Palallel sentence mining

Ivan Lakhtin

July 2025

Abstract

We present a method for mining high-quality parallel sentence pairs from large, unaligned bilingual corpora. Our approach is based on multilingual sentence embeddings produced by LaBSE, combined with approximate nearest neighbor search using FAISS. To ensure high precision, we apply a two-stage filtering process: (1) scoring candidate pairs using margin-based similarity, and (2) token-level filtering based on lemmatization and bilingual lexical overlap. We evaluate our method on standard benchmark datasets (Yandex, News Commentary, and CommonCrawl) and compare it to the widely used LASER-based CCMatrix baseline. Experimental results show that LaBSE embeddings, when paired with margin scoring and combined filtering, significantly improve the quality of extracted sentence pairs. Our findings highlight the importance of both strong multilingual representations and careful filtering in building clean parallel corpora for machine translation and cross-lingual NLP tasks.

link to repo: `https://github.com/ialakhtin/paragraph_mining`.

1 Introduction

One of the key tasks in the field of computational linguistics and natural language processing is the automatic alignment of sentences in different languages that are translations of each other. This task is fundamental for the creation of parallel corpora, which are widely used in machine translation, multilingual language model training, linguistic research, and other applications. However, most available parallel corpora are limited in size, domain, and language coverage. At the same time, a vast amount of potentially parallel text exists on the internet and in other sources, yet remains unannotated.

The goal of this project is to develop and implement a method for automatically identifying parallel sentences in an unaligned corpus. An unaligned corpus refers to a collection of texts in two (or more) languages without explicit annotation at the sentence alignment level. The main focus is on identifying sentence pairs that are translational equivalents, without prior knowledge of the document structure or thematic similarity.

This project explores methods for representing sentences in a shared vector space using modern multilingual language models, as well as algorithms for comparing and filtering candidate sentence pairs. The practical value of this work lies in the ability to automatically enrich parallel corpora by discovering new aligned sentence pairs in real-world textual data. This, in turn, contributes to improving the quality of machine translation and other multilingual technologies.

1.1 Team

Ivan Lakhtin made this project

2 Related Work

Automatic identification of parallel or translational sentence pairs from unaligned corpora has attracted substantial research attention over the past decades. Traditional approaches often relied on sentence-length statistics and lexical correspondences, while recent advances leverage multilingual embeddings and neural filtering techniques.

With the advent of multilingual embeddings, there appeared methods like WikiMatrix [Schwenk et al., 2019] that have mined over 135 million parallel sentences across 1,620 language pairs using multilingual sentence embeddings, without any document alignment supervision. Building on similar principles, CCMatrix [Schwenk et al., 2020] scaled this approach using Common Crawl data to extract 4.5 billion high-quality parallel sentences across dozens of languages, achieving state-of-the-art results on WMT translation tasks.

In the most recent developments, [Steingrímsson et al., 2023] introduced SentAlign, a tool that applies LaBSE embeddings to exhaustively score sentence pairs within potentially very large document pairs. Utilizing divide-and-conquer search and localized re-evaluation, SentAlign outperforms several established systems in both alignment accuracy and downstream MT quality.

There are also comparative researches that have benchmarked aligners including BSA, Hunalign, Bleualign, Vecalign, and Bertalign [Forgac F, 2023]. These studies reveal that tools incorporating vector-based similarity consistently outperform those relying solely on length or lexical features.

However, the peculiarity of most of the works is that they work with ready-made pairs of documents. This allows authors to make assumptions about which pairs of sentences are translations based on the pairs already found. But in the problem under consideration, the data does not have such a structure, so most of the results will not be applied.

3 Model Description

My approach is inspired by the large-scale bitext mining pipeline described in CCMatrix [Schwenk et al., 2020] and is adapted to extract high-quality parallel sentence pairs from an unaligned Russian–English corpus. The pipeline includes four main stages: sentence embedding using LaBSE, FAISS-based bidirectional retrieval, similarity filtering, and token-level lexical filtering.

3.1 Sentence Embedding with LaBSE

To obtain semantically meaningful sentence representations, we use the Language-Agnostic BERT Sentence Embedding (LaBSE) model [Feng et al., 2022], which maps multilingual input into a shared embedding space. We consider two configurations of LaBSE:

- **Pretrained LaBSE**: The original publicly available model, used without any additional training.
- **Fine-tuned LaBSE**: The same model further fine-tuned on a domain-specific corpus of parallel news texts, with the goal of adapting the embedding space to the stylistic and topical characteristics of news language.

