Using Subword-Embeddings for Bilingual Lexicon Induction in Bantu Languages

Adrian Breiding
adrian.johannes.breiding.1@hu-berlin.de

Alan Akbik
alan.akbik@hu-berlin.de

Abstract

Bilingual Lexicon Induction (BLI) is a valuable tool in machine translation and cross-lingual transfer learning, but it remains challenging for agglutinative and low-resource languages. In this work, we investigate the use of weighted sub-word embeddings in BLI for agglutinative languages. We further evaluate a graph-matching and Procrustes-based BLI approach on two Bantu languages, assessing its effectiveness in a previously underexplored language family. Our results for Swahili with an average P@1 score of 51.84% for a 3000 word dictionary demonstrate the success of the approach for Bantu languages. Weighted sub-word embeddings perform competitively on Swahili and outperform word embeddings in our experiments with Zulu.

1 Introduction

Bilingual Lexicon Induction (BLI) is the task of automatically generating translation pairs from monolingual corpora, typically using a small seed dictionary to align two separate semantic spaces. For an example of a generated output, see Figure 1. These lexicons serve as a critical bridge for many NLP tasks, such as providing translations for out-of-vocabulary words in Machine Translation (MT) (Irvine and Callison-Burch, 2017) and facilitating cross-lingual transfer learning for under-represented languages (Wang et al., 2022).

Despite its potential, current BLI research is constrained by a significant "resource and relatedness" bias. Most work focuses on high-resource, closely related language pairs where the underlying corpora are drawn from similar domains. Furthermore, evaluations claiming to address low-resource settings often do so by down-sampling massive datasets (Marchisio et al., 2022), rather than confronting the noise and sparsity of authentic low-resource environments. While recent efforts have begun to bridge this gap (Nakashole, 2019; Bhowmik and Ralescu, 2023), African languages, specifically the Bantu language family, remain almost absent from the literature.

Swahili–English Dictionary

kununua	V.	buy
cheti	N.	certificate
vyeti	N.PL.	certificates
mjadala	N.	discussion
mijadala	N.PL.	dialogue (*discussions*)
takriban	ADV.	nearly
kusini	N.	south
		⋮

p. 42

Figure 1: Excerpt from a generated Swahili–English bilingual dictionary.

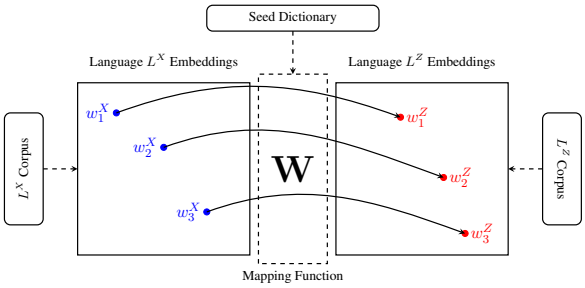

Figure 2: Schematic visualization of bilingual lexicon induction using embeddings

A more fundamental limitation, however, lies in the architectural assumptions of existing BLI frameworks. Nearly all current approaches are built on word-level embeddings, which treat words as indivisible, atomic units. This paradigm assumes a rough one-to-one correspondence between words in different languages. Additionally, approaches often assume the existince of a linear mapping between the embedding spaces of the two languages, a premise known as the isomorphism hypothesis (Søgaard et al., 2018). See Figure 2 for an example. While this may hold for morphologically simple or related languages (e.g., English and Spanish), it breaks down when applied to the agglutinative

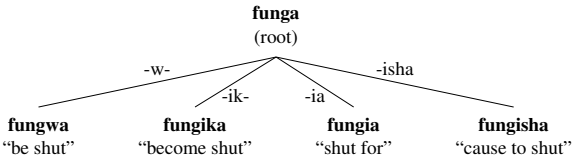

Figure 3: Nuances of In- and Suffixes in Swahili (Encyclopaedia Britannica Editors, 2025)

morphology of Bantu languages.

