations are fundamental because they enable mathematical operations over language, allowing models to measure semantic similarity, retrieve nearest neighbors, and perform cross-lingual comparisons once embeddings are aligned into a shared space. A standard way to quantify similarity between two word vectors $\mathbf{v}_{w_i}$ and $\mathbf{v}_{w_j}$ is cosine similarity:

$$\text{cosine}(\mathbf{v}_{w_i}, \mathbf{v}_{w_j}) = \frac{\mathbf{v}_{w_i}^\top \mathbf{v}_{w_j}}{\|\mathbf{v}_{w_i}\| \, \|\mathbf{v}_{w_j}\|} \tag{2}$$

A higher cosine similarity indicates that the two words are closer in the embedding space and are therefore more semantically related under the distributional assumption that *"words appearing in similar contexts tend to have similar meanings."*

3.2 Embedding Methods Overview

Several techniques exist to transform textual data into numerical representations suitable for computational processing. In this project, five representation methods were explored to understand their properties and suitability for multilingual semantic analysis.

- **One-Hot Encoding:** Each word is represented by a binary vector whose length equals the vocabulary size. A single element corresponding to the word index is set to 1 while all others are 0. Although simple and lossless, this representation is high-dimensional, sparse, and does not capture semantic relationships between words.
- **TF-IDF (Term Frequency–Inverse Document Frequency):** TF-IDF represents words based on their importance within a document relative to a corpus. It down-weights frequently occurring words and highlights informative terms. While effective for document

similarity and information retrieval, TF-IDF vectors remain sparse and do not encode semantic similarity between words.

- **Word2Vec:** Word2Vec is a neural embedding method that learns dense vector representations by predicting context words from a target word (Skip-gram) or vice versa (CBOW). It captures semantic relationships through contextual co-occurrence patterns, enabling meaningful similarity comparisons between words.
- **FastText:** FastText extends Word2Vec by representing words as compositions of character n-grams. This subword modeling allows the method to generate embeddings for rare and out-of-vocabulary (OOV) words and to better capture morphological structure, making it particularly robust for multilingual and cross-lingual tasks.
- **GloVe (Global Vectors):** GloVe is a count-based embedding method that learns word vectors using global word co-occurrence statistics from large corpora. By combining global statistical information with vector space modeling, GloVe captures semantic relationships and provides stable pretrained embeddings.

Among these methods, distributional embeddings such as Word2Vec, FastText, and GloVe provide dense semantic representations, while FastText was specifically used for cross-lingual alignment due to its robustness to rare words and morphological variation.

3.3 Distributional Embeddings (Core Methods)

To capture semantic relationships between words, this project relies on *distributional embeddings*, which are based on the distributional hypothesis that words appearing in similar contexts tend to share similar meanings. Unlike sparse representations, these methods learn dense vectors that encode semantic and syntactic structure.

3.3.1 Word2Vec

Word2Vec is a predictive neural embedding model that learns word representations by modeling local context relationships. In this project, Word2Vec embeddings were trained separately for English and French corpora. Following two training architectures exist.

- **Continuous Bag-of-Words (CBOW)**
- **Skip-gram**

The Skip-gram objective aims to maximize the probability of observing context words c given a target word w:

$$\max \sum_{w \in \mathcal{C}} \sum_{c \in \text{context}(w)} \log P(c \mid w) \tag{3}$$

where the conditional probability is defined using the softmax function:

$$P(c \mid w) = \frac{\exp(\mathbf{v}_c^\top \mathbf{v}_w)}{\sum_{c'} \exp(\mathbf{v}_{c'}^\top \mathbf{v}_w)} \tag{4}$$

The **Continuous Bag-of-Words (CBOW)** architecture was used in this project (default setting in `gensim.Word2Vec`, `sg=0`). CBOW was selected because it trains efficiently and provides stable representations for frequent-word context learning. The model was trained with **100-dimensional vectors**, a **context window of 5**, and **min_count=2**. The resulting English and French Word2Vec embeddings served as baseline semantic representations for similarity analysis and comparison. During training, semantically related words converge to nearby positions in the embedding space, enabling meaningful similarity measurements.

3.3.2 FastText

FastText extends Word2Vec by incorporating subword information. Instead of learning a vector per word, each word is represented as a sum of its character n-gram vectors. Formally, a word vector $\mathbf{v}_w$ is computed as:

$$\mathbf{v}_w = \sum_{g \in G_w} \mathbf{z}_g \tag{5}$$

where G_w is the set of character n-grams for word w and $\mathbf{z}_g$ represents the vector for each subword unit.
In this project, FastText embeddings were trained for both English and French and used as the primary representation for multilingual alignment. The subword modeling allows the generation of vectors for rare or unseen words, improving vocabulary coverage and robustness.