Fine-tuning was performed using a contrastive loss objective, encouraging embeddings of known parallel sentences to be close while pushing non-parallel ones apart. The structure of the original model is presented on Loss function was taken from the original paper:

$$L = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{\phi(x_i,y_i)-m}}{e^{\phi(x_i,y_i)-m} + \sum_{j\neq i} e^{\phi(x_i,y_j)}},$$

where m is a margin between positive and negative examples. In the original paper $m = 0.3$, therefore I also used $m = 0.3$.

To speed up the training stage and due to GPU memory limitations, we employed LoRA (Low-Rank Adaptation) with parameters $r = 32, \alpha = 32$ to efficiently train the model.

3.2 FAISS-Based Bidirectional Retrieval

To efficiently retrieve candidate sentence pairs, we construct approximate nearest-neighbor indices using FAISS (Johnson et al., 2017):

- A FAISS index is built from the embeddings of Russian sentences, and English sentence embeddings are queried against it.
- The process is repeated in reverse: an index is built from English embeddings, and Russian sentences are used as queries.

This bidirectional search compensates for potential asymmetries in sentence structure and translation length, increasing overall recall and alignment quality.

Since the evaluation dataset contains 700k pairs, it was decided to use an index with the parameters **OPQ32,IVF16384,PQ32** and **nprobe = 32**

3.3 Cosine Similarity Filtering

For each candidate pair retrieved via FAISS, we calculate cosine similarity between their LaBSE embeddings. Only pairs with a similarity score above a defined threshold are retained for further analysis. This step helps filter out semantically unrelated sentence pairs.

3.4 Token-Level Lexical Filtering

To further eliminate semantically similar but non-parallel pairs (e.g., formulaic content or thematic overlaps), we apply a final token-based lexical filtering step:

- **Lemmatization**: Each sentence is tokenized and lemmatized using language-specific tools (pymorphy3 for Russian and nltk for English) to normalize inflectional variation.

- **Dictionary-Based Translation**: Lemmas from Russian sentences are translated into English using a custom-built bilingual dictionary.

- **Overlap Matching**: We calculate the overlap between the set of translated Russian lemmas and the token set of the original English sentence. And vice versa, by translating English lemmas into Russian.

  $$Score = \frac{|T_1 \cap T_2|}{max(|T_1|, |T_2|)}$$

  where T_1, T_2 are token sets of the translated and original sentences.

- **Thresholding**: Only sentence pairs that exceed the overlap threshold in both translation directions are retained. This step significantly reduces false positives, particularly in cases where LaBSE embeddings yield high similarity for topically related but non-equivalent sentences.

4 Dataset

4.1 Dictionary construction for Lexical Filtering

To support lexical verification during sentence pair filtering, we constructed two bilingual lemma-level dictionaries: one from Russian to English and one from English to Russian. The dictionaries were built using frequency-based word selection, lemmatization, and automatic translation, as outlined below.

1. **Lemma Extraction**

   - Russian Lemmas: We took **1.5M** russian words from open-source project. Each word was lemmatized using pymorphy3, a morphological analyzer for Russian. This produced a normalized list of **160k** lemmas representing the core Russian vocabulary.

- English Lemmas: The top **100k** most frequent English words were collected from another open-source project. Lemmatization was performed using NLTK's built-in WordNet lemmatizer. It resulted in **80k** english lemmas.

2. **Automatic Translation** To build bilingual mappings, both lemma lists were translated into the opposite language using **Argos Translate**, an open-source neural machine translation system. Translations were performed in both directions:

    - Russian lemmas → English
    - English lemmas → Russian

    No human curation was applied at this stage; translations were taken as-is to maximize coverage and maintain speed.

3. Dictionary Structure As a result, we obtained:

    - A Russian-to-English lemma dictionary: mapping each Russian lemma to one or more English equivalents.
    - An English-to-Russian lemma dictionary: mapping each English lemma to Russian counterparts.
    - Dictionaries from words to lemmas to speed up the lemmatization stage

These dictionaries were used in the token-level lexical filtering stage (see Section 4 of Methodology) to estimate lexical overlap between sentences across languages.