Challenges of Agglutinative Morphology. In Bantu languages, a single verb root can generate several unique surface forms through the addition of noun-class prefixes, tense markers, and derivational suffixes (see Figure 3). When BLI is performed at the word level, this leads to two critical failures:

- Vocabulary Fragmentation: The semantic meaning of a single root is scattered across dozens of unique "atoms" (e.g., ninauza, utauza, walituuza), each appearing with low frequency. This prevents the model from learning a cohesive representation for the core concept ("to sell"). See Figure 1 for an example.

- Structural Misalignment: A single complex word in Swahili often encodes the semantic equivalent of an entire English phrase. This creates a geometric divergence between embedding spaces, making linear mapping techniques like the orthogonal Procrustes approach mathematically insufficient.

In this paper, we address these challenges by investigating BLI for two representative Bantu languages: Swahili and Zulu. We leverage the `Fundus` library (Dallabetta et al., 2024) to build specialized newspaper-based corpora and generate seed dictionaries via `Google Translate`. To move beyond the limitations of word-level models, we propose and evaluate the use of sub-word level embeddings.

We further test a hybrid approach that combines graph-matching and Procrustes-based mapping (Marchisio et al., 2022), providing, to our knowledge, the first rigorous evaluation of this system on Bantu-English pairs. Finally, we analyze the degree of isomorphy between these spaces to quantify the difficulty of the Bantu-English BLI task.

Contributions. Our main contributions are:

- Identifying and demonstrating the failure modes of word-level BLI when applied to the agglutinative structures of the Bantu language family.

- Validating that sub-word embeddings mitigate these failures by capturing morphological nuances and improving alignment with English.

- Providing a benchmark for BLI on two low-resource African languages using authentic, non-simulated datasets.

2 Related Work

Harris (1954) theorizes that words in similar contexts often have similar meanings. Following this intuition, Mikolov et al. (2013b) presented a method, where a single-layer feed-forward neural network is used to predict a word given its context, and the embeddings can be extracted from the weights of the hidden layer. In their experiments, they found that these embeddings capture some semantic information by computing “Madrid” - “Spain” + “France” and extracting the nearest neighbor, which yields “Paris”.

Experiments with Korean and Swahili, both (highly) agglutinative languages, show a higher performance when using syllable-aware embeddings (Shikali et al., 2019; Choi et al., 2017). In both settings, the researchers generated trained syllable vectors in combination with a convolutional neural network. Since these approaches require more resources, we focused on an alternative: Byte-Pair embeddings, which have the advantage of requiring a lower amount of resources compared to other subword unit embedding approaches (Sennrich et al., 2016; Heinzerling and Strube, 2018). It is based on Byte-Pair encoding (Gage, 1994), where a new symbol iteratively replaces the most common symbol pair. After encoding, the symbols should represent the most common character strings. Training embeddings on these symbols will result in them reflecting the most frequent subwords in the hope of corresponding to semantically significant morphemes. By varying the number of merge operations, the resulting embeddings can be manipulated to represent shorter character sequences or many frequently occurring words primarily.

2.1 The Procrustes Approach

Mikolov et al. (2013a) presented the original idea that later led to the Procrustes method after the

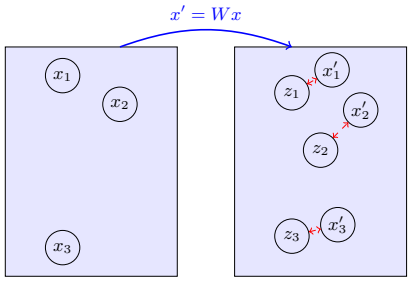

Figure 4: Visualization of the Procrustes approach. Adapted from Marchisio et al. (2021)