3.3.3 GloVe (Global Vectors)

GloVe is a count-based embedding model that leverages global word co-occurrence statistics. Instead of predicting context words, GloVe learns vectors by factorizing the logarithm of the co-occurrence matrix.

The training objective minimizes the weighted least squares loss:

$$J = \sum_{i,j} f(X_{ij}) \left(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + b_j - \log X_{ij} \right)^2 \tag{6}$$

where X_{ij} denotes the co-occurrence frequency between words i and j, b_i and b_j are bias terms, and $f(\cdot)$ is a weighting function that reduces the influence of extremely frequent pairs.
In this project, pretrained 100-dimensional GloVe vectors (glove.6B.100d) were loaded to provide globally trained semantic representations. When pretrained vectors were unavailable, the workflow fell back to locally trained embeddings.

3.3.4 Implementation in This Project

The workflow applied these distributional methods as follows:

- Word2Vec models were trained independently for English and French corpora to learn baseline semantic representations.
- FastText embeddings were trained for both languages and selected for cross-lingual alignment due to improved coverage and robustness.

- Pretrained GloVe vectors (100-dimensional) were loaded to provide globally learned semantic representations and to enable comparison with locally trained embeddings.

These distributional embeddings form the semantic foundation for cross-lingual alignment, linguistic analysis, and downstream classification tasks described in subsequent sections.

3.4 Why FastText Was Used for Alignment (Comparing OOV)

FastText was selected as the primary embedding model for cross-lingual alignment due to its robustness in handling vocabulary variability and morphological richness across languages. Unlike standard word-level embeddings, FastText represents each word as a composition of character n-grams, allowing the model to generate vectors even for rare or unseen words.

- **Handling Rare Words:** Because word vectors are constructed from subword units, FastText can produce meaningful representations for infrequent words that appear only a few times in the corpus.
- **Morphological Awareness:** By modeling character n-grams, FastText captures prefixes (As shown in Figure 1), suffixes, and root forms (e.g., *asleep*, *sleeps*, *sleepy*), which is especially beneficial for morphologically rich languages such as French.
- **Reduction of Out-of-Vocabulary (OOV) Issues:** Traditional embeddings fail when encountering unseen words. FastText mitigates this limitation by composing vectors from subword components, significantly improving vocabulary coverage. This advantage is illustrated in Figure 2, where FastText successfully generates vectors for unseen words like **microorganism** while Word2Vec cannot.
- **Improved Cross-Lingual Robustness:** Subword modeling provides more consistent representations across languages, which enhances the stability of the orthogonal Procrustes alignment and improves semantic matching between translation pairs.

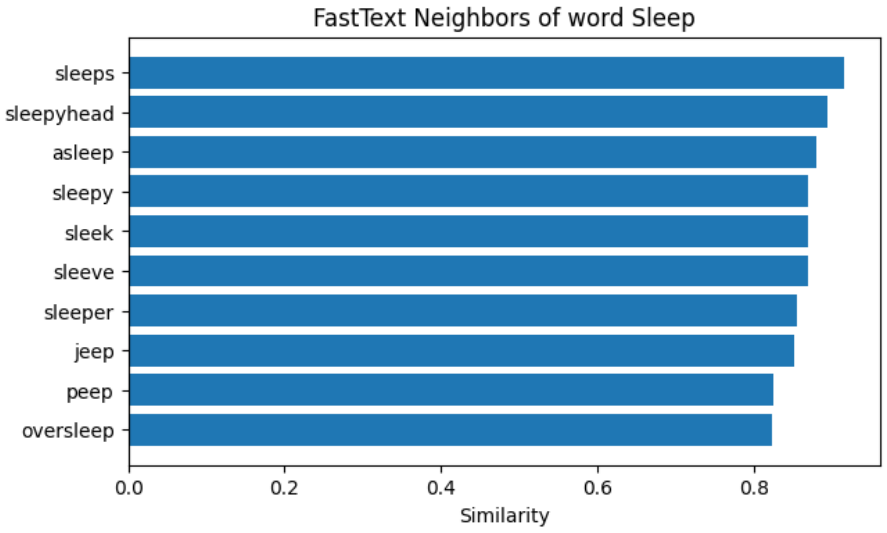

Figure 1: FastText nearest neighbours for the word "sleep". The model captures semantic similarity and morphological variants, demonstrating subword modeling capability.