4.2 Collected datasets

To support both evaluation and model fine-tuning, we use two types of parallel corpora, each selected for a specific role in the project:

- **Evaluation Dataset** To evaluate the quality of sentence alignment and parallel sentence mining, we required a dataset with high-quality, human-translated sentence pairs. For this purpose, we combined the Yandex corpus and the News Commentary dataset from WMT19. Both sources are widely used in machine translation research and contain professionally aligned bilingual content, which makes them suitable for precision-focused evaluation.

- **Fine-tuning Dataset** For domain adaptation of the LaBSE model, we used a large-scale dataset from the Common Crawl collection, also provided as part of WMT19. This corpus offers wide coverage of topics and linguistic variation, but contains noise and redundancy, so we applied a rigorous preprocessing pipeline before using it for fine-tuning.

- **ParaCrawl** We additionally examined ParaCrawl, a large-scale automatically aligned web corpus. But, atrer running preprocessing pipeline, we rejected this corpus because of high drop rate

4.3 Preprocessing Pipeline

Both datasets were preprocessed using the same multi-stage normalization and filtering pipeline designed to improve sentence quality and remove noise. The following steps were applied:

1. **Normalization**: Lowercasing, character unification (quotes, brackets, dashes), and punctuation deduplication.
2. **Character Validity Check**: Filtering out sentences with invalid or non-linguistic symbols.
3. **Language Verification**: Automatic language detection using fast-langdetect python module to ensure each sentence is in the correct language.
4. **Numerical Consistency**: Ensuring that numerical expressions match between source and target.
5. **Deduplication**: Removal of exact and near-duplicate pairs.
6. **Outlier Removal**: Removal of the top 1% longest sentences to reduce noise and segmentation artifacts.

This preprocessing pipeline ensured that both the evaluation set and the fine-tuning data were clean, linguistically consistent, and suitable for high-precision modeling and assessment. In the Tab. 1 you can see the result of running this pipeline on all source datasets.

Dataset	Purpose	Initial Size	% Removed
News Commentary	Evaluation	281,145	16.0%
Yandex	Evaluation	1,000,000	14.5%
Common Crawl	Fine-tuning	878,317	36.3%
ParaCrawl	Considered (rejected)	1,000,000 (sample)	57.8%

Table 1: Statistics of the preprocessing of the source datasets. There were too much bad pairs in ParaCrawl, so it was rejected

5 Experiments

5.1 Metrics

To assess the quality of the proposed parallel sentence mining approach, we constructed a custom evaluation dataset based on the preprocessed test splits of the Yandex and News Commentary corpora. The dataset was designed to simulate a realistic and partially aligned bilingual corpus, in which only a subset of the sentences forms valid translation pairs.

5.1.1 Evaluation Dataset Construction

The dataset was built using the following procedure:

1. **Positive Pairs**: One-third of the original aligned sentence pairs were kept intact, i.e., both the Russian and English sentences were included.
2. **Russian-Only Samples**: From another third of the dataset, only the Russian side was retained; the corresponding English sentences were discarded.
3. **English-Only Samples**: From the remaining third, only the English side was kept, discarding the Russian counterparts.

As a result:

- Two-thirds of the Russian and English sentences were preserved.
- Only half of these sentences had a valid translation pair in the dataset.

Both the Russian and English subsets were randomly shuffled to avoid any implicit ordering or alignment cues.

The final evaluation task simulates a real-world scenario in which a parallel mining system must extract true translation pairs from two unaligned corpora with partial overlap.

5.1.2 Evaluation Procedure

The model receives two inputs:

- A list of Russian sentences
- A list of English sentences

It is expected to return a list of predicted parallel sentence pairs. These predictions are then compared against the ground-truth list of aligned pairs constructed during dataset generation.

5.1.3 Evaluation Metrics

We evaluate model performance using the following standard metrics:

- Precision (P) – the proportion of predicted sentence pairs that are correct.
- Recall (R) – the proportion of actual translation pairs that were successfully identified.

- F-β-Score – the harmonic mean of precision and recall with a weighting factor β.

Since our goal is to maximize the quality of extracted sentence pairs rather than their quantity, we emphasize precision over recall. Accordingly, we use the F-0.5-score, which weights precision higher than recall:

$$F_{0.5} = (1 + 0.5^2)\frac{Precision * Recall}{0.5^2 Precision + Recall}$$

This metric better reflects our objective of producing a clean and reliable parallel corpus.

5.2 Baselines

The most straightforward solution to the parallel sentence mining problem is an exhaustive pairwise comparison between all source and target sentences, followed by the application of filtering criteria such as LaBSE score thresholds, token-level overlap, or other similarity metrics. While conceptually simple, this approach is computationally infeasible for large-scale corpora due to its quadratic time complexity.