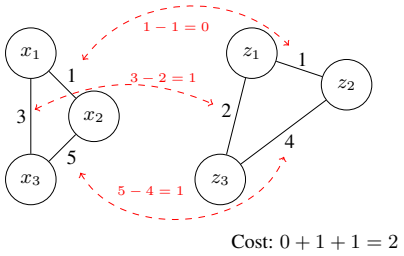

Figure 5: Visualization of the graph-based approach. Adapted from Marchisio et al. (2021)

discovery that embeddings can capture linguistic regularities. Hence, postulating a mapping from the source embedding space to the target embedding space seemed reasonable. Concretely, given a set of word pairs and their corresponding embeddings $\{x_i, z_i\}_{i=0}^{n-1}$ with $x_i, z_i \in \mathbb{R}^d$, the goal is to find a matrix $W \in \mathbb{R}^{d\times d}$, such that Wx_i is a good approximation of z_i. This can be represented as the following optimization problem:

$$\arg\min_{W\in\mathbb{R}^{d\times d}} \sum_{i=0}^{n-1} \|Wx_i - z_i\|_2^2 \tag{1}$$

Restricting $W \in O(d)$ to be orthogonal and replacing the Euclidean norm with the Frobenius norm yields the orthogonal Procrustes problem, which is exactly solvable (Conneau et al., 2018). Having found a mapping between the embedding spaces, we can now use it to predict translations. In the original paper, this was done by extracting the nearest neighbors using the cosine similarity metric. In higher-dimensional spaces, it can be observed that so-called *hubs* and *anti-hubs* are formed. These hubs are the nearest neighbors for many other points with a high probability, whereas anti-hubs are not the nearest neighbors for any other point (Radovanović et al., 2010). Naturally, this causes issues when extracting potential translation candidates using a standard nearest-neighbor measure. To produce a more reliable matching, Conneau et al. (2018) introduce the Cross-Domain Similarity Local Scaling Measure, which intuitively combats hubness by penalizing points with dense neighborhoods while boosting the similarity values for anti-hubs.

The underlying assumption is that the embedding spaces are "isomorphic", which is understood to be "geometrically similar" in this context. While it is reasonable for related languages, experiments have shown that it is less sensible in cases where the language similarity decreases or in cases of varying domains of the monolingual training data sets (Søgaard et al., 2018; Patra et al., 2019).

2.2 The Graph-Based Approach

To attempt BLI using graph matching (Marchisio et al., 2022), one can consider words as nodes in monolingual, weighted, undirected graphs $G_X = (V_X, E_X, w_X), G_Z = (V_Z, E_Z, w_Z)$. The edge weights are computed using the cosine similarity metric. The fundamental idea is that relationships between words are reflected in the similarity of the corresponding embeddings, and these similarities remain relatively consistent across different languages. With this assumption, one can try to find the optimal permutation π aligning $\pi(V_Z)$ with V_X, minimizing the edge disagreement.

To quantify the edge disagreement of the two graphs, it is intuitive to choose a metric similar to:

$$\|A_X - PA_ZP^T\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}[(A_X)_{ij} - (PA_ZP^T)_{ij}]^2} \tag{2}$$

also known as the Frobenius norm. The problem is NP-hard, but functional approximations are available. For one such approximation, the optimization of equation 2 can be attempted by using an instance of the Optimal Transport (OT) Problem

$$\arg\min_{P\in U(r,c)} \operatorname{tr}(P^\top M) \tag{3}$$

with $U(r, c)$ the transportation polytope and a cost matrix $M \in \mathbb{R}^{n\times n}$. A well-performing algorithm for solving the OT problem in graph matching, also in large, non-isometric cases, is the *Graph Matching via OptimAl Transport* (GOAT) algorithm (Saad-Eldin et al., 2021).

The performance of this approach and the translation matrix method increases with the number of translation pairs, so-called seeds, that are passed

into the algorithm. Experiments with various languages and varying numbers of seeds show that graph matching outperforms the nearest-neighbor method in most cases. The performance is further robust on dissimilar languages and low supervision (Marchisio et al., 2022).

2.3 Prior Work on Bantu Languages

Nakashole (2019) attempts BLI for Bantu languages by exploiting grammatical similarities. For one Bantu language L_1 with a small bilingual dictionary, they modify the corpus of a second Bantu language L_2 and merge the corpus of L_1 with the modified corpus of L_2. They then try to find a projection matrix that maps the English embeddings to the combined Bantu word embeddings as introduced by Mikolov et al. (2013a), while only requiring the seed dictionary for L_1. This approach scores $0.30, 0.56$ and 0.58 in precision at top-k ($P@k$) for $k \in \{1, 5, 10\}$ respectively for L_1 and $0.10, 0.18$ and 0.20 for L_2.