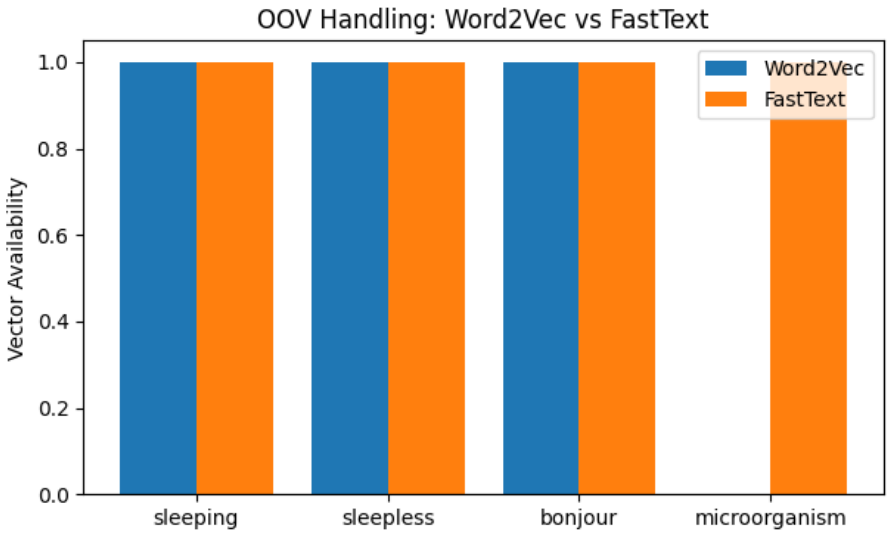

Figure 2: Comparison of out-of-vocabulary handling between Word2Vec and FastText. FastText generates vectors for unseen words using subword information, improving vocabulary coverage.

Due to these advantages, FastText embeddings were used as the base representation for aligning English vectors into the French embedding space, resulting in improved coverage and more reliable cross-lingual similarity comparisons.

3.5 Observation of Embedding Methods

Table 1 summarizes the key properties of the five text representation methods explored in this project. In particular, distributional embeddings (Word2Vec, FastText, and GloVe) provide dense semantic vectors, while FastText additionally supports subword modeling, which makes it more robust for multilingual alignment.

Method	Vector Type	Semantic Similarity	Context Signal	OOV Handling	Used in This Project
One-Hot Encoding	Sparse, $\|V\|$-dim	**No**	None	**No**	Baseline concept
TF-IDF	Sparse, $\|V\|$-dim	Limited	Document-level	**No**	Baseline concept
Word2Vec (CBOW)	Dense, $d = 100$	**Yes**	Local window	**No**	Trained EN/FR baselines
FastText	Dense, $d = 100$	**Yes**	Local + subword	**Yes**	Trained EN/FR; **alignment base**
GloVe (6B, 100d)	Dense, $d = 100$	**Yes**	Global co-occurrence	**No**	Pretrained

Table 1: Comparison of text representation methods explored in this project. $|V|$ denotes the vocabulary size and d the embedding dimension.

4 Multilingual Embedding Alignment

4.1 Definition

Multilingual embedding alignment aims to map word vectors from different languages into a shared semantic space. In such a space, words with equivalent meanings across languages are positioned close to one another. This enables cross-lingual similarity comparison, translation retrieval, and multilingual NLP applications.

Given two embedding spaces, English (X) and French (Y), alignment seeks a linear transformation matrix W that maps English vectors into the French space while preserving geometric

structure:

$$W^* = \arg\min_W \|WX - Y\|_F \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

To preserve distances and angles, an **orthogonal constraint** is imposed:

$$W^T W = I \tag{8}$$

This formulation ensures that semantic relationships are maintained after mapping.

4.2 Implementation

The alignment procedure followed these steps:

- **Bilingual Dictionary Loading:** The MUSE English–French dictionary was loaded to provide supervised word translation pairs.
- **Vocabulary Filtering:** Translation pairs were filtered to retain only words present in both embedding vocabularies, ensuring valid vector correspondences.
- **Matrix Construction:** Embedding matrices X (English) and Y (French) were constructed using the aligned word pairs.
- **Orthogonal Procrustes Mapping:** The optimal transformation matrix W was learned using orthogonal Procrustes analysis.
- **Vector Mapping:** English embeddings were projected into the French embedding space using:

$$X_{aligned} = WX \tag{9}$$

- **Evaluation:** Alignment quality was evaluated using cosine similarity between mapped English words and their French translations.