Consequently, in any realistic setting involving millions of sentences, more efficient approximate search methods must be employed — such as nearest neighbor search over vector embeddings using FAISS or similar indexing libraries.

Therefore, for experimental evaluation, we adopt as our baseline the sentence mining pipeline described in the **CCMatrix paper**, which uses:

- LASER[Schwenk and Douze, 2017] embeddings
- Margin-based similarity scoring

$$margin(x, y) = \frac{2k(x, y)}{\sum_{x_i \in NN(y,k)}(x_i, y) + \sum_{y_i \in NN(x,k)}(x, y_i)}$$

- FAISS-based retrieval

To calculate the margin, they simultaneously search for the best pair for each sentence and the k nearest neighbors (Fig. 1).

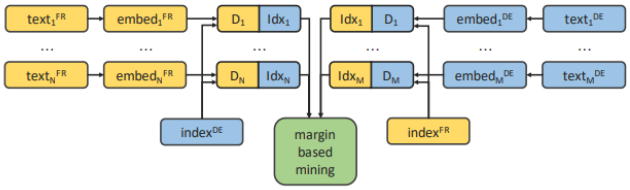

Figure 1: Margin-based score

5.3 Experiment Setup

5.3.1 Hardware Configuration

- GPU: NVIDIA RTX 4090 (24GB)
- CPU: 4 cores
- RAM: 16GB

5.3.2 Comparison of Embedding Models and Scoring Functions

In the first experiment, we compared three different models for generating sentence embeddings:

- LASER – multilingual encoder from the original CCMatrix paper.
- LaBSE – pre-trained multilingual model developed by Google.
- Fine-tuned LaBSE – our version of LaBSE adapted to the news domain using the Common Crawl dataset.

For similarity scoring, we applied the following approaches:

- For LASER, we used the **margin-based** score as defined in CCMatrix.

  $$margin(x, y) = \frac{2k(x, y)}{\sum_{x_i \in NN(y,k)}(x_i, y) + \sum_{y_i \in NN(x,k)}(x, y_i)}$$

  , where k=const, $NN(x, k)$ - k nearest neigbours to the x according to the index.

- For LaBSE and Fine-tuned LaBSE, we evaluated two variants:
  - Direct **cosine similarity**
  - **Margin-based** score (same as LASER)

We applied the same filtering parameters across all configurations:

- Token-level filtering threshold: **0.1**
- Score threshold:
  - Without margin: **0.7**
  - With margin: **1.06** (as in the original CCMatrix setup)

This experiment was designed to identify the best model and scoring method in terms of precision and recall on the constructed evaluation dataset.

5.3.3 Effectiveness of Filtering Techniques

Using the best-performing model from Experiment 1, we examined the impact of different filtering strategies on the quality of the extracted sentence pairs. Four configurations were tested:

- **No Filtering** – all candidate pairs returned by the nearest neighbor search.
- **Score-based Filtering Only** – filtering based on the model similarity score.
- **Token-level Filtering Only** – filtering using lexical overlap based on the bilingual lemma dictionary.
- **Combined Filtering** – both model score and token-level filtering applied.

5.3.4 Final Comparison Against LASER Baseline

In the final experiment, we compared the performance of our best model and full filtering pipeline against the LASER-based system without token-level filtering, which reflects the setup described in the original CCMatrix work.

6 Results

6.1 Comparison of Embedding Models and Scoring Functions

The results of the first experiment (see Tab. 6.1) reveal several important insights regarding the behavior of different sentence embedding models and scoring functions:

1. **Fine-tuning LaBSE did not improve performance**. In fact, the fine-tuned model underperformed compared to the pre-trained version. We attribute this to the insufficient number of negative examples during training. While the original LaBSE paper used more than 1,000 negatives per anchor, our setup was limited to just 48 negatives due to GPU memory constraints. This limitation significantly reduced the contrastive signal needed for effective fine-tuning.

2. **Margin-based scoring consistently outperforms raw cosine similarity**, both for LaBSE and Fine-tuned LaBSE. This aligns with the findings of the CCMatrix paper and confirms the benefit of relative scoring that accounts for local density in the embedding space.

3. **Pre-trained LaBSE outperforms LASER**, demonstrating better precision and $F_{0.5}$ score under both scoring regimes. This suggests that newer multilingual encoders like LaBSE are more suitable for high-precision parallel sentence extraction tasks.