2.4 Reliability of `Google Translate`

There is little work available on the current reliability of `Google Translate` for English-Swahili and English-Zulu translations. For Swahili, Okafor (2025) has reported significant improvement potential for `Google Translate` for text translation in the medical domain. Concretely, AfromT (Iyamu, 2024), a domain-specific translation framework for African languages, improves performance on scientific and medical texts by $\sim$ 19 % relative to `Google Translate`. However, AfromT remains $\sim$ 25 % below the performance of comparable models trained on high-resource languages. A qualitative study by Sangili (2024), based on a randomly selected sample, reports "excellent" performance on individual lexemes but notes a marked decline in translation quality for multiword phrases and figurative usages; the author does not report a formal sample size for that claim.

For Zulu, Khoboko et al. (2025) have assessed `Google Translate` using the `BLEU` (Papineni et al., 2002), `G-Eval` (Liu et al., 2023) and `ChFr++` (Yu et al., 2021) metrics. They report scores of 5.93, 55.12 and 95.6 % respectively. The low score for `BLEU`, an n-gram overlap metric, indicates a low surface-level agreement with reference translations. `G-Eval` and `ChFr++` attempt to capture meaning preservation on different scales, suggesting that `Google Translate` often preserves semantics.

Taken together, these findings suggest that Google Translate is sufficient for generating the limited set of common-word seeds required in our experiments.

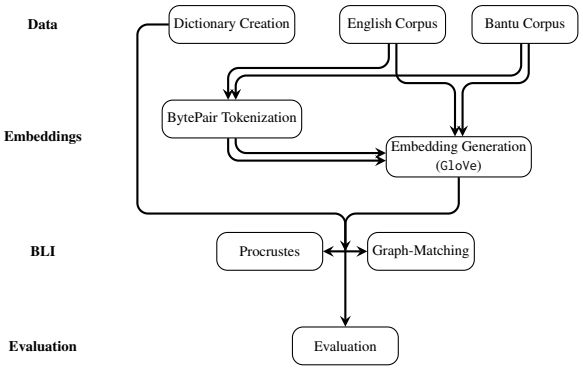

Figure 6: Overview of the four elemental steps in creating a dictionary using BLI.

3 Methodology

The process is split into four steps: data attainment, embedding training, BLI, and evaluation. Figure 6 provides an overview.

3.1 Data Acquisition

One of the main criteria for a possible data source in the selection process was its bilingual availability, since using two corpora from varying domains could drastically reduce the performance (Søgaard et al., 2018). Newspapers from a bilingual publisher may satisfy this constraint. Using the `Fundus` library (Dallabetta et al., 2024) we crawled Daily News[1] and Habari Leo[2] for Swahili - English and Eyethu News[3] (only Zulu articles), Ilanga News[4] and The Citizen[5] for Zulu - English. The plaintexts of the articles are combined into a single file, converted to lower-case, stripped of non-letter characters, and all numbers replaced with zeroes.

This process generates a Swahili - English corpus with 194 605 and 589 460 sentences respectively and a Zulu - English corpus with 146 967 each. We also translate the Zulu corpus with `Google Translate`, resulting in the "translated" corpus.

Both the Procrustes and the graph-matching algorithm require seed translations as input. The 10 000 most frequent words in each corpus are compiled as a list, translated using `Google Translate` and

[1] https://dailynews.co.tz
[2] https://habarileo.co.tz
[3] https://eyethunews.co.za
[4] https://ilanganews.co.za
[5] https://citizen.co.za

combined into a unidirectional dictionary for each language pair. This dictionary can be utilized as input seeds and for automatic evaluation.

In principle, reliance on `Google Translate` does not limit the applicability of this approach to additional low-resource languages, as it only requires a small number of seeds for competitive results. These amounts can be easily obtained from a single native speaker or a small existing dictionary.