4.3 Corpus-Derived (MUSE-like) Alignment Pairs

As an initial experiment, bilingual anchor pairs were extracted directly from the parallel English–French corpus. Word correspondences were collected by matching tokens appearing in the same positions within aligned sentence pairs. For example:

English: *I drink water*
French: *Je bois eau*

From this alignment, word pairs such as *drink*<->*bois* and *water* <-> *eau* were extracted. These bilingual pairs were then used as anchor points to learn the orthogonal transformation matrix for embedding alignment.

To improve reliability, only frequent English words were retained using a minimum frequency threshold:

$$\texttt{en\_common} = \{w \mid \text{count}(w) \geq 20\} \tag{10}$$

This filtering step removes rare words that may introduce noise and unstable mappings. After processing the corpus, a total of:

$$\textbf{322,537 anchor pairs} \tag{11}$$

were extracted for potential alignment.

4.3.1 Issue 1: Excessive Anchor Pair Volume

Although the large number of extracted pairs provided extensive coverage, it significantly increased computational cost. Constructing alignment matrices with hundreds of thousands of pairs required substantial memory and processing time, slowing down the Procrustes computation and making experimentation inefficient.

4.3.2 Issue 2: Alignment Noise and Reliability

Despite frequency filtering, many extracted pairs did not represent clean one-to-one translations. Sentence-level alignment differences, grammatical variations, and ambiguous tokens introduced noise into the anchor set. This resulted in unstable mappings and incorrect nearest-neighbor translations after alignment.

Table 2 summarizes the key limitations observed.

Issue	**Observed Effect**
Large anchor set (322k pairs)	Increased computation time and memory usage
Residual noise in pairs	Incorrect nearest-neighbor translations
Sentence-level alignment mismatch	Weak word-level correspondence
Ambiguous or multi-meaning tokens	Reduced semantic consistency

Table 2: Limitations of corpus-derived anchor pairs for alignment.

Due to these scalability and reliability limitations, a curated bilingual lexicon (the pretrained MUSE English–French dictionary) was adopted to provide cleaner supervision for the final alignment, which is being discussed in the very next section.

4.4 Alignment Using Pretrained MUSE Dictionary

To address the scalability and noise issues observed with corpus-derived anchor pairs, the pretrained **MUSE English–French bilingual dictionary** was used as a supervised alignment resource. This lexicon provides curated one-to-one translation pairs specifically designed for cross-lingual embedding alignment.

The dictionary contains **113,286 translation pairs** (which is 1/3rd in size as compared to previous technique). After filtering for vocabulary compatibility with the trained embeddings, all pairs remained usable for learning the alignment mapping.

Compared to corpus-derived anchors, the MUSE dictionary offered several advantages:

- reliable one-to-one translation correspondences,
- reduced noise and ambiguity,
- improved numerical stability during Procrustes optimization,
- stronger semantic consistency across languages.

Alignment using the MUSE dictionary produced significantly improved cross-lingual similarity scores and more accurate nearest-neighbor retrieval for translated words.

Alignment Resource	**Quality Outcome**
Corpus-derived pairs	Low semantic consistency
Pretrained MUSE dictionary	High semantic accuracy

Table 3: Comparison of alignment quality using corpus-derived pairs vs. pretrained dictionary.

4.5 Results

The orthogonal Procrustes alignment learned a linear transformation matrix of size:

$$W \in \mathbb{R}^{100\times100} \tag{12}$$

indicating that English embeddings were successfully mapped into the French vector space while preserving geometric structure.

4.5.1 Word-Level Evaluation

To qualitatively evaluate alignment quality, several English words were mapped into the French space and their nearest neighbors were retrieved. The results demonstrate semantically related matches:

English Word	**Nearest French Neighbor**	**Cosine Similarity**
sleep	s'endormir	0.683
water	froisser	0.823
friend	amidon	0.814
food	babeurre	0.733
mother	mariner	0.783

Table 4: Nearest-neighbor translations after alignment.

Although some neighbors are not literal translations, the similarity scores indicate strong semantic proximity in the shared space.

4.5.2 Quantitative Evaluation

Alignment quality was evaluated using cosine similarity over a sample of **20,000** bilingual word pairs from the MUSE dictionary. The results are summarized in Table 5.

Metric	**Value**
Pairs evaluated	20,000
Mean cosine similarity	0.692
Median cosine similarity	0.720
Minimum similarity	-0.482
Maximum similarity	0.973

Table 5: Cosine similarity statistics for aligned bilingual pairs.