These results guided our decision to use LaBSE with margin-based scoring as the default embedding model in subsequent experiments.

Model	**Precision**	**Recall**	$\mathbf{F_{0.5}}$
LASER (margin)	0.890	0.858	0.883
LaBSE (cosine)	0.881	0.773	0.857
LaBSE (margin)	0.946	**0.888**	**0.934**
Fine-tuned LaBSE (cosine)	0.179	0.500	0.205
Fine-tuned LaBSE (margin)	**0.991**	0.040	0.171

Table 2: Comparison of embedding models and scoring methods. LaBSE with margin scoring achieves the best balance between precision and recall.

6.2 Impact of Filtering Methods

As shown in Table 6.2, combining both filtering strategies — model score thresholding and token-level lexical overlap — significantly improves overall alignment quality. Each method individually addresses different types of noise in the candidate pairs:

- Score-based filtering helps remove semantically unrelated sentence pairs with low LaBSE similarity.
- Token-level filtering, based on bilingual lemma translation and word overlap, reduces spurious matches that may have high embedding similarity but diverge in lexical meaning (e.g., due to LaBSE collision errors).

Filtering Strategy	**Precision**	**Recall**	$\mathbf{F_{0.5}}$
No Filtering	0.306	**0.941**	0.354
Score Only	0.931	0.903	0.926
Token-Level Only	0.437	0.923	0.488
Score + Token-Level	**0.946**	0.888	**0.934**

Table 3: Effect of different filtering strategies using the best embedding model. Combined filtering improves both precision and $\mathrm{F}_{0.5}$ score.

The final comparison confirms that our full pipeline — based on LaBSE with margin scoring and combined filtering — outperforms the LASER-based baseline in terms of both precision and $F_{0.5}$ score. While LASER offers acceptable recall (see Tab. 6.3), its lower precision leads to a larger number of false positives. This underscores the importance of using stronger embeddings and multi-stage filtering when high-quality parallel sentence extraction is the goal.

Model	**Precision**	**Recall**	**$F_{0.5}$**
LASER (baseline)	0.854	0.874	0.858
Ours (Fine-tuned LaBSE + filtering)	**0.946**	**0.888**	**0.934**

Table 4: Final comparison between the proposed pipeline and the LASER-based baseline. Our approach shows improved precision and $F_{0.5}$.

6.3 Final Comparison Against LASER Baseline

7 Conclusion

In this project, we developed and evaluated a pipeline for mining parallel sentence pairs from raw, noisy bilingual corpora. Our approach leverages multilingual sentence embeddings (LaBSE), efficient approximate search via FAISS, and a multi-stage filtering system that combines semantic similarity with lexical verification.

The key findings are as follows:

- LaBSE outperforms LASER in both precision and $F_{0.5}$ score, making it better suited for high-precision sentence alignment.
- Margin-based scoring provides a consistent performance boost over raw cosine similarity, confirming prior results from CCMatrix.
- Fine-tuning LaBSE did not improve results in our setting, likely due to insufficient negative samples per batch caused by hardware constraints.
- Combining model score filtering with token-level lexical filtering is crucial to eliminate false positives, including semantically unrelated pairs and embedding collisions.

Overall, the pipeline achieves robust performance on realistic datasets and can be scaled to large corpora using FAISS. Future work may focus on improving fine-tuning efficiency, integrating language-specific heuristics, and exploring transformer-based reranking of candidate pairs.

References

[Feng et al., 2022] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2022). Language-agnostic bert sentence embedding. -.

[Forgac F, 2023] Forgac F, Munkova D, M. M. K. L. (2023). Evaluating automatic sentence alignment approaches on english-slovak sentences. -.

[Schwenk et al., 2019] Schwenk, H., Chaudhary, V., Sun, S., Gong, H., and Guzmán, F. (2019). Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. -.

[Schwenk and Douze, 2017] Schwenk, H. and Douze, M. (2017). Learning joint multilingual sentence representations with neural machine translation.

[Schwenk et al., 2020] Schwenk, H., Wenzek, G., Edunov, S., Grave, E., and Joulin, A. (2020). Ccmatrix: Mining billions of high-quality parallel sentences on the web. -.

[Steingrímsson et al., 2023] Steingrímsson, S., Loftsson, H., and Way, A. (2023). Sentalign: Accurate and scalable sentence alignment. -.