3.2 Embeddings

A natural choice for a type of embeddings are subword embeddings, for intuitively, they have the potential to better reflect relationships between words like *andika* (write) and *andik**ia*** (write **to**) (Mpiranya, 2023). For this work, we use Byte Pair embeddings (Sennrich et al., 2016), which are trained using the training scripts provided in the GitHub repository[6] corresponding to the publication (Heinzerling and Strube, 2018). In the first step, the corpus is tokenized and encoded using `SentencePiece` (Kudo and Richardson, 2018). The embeddings are then trained on the encoded corpora using `GloVe` (Pennington et al., 2014). In the final step, the embeddings must be re-arranged in the same order as in the BPE vocabulary file generated in the tokenization phase.

Both of the approaches require word embeddings. A given word is encoded using the learned BPE model. The corresponding embedding is computed by calculating the (weighted) arithmetic mean of the subword embeddings. Given a word w^X with a tokenization $(t_1, \cdots, t_n)$. Let τ_i be the embedding corresponding to the token t_i, then the mean word embeddings are computed by:

$$x = \frac{1}{n} \sum_{i=1}^{n} \tau_i \tag{4}$$

The weighted mean embeddings are calculated using the token length $|t_i|$ as weights.

$$x = \sum_{i=1}^{n} \frac{|t_i|}{\sum_{j=1}^{n} |t_j|} \tau_i \tag{5}$$

3.3 Bilingual Lexicon Induction

Generally, Marchisio et al. (2022) have shown that the best results are achieved using a combination of Procrustes and graph matching. Choosing the correct algorithm to start with (heavily) depends on the number of seeds used. Given the moderate amount of seeds available, starting with graph matching has proven to be the better choice, which is why the following introduction will follow this order.

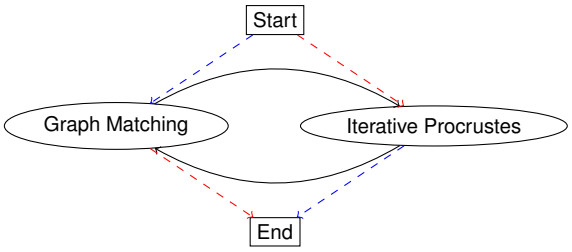

Figure 7: Overview of the system combination experimental setup. Adapted from Marchisio et al. (2022)

First, graph matching is run in the forward and reverse directions. For each direction a hypothesized mapping $h_i : \{1, n\} \rightarrow \{1, n\}, i \in \{f, b\}$ is returned. The indices f, b represent forward and backward, respectively. The hypothesis sets for each direction can then be defined as $H_f = \{(a, b)|h_f(a) = b\}$ and $H_b = \{(a, b)|h_b(b) = a\}$. The intersection $H = H_f \cap H_b$ is used as an input for Procrustes.

Similarly, the Procrustes problem is solved in both directions using the gold seeds and the hypotheses from the graph-matching approach. The hypotheses are extracted using 1-nearest neighbors. Instead of the commonly used cosine metric, we will use Cross-Domain Similarity Local Scaling (Conneau et al., 2018) because it is more resistant against hubness. Multiple iterations of Procrustes are run in total, with the intersections of forward and backward hypotheses from the previous round used as additional seeds. After five iterations, the hypothesis intersection is passed into the graph-matching algorithm.

After 20 iterations, H_f is returned as the final hypothesis and can be used to generate the dictionary.

4 Results

The experiment's goal was to create a dictionary for the 3000 most frequent words in the English and Bantu corpora, assuming that occurrences of a word and its translation are of a similar magnitude. Most parameters relevant to the graph matching or Procrustes processes were used as in the original paper. To optimize the final dictionary, we performed a parameter search over several parameters: (1) Merge operations in the training of

[6] https://github.com/bheinzerling/bpemb

Experiment	Num. Seeds
Swahili	866
Zulu	929
Zulu (t)	1030

Table 1: Number of available seeds per experiment. Zulu (t) indicates Zulu with translated corpus.

the Byte Pair Embeddings (10 000, 20 000, 50 000), (2) Computation of word embeddings using the mean or weighted average, (3) Number of seeds (50, 75, 100), and (4) Ending with Procrustes or graph matching.