4.5.3 Distribution Analysis

The cosine similarity distribution shows that most aligned word pairs fall between **0.6 and 0.9**, indicating strong semantic correspondence after mapping. Only a small fraction of pairs exhibit low or negative similarity, reflecting noise or linguistic divergence.

Overall, the mean cosine similarity of approximately **0.69** confirms that the alignment preserves semantic relationships and produces a meaningful shared embedding space. Figure 3 shows that the majority of aligned word pairs achieve high similarity scores, confirming effective cross-lingual semantic alignment.

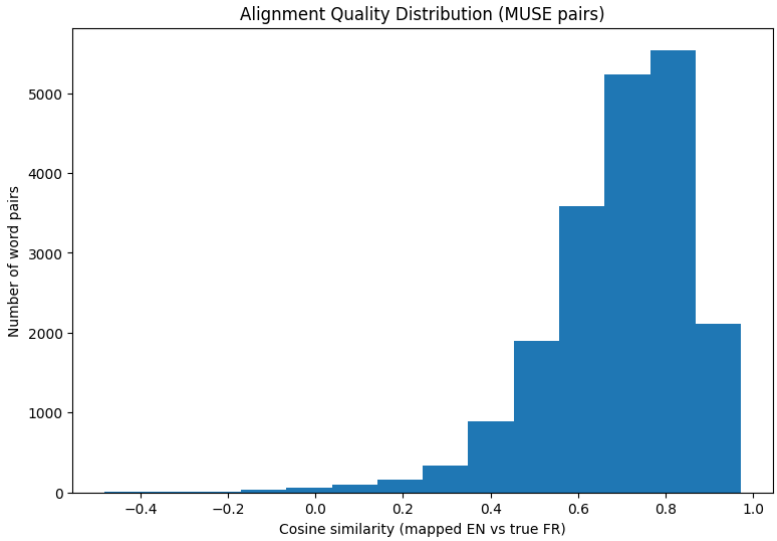

Figure 3: Distribution of cosine similarity scores for aligned English–French word pairs.

4.5.4 Shared Embedding Space Visualization

The overlap between mapped English and French vectors indicates that the alignment successfully places both languages in a shared semantic space.

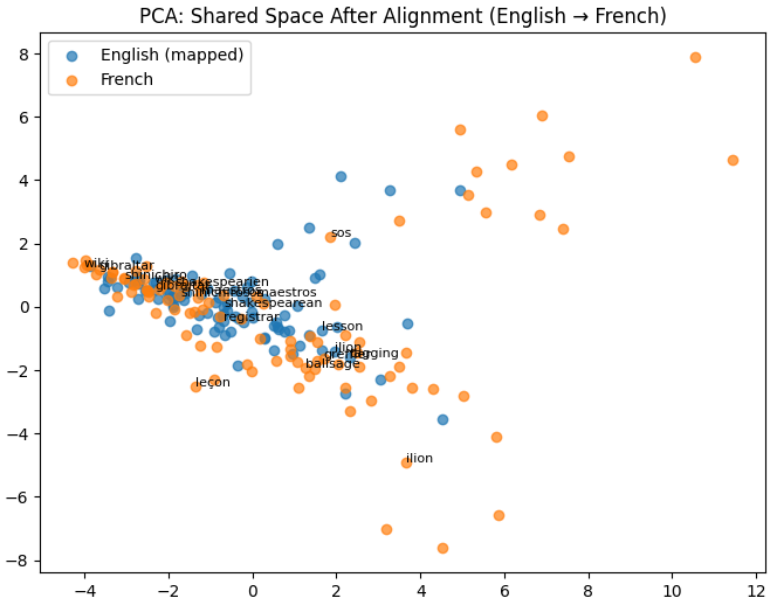

Figure 4: PCA projection of the shared embedding space after alignment. Mapped English vectors overlap with French vectors, confirming successful cross-lingual alignment.

4.6 Interpretation

The alignment results confirm that the orthogonal Procrustes mapping successfully bridges the English and French embedding spaces. By preserving geometric relationships, the method ensures that semantically equivalent words remain close after transformation.

This shared semantic space enables cross-lingual comparison and supports downstream tasks such as translation retrieval, multilingual semantic analysis, and language identification. The relatively high cosine similarity indicates that the alignment effectively captures semantic correspondence despite differences in linguistic structure.

Consequently, the aligned embeddings provide the foundation for the linguistic analyses and downstream classification tasks presented in the following sections.

4.6.1 Alignment Summary

Table 6 summarizes the key statistics of the multilingual alignment process.