The experiment was run 10 times for each combination of parameters on a randomized seed input. The pseudorandom number generator was seeded with the same value for each iteration across all experiments. The evaluation was performed using the unused seeds. An overview of the total available seeds for testing and supervision is available in Table 1. The implementations of Marchisio et al. (2022) served as a foundation for our experiments. The code can be found on GitHub[7].

4.1 BLI for Bantu Languages

Tables 2 and 3 contains the parameter combinations yielding the highest averaged score for each type of embedding and experiment with a small corpus. One observes that in our number of seeds regime, the best results are achieved with the highest number of seeds and when ending with Procrustes. A finding that coincides with the results of Marchisio et al. (2022).

The optimal vocabulary size appears to be dependent on the language, with 50 000 for Swahili and 20 000 for Zulu. Similarly, the optimal embedding types are language dependent. While the weighted outperform the mean Byte Pair embeddings, the basic word embeddings still provide better results for Swahili. Finally, the performance for Zulu is significantly improved by switching the crawled corpus with the translated corpus.

4.2 Isomorphism of Embedding Spaces

Another factor that helps understand the differences in performance is the degree of isomorphism of the two embedding spaces (Marchisio et al., 2022). Using the implementations by Vulić et al. (2020), we calculate the Gromov-Hausdorff (GH) Distance and Laplacian Eigenvector Similarities (EVS) for selected embedding spaces. Similar to the process of BLI, the similarity measures are computed using the embeddings of the 3000 most frequent words from each language, using the best performing parameter combination for each experiment (see Tables 2 and 3).

Experiment	GH Distance	Eigenvector Sim.
Swahili	0.06	12.94
Zulu	0.13	9.04
Zulu (t)	0.18	16.49

Table 4: Eigenvector similarity and Gromov-Hausdorff (GH) distances for selected language pairs

5 Discussion

Averaging 51.84% for the best parameter combination, this approach is highly successful for Swahili, yielding better results than some results for distant language pairs in the original paper by Marchisio et al. (2022). Yet, our scores are not directly comparable with theirs because of the larger training corpus and larger created dictionary size in the original experiments. Nevertheless, they do provide an adequate point of reference and demonstrate a successful application of this approach for an African language.

The performance is significantly worse for Zulu, averaging 2.37% for the best parameter combination with the crawled corpus. The precision can be improved by replacing the crawled English corpus with the translated English corpus, averaging 23.35%, showing that BLI can also be successfully applied to Zulu. Though it remains unclear, why the performances vary so strongly. An indicator is the low GH and EVS distances for Swahili, which correlate with a stronger performance in BLI (Marchisio et al., 2022). Consequently, one would expect the crawled corpus for Zulu to perform better than the translated corpus, which contradicts our observations. This discrepancy suggests that GH and EVS distances alone are insufficient to explain the observed performance differences, and that additional factors, such as corpus quality or domain mismatch, may also be relevant.

For the two Zulu experiments an additional factor may be a sensitivity to varying domains. The Swahili corpora were crawled from a bilingual publisher, mostly publishing similar articles in Swahili on Habari Leo and in English on Daily News. In

[7] https://github.com/addie9800/bantu-bli

Emb. Type	Vocab.	Score (Avg.)
Mean	50 000	44.03 ± 1.21
Weighted	50 000	49.46 ± 0.76
Word	-	**51.84** ± 1.01

Table 2: BLI for Swahili, showing the best averaged score per experiment and embedding type. All experiments achieved best results with 100 seeds and ending with Procrustes.

Corpus	Emb. Type	Vocab.	Score (Avg.)
Crawled	Mean	20 000	2.28 ± 0.46
Crawled	**Weighted**	20 000	**2.37** ± 0.46
Crawled	Word	-	1.75 ± 0.47
Translated	Mean	20 000	18.26 ± 1.40
Translated	**Weighted**	20 000	**23.35** ± 0.63
Translated	Word	-	13.89 ± 0.99

Table 3: BLI for Zulu, showing the best averaged score per experiment and embedding type. All experiments achieved best results with 100 seeds and ending with Procrustes.

absence of such a publisher for Zulu, we resorted to crawling from related publishers, which likely don't show such an overlap. The subsequent translation then effectively generated a parallel corpus, boosting the performance. It remains unclear, why also the translated corpus for Zulu performs significantly worse than Swahili.

It further appears to be beneficial to use subword embeddings for Bantu languages. In the case of Zulu, it outperforms regular word embeddings while achieving competitive results in Swahili. Additionally, the experiments appear to confirm the intuition that longer tokens should be given more weight, as the weighted embeddings consistently score higher than the mean embeddings.

Limitations

The experiments were only performed on two representatives of the Bantu language family, which are comparatively high-resourced, as they have online newspapers available and are additionally also supported by `Google Translate`. Repeating the experiments with more languages may provide insights into causes of the varying performance. Additionally, the experiments were performed using 3000 embeddings, whereas Marchisio et al. (2022) work with 200 000, and it remains open how well this approach scales for larger systems in the Bantu setting.

Previous work suggests (see section 2.4) that translation quality is sufficient, especially for individual words, yet no quantitative analysis of the quality of the generated seeds was performed, which would provide greater confidence in the results.

Finally, the strong results for Zulu were obtained using an English corpus generated by translating the Zulu corpus, effectively creating a parallel corpus, which is not realistic for an arbitrary low-resource language.

References

Kowshik Bhowmik and Anca Ralescu. 2023. Bridging the resource gap in cross-lingual embedding space. In *Modelling and Development of Intelligent Systems*, pages 122–135, Cham. Springer Nature Switzerland.

Sanghyuk Choi, Taeuk Kim, Jinseok Seol, and Sanggoo Lee. 2017. A syllable-based technique for word embeddings of Korean words. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 36–40, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. *Preprint*, arXiv:1710.04087.

Max Dallabetta, Conrad Dobberstein, Adrian Breiding, and Alan Akbik. 2024. Fundus: A simple-to-use news scraper optimized for high quality extractions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 305–314, Bangkok, Thailand. Association for Computational Linguistics.

Encyclopaedia Britannica Editors. 2025. Swahili language. Accessed 19 December 2025.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Raphael Iyamu. 2024. Machine translation and nlp tools: Developing and refining language technologies for african languages. *International Journal For Multidisciplinary Research*.

Pitso Walter Khoboko, Vukosi Marivate, and Joseph Sefara. 2025. Optimizing translation for low-resource languages: Efficient fine-tuning with custom prompt engineering in large language models. *Machine Learning with Applications*, 20:100649.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Kelly Marchisio, Youngser Park, Ali Saad-Eldin, Anton Alyakin, Kevin Duh, Carey Priebe, and Philipp Koehn. 2021. An analysis of Euclidean vs. graph-based framing for bilingual lexicon induction from word embedding spaces. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 738–749, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kelly Marchisio, Ali Saad-Eldin, Kevin Duh, Carey Priebe, and Philipp Koehn. 2022. Bilingual lexicon induction for low-resource languages using graph matching via optimal transport. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2545–2561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *Preprint*, arXiv:1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Preprint*, arXiv:1310.4546.

Ferdinand Mpiranya. 2023. *English-Swahili Swahili-English Immersive Dictionary*, 1 edition. Routledge.

Ndapa Nakashole. 2019. Bilingual dictionary induction for bantu languages. *Preprint*, arXiv:1811.07080.

Ugochi Okafor. 2025. Multilingual NLP for African healthcare: Bias, translation, and explainability challenges. In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 221–229, Vienna, Austria. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R Gormley, and Graham Neubig. 2019. BLISS in non-isometric embedding spaces.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović;. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(86):2487–2531.

Ali Saad-Eldin, Benjamin D. Pedigo, Carey E. Priebe, and Joshua T. Vogelstein. 2021. Graph matching via optimal transport. *Preprint*, arXiv:2111.05366.

Nabeta K. N. Sangili. 2024. Digitising kiswahili for translation economy. *Journal of Kiswahili and Other African Languages*, 2(2):44–51.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Casper S. Shikali, Zhou Sijie, Liu Qihe, and Refuoe Mokhosi. 2019. Better word representation vectors using syllabic alphabet: A case study of swahili. *Applied Sciences*, 9(18).