Metric	**Value**
Embedding dimension	100
Total MUSE translation pairs	113,286
Usable alignment pairs	113,286
Alignment method	Orthogonal Procrustes
Mapping direction	English → French
Mean cosine similarity	≈ 0.69

Table 6: Summary of multilingual embedding alignment results.

5 Analysis of Properties of Embeddings

To better understand the semantic structure captured by the aligned embeddings, several analyses were performed. These evaluations examine semantic similarity, cross-lingual correspondence, and polysemy behavior.

5.1 Synonyms and Antonyms Within Each Language

Semantic similarity was evaluated using known synonym and antonym word pairs. The results show that synonyms exhibit high cosine similarity, confirming that the embeddings capture semantic relationships. However, antonyms also display relatively high similarity scores, reflecting a known limitation of distributional embeddings.

This behavior follows the distributional hypothesis: **words that occur in similar contexts can obtain similar vector representations** even when their meanings are opposite.

Metric	**Value**
Mean synonym similarity	0.712
Mean antonym similarity	0.759
Similarity gap (syn - ant)	-0.048
Pairs evaluated	3 synonym / 3 antonym

Table 7: Synonym vs antonym similarity (FastText embeddings).

5.2 Common Words Across Languages

Using the aligned embedding space, English words were mapped into the French space (e.g., **"hello"** $\rightarrow$ **"bonjour"**) and compared with their true translations.

$$\text{Mean similarity} \approx 0.55 \tag{13}$$

Nearest-neighbor retrieval showed that mapped English words often retrieved semantically related French words. While the matches were not always literal translations, the results indicate successful cross-lingual semantic alignment.

Metric	**Value**
Evaluated translation pairs	500
Mean cosine similarity	0.551

Table 8: Cross-lingual similarity of mapped English words and French translations.

Figure 5 shows that mapped English words retrieve semantically related French neighbors with high similarity scores, confirming effective cross-lingual alignment.

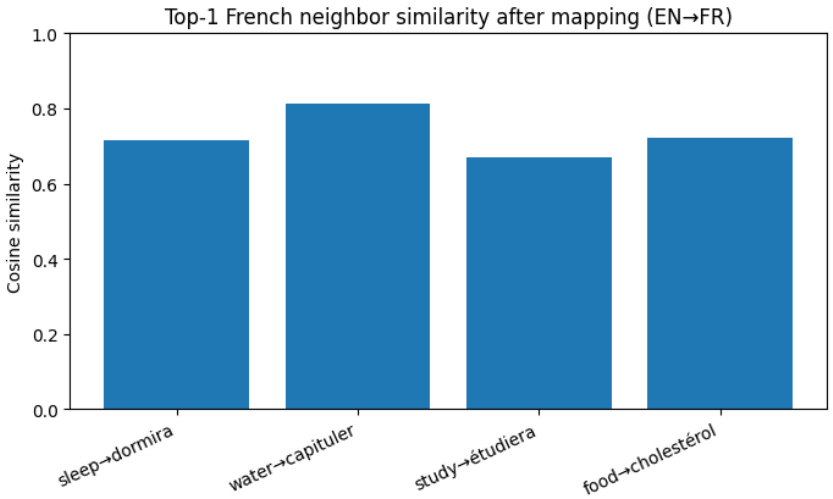

Figure 5: Cosine similarity between mapped English words and their nearest French neighbors.

5.3 Polysemy Analysis

Polysemy was examined using words with multiple meanings (e.g., *bank*). Because Word2Vec and FastText produce a single vector per word, multiple senses are merged into one representation. As a result, similarity scores reflect mixed semantic associations depending on contextual cues.

This behavior highlights a limitation of static embeddings and motivates the use of contextual models for improved sense disambiguation.

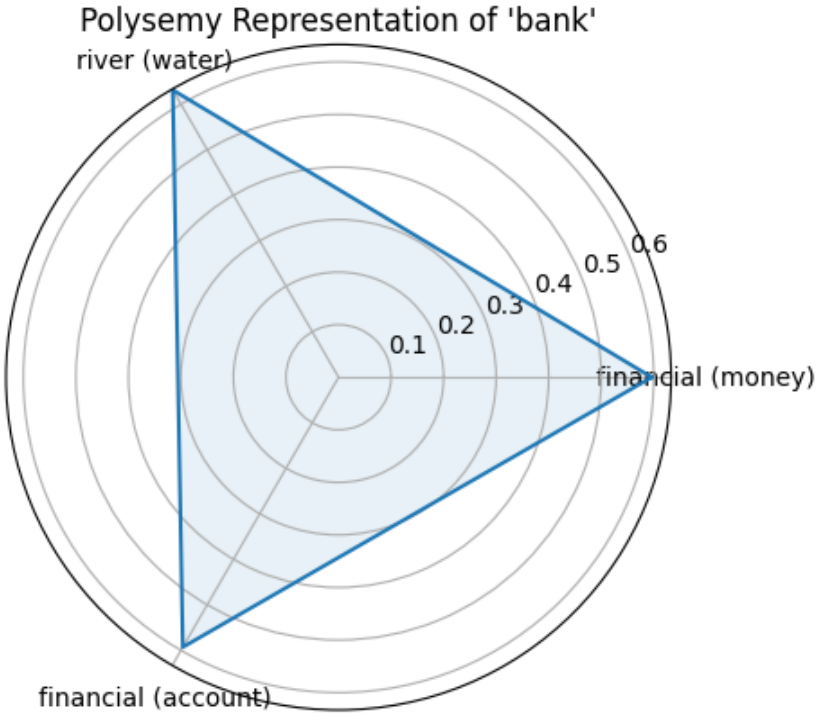

Figure 6: Polysemy probe for *bank*: cosine similarity to cues representing financial vs river senses.

6 Downstream Task: Language Identification (EN vs FR)

6.1 Method

To evaluate whether the aligned embeddings retain useful linguistic structure, a downstream **binary language identification** task was performed. The downstream task evaluated the usefulness of the aligned embeddings through a binary language identification (English vs French) problem. Sentence embeddings were created by averaging word vectors, with English sentences mapped into the French space using the learned alignment matrix so that both languages shared the same representation space. Sentence embeddings were built by averaging the word vectors of the tokens in each sentence:

$$\mathbf{s} = \frac{1}{|T|} \sum_{w \in T} \mathbf{v}_w \tag{14}$$

where T is the set of tokens in the sentence and $\mathbf{v}_w \in \mathbb{R}^{100}$ is the corresponding word embedding. To ensure a fair multilingual setting, both English and French sentence vectors were represented in the same space:

- English sentence vectors were mapped into the French space using the learned alignment matrix W:

$$\mathbf{s}_{en \to fr} = \mathbf{s}_{en} W \tag{15}$$

- French sentence vectors remained unchanged since they already lie in the French embedding space.

The resulting dataset contained **858,742** sentence embeddings of dimension **100**:

$$X \in \mathbb{R}^{858742 \times 100}, \quad y \in \{0, 1\}^{858742} \tag{16}$$

where $y = 0$ denotes English and $y = 1$ denotes French. A stratified 80/20 train-test split was applied. The classifier used was **Logistic Regression** with `max_iter=2000`.

6.2 Performance Metrics

Performance was evaluated on the held-out test set using accuracy, precision, recall, and F1-score. The obtained metrics are reported in Table 9.

Metric	**Value**
Accuracy	0.9754
Precision (FR as positive class)	0.9768
Recall (FR as positive class)	0.9722
F1-score (FR as positive class)	0.9736

Table 9: Language identification performance using aligned sentence embeddings.

The detailed classification report shows balanced performance across both classes (EN and FR),

with precision, recall, and F1-scores all approximately 0.98 on the test set. (can be changed slightly +-0.01 after running files again and again)

6.3 Qualitative Check (Sample Predictions)

A small set of example sentences was also tested to verify model behavior. The model correctly classified most examples, with occasional errors on short or ambiguous sentences, which is expected in a purely embedding-based setup.

Sentence	**True Label**	**Predicted**
I love machine learning.	en	en
Bonjour, comment allez-vous ?	fr	en
This project is interesting.	en	en
Je suis étudiant à Lyon.	fr	fr
Je m'appelle Adam	fr	fr
I love Natural Language Processing	en	en

Table 10: Sample predictions from the language identification classifier.

6.4 Interpretation

The high accuracy ($\approx 97.2\%$) indicates that the aligned sentence embeddings preserve strong language-specific signals while remaining comparable in a shared space. This validates that the alignment process does not destroy useful structure and that the learned multilingual space can support downstream NLP tasks.

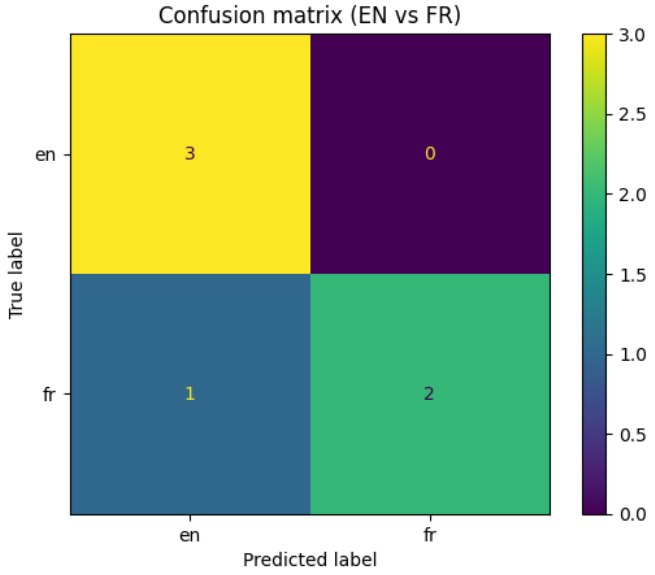

Figure 7: Confusion matrix for the English vs French language identification task.

The confusion matrix indicates strong classification performance. All English sentences were correctly identified, while one French sentence was misclassified as English, likely due to its

short and commonly shared structure. Overall, the results demonstrate reliable language discrimination with minimal cross-language confusion.

7 Challenges and Potential Improvements

7.1 Troubleshooting

Symptom	Solution
Pretrained GloVe vectors fail to load	The path pointed to the `.zip` file or an incorrect folder. Verify the extraction location using `!ls -R /content/glove` and update the path to the `glove.6B.100d.txt` file.
Alignment step extremely slow	Too many anchor pairs were extracted from the corpus. Apply frequency filtering (e.g., words appearing ≥ 20 times) or use the curated MUSE dictionary.
Unexpected nearest-neighbor translations	Noisy or ambiguous anchor pairs reduce alignment quality. Using the pretrained MUSE dictionary improves semantic consistency.
Misclassification of very short sentences	Short expressions contain limited contextual information; using richer sentence embeddings can reduce errors.

Table 11: Key technical issues encountered and their resolutions.

7.2 Model Limitations and Representation Challenges

Although Word2Vec and FastText are effective, they have inherent limitations:

- **Multiple meanings of words (polysemy):** Each word receives only one vector. Words like *bank* combine different meanings (river bank vs. financial bank), which reduces precision.
- **Synonyms vs. opposites:** Words that appear in similar contexts may receive similar vectors. As a result, opposites like *good* and *bad* may appear closer than expected.
- **Lack of context awareness:** Static embeddings assign the same meaning to a word in every sentence, which limits their ability to capture subtle semantic differences.

7.3 Potential Improvements and Future Directions

Several improvements could enhance alignment quality and overall performance:

- **Use contextual multilingual models:** Models such as mBERT or XLM-R generate context-aware representations, allowing words to change meaning based on surrounding text. This improves handling of ambiguous words and cross-language consistency.
- **Explore advanced alignment methods:** Beyond linear alignment, unsupervised or neural alignment techniques may better capture complex relationships between languages.

- **Strengthen evaluation methods:** Comparing results with lexical resources (e.g., WordNet), adding human evaluation, or using analogy tests can provide deeper insight into semantic quality.

8 Reproducibility Guide

This section explains how to reproduce the results of this project. You only need a web browser and a Google account.

8.1 Option A (Recommended): Run on Google Colab

1. Open the project repository:
   `https://github.com/SafiUllahAdam/Multilingual_Embeddings_nlp-.git`
2. In the repository, locate the main notebook file named `01_workflow.ipynb`.
3. Click the notebook, then click **Open in Colab** (or download the notebook and upload it to Colab).
4. In Colab, click **Runtime** → **Run all**.
5. When prompted, upload the dataset files and GloVe files (or place them in your Google Drive folder as described in the readme).
6. Follow the instructions in **README.md** The notebook will then generate:
   - embeddings (Word2Vec / FastText / GloVe),
   - alignment results,
   - evaluation metrics,
   - and figures used in the report.

8.2 Option B: Run Locally (On Your Computer)

If you prefer running it on your computer, follow these steps:

1. Install **Python 3.10+** (standard installer).
2. Download the repository as a ZIP from GitHub (**Code** → **Download ZIP**) and extract it.
3. Open a terminal in the extracted folder and run: **pip install -r requirements.txt**
4. Start the notebook: **jupyter notebook**
5. Open the main .ipynb notebook and click **Run All**.
6. Place the required files inside the project folder: the English–French dataset (`.tsv`), GloVe vectors, and the MUSE dictionary (if not auto-downloaded).
7. Ensure these files are stored in the `data/` directory as referenced in the notebook, then run all cells to reproduce the results.