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are all good word vector spaces isomorphic? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1063–1077.

A Parameter Search Results

Tables 5, 6 and 7 show the results of the hyperparameter search across all parameters: embedding type, vocabulary size, number of seeds, whether or not to end with Procrustes (P-End) for Swahili, Zulu and Zulu with the translated corpus.

Emb. Type	#Seeds	Vocab.	P-End?	Score (Avg.)
Mean	50	10 000	True	41.04 ± 1.30
Mean	50	10 000	False	29.13 ± 2.54
Mean	75	10 000	True	41.73 ± 1.24
Mean	75	10 000	False	35.23 ± 1.84
Mean	100	10 000	True	43.07 ± 0.85
Mean	100	10 000	False	37.23 ± 0.92
Mean	50	20 000	True	42.33 ± 0.90
Mean	50	20 000	False	30.04 ± 4.14
Mean	75	20 000	True	43.12 ± 0.86
Mean	75	20 000	False	36.11 ± 1.61
Mean	100	20 000	True	43.02 ± 0.44
Mean	100	20 000	False	38.17 ± 1.45
Mean	50	50 000	True	43.53 ± 0.77
Mean	50	50 000	False	23.89 ± 4.36
Mean	75	50 000	True	43.35 ± 1.08
Mean	75	50 000	False	34.86 ± 1.92
Mean	*100*	*50 000*	*True*	44.03 ± 1.21
Mean	100	50 000	False	37.48 ± 1.18
Weighted	50	10 000	True	46.81 ± 1.36
Weighted	50	10 000	False	37.85 ± 2.35
Weighted	75	10 000	True	47.82 ± 0.79
Weighted	75	10 000	False	41.69 ± 0.73
Weighted	100	10 000	True	48.40 ± 0.81
Weighted	100	10 000	False	43.20 ± 1.20
Weighted	50	20 000	True	47.62 ± 0.95
Weighted	50	20 000	False	38.47 ± 1.13
Weighted	75	20 000	True	48.01 ± 0.41
Weighted	75	20 000	False	41.61 ± 1.29
Weighted	100	20 000	True	49.24 ± 1.15
Weighted	100	20 000	False	43.78 ± 0.78
Weighted	50	50 000	True	47.64 ± 0.72
Weighted	50	50 000	False	27.57 ± 8.96
Weighted	75	50 000	True	49.34 ± 0.93
Weighted	75	50 000	False	40.84 ± 1.01
Weighted	*100*	*50 000*	*True*	49.46 ± 0.76
Weighted	100	50 000	False	42.64 ± 1.22
Word	50	-	True	50.90 ± 0.98
Word	50	-	False	37.30 ± 2.36
Word	75	-	True	51.39 ± 1.05
Word	75	-	False	43.61 ± 1.20
Word	**100**	-	**True**	$\mathbf{51.84 \pm 1.01}$
Word	100	-	False	46.63 ± 1.46

Table 5: Parameter search results using the Swahili Corpus

Emb. Type	#Seeds	Vocab.	P-End?	Score (Avg.)
Mean	50	10 000	True	0.99 ± 0.49
Mean	50	10 000	False	0.54 ± 0.44
Mean	75	10 000	True	1.28 ± 0.45
Mean	75	10 000	False	0.80 ± 0.47
Mean	100	10 000	True	2.02 ± 0.31
Mean	100	10 000	False	0.94 ± 0.38
Mean	50	20 000	True	1.09 ± 0.72
Mean	50	20 000	False	0.48 ± 0.21
Mean	75	20 000	True	1.51 ± 0.48
Mean	75	20 000	False	0.73 ± 0.29
Mean	*100*	*20 000*	*True*	2.28 ± 0.46
Mean	100	20 000	False	0.69 ± 0.36
Mean	50	50 000	True	1.08 ± 0.33
Mean	50	50 000	False	0.31 ± 0.26
Mean	75	50 000	True	1.60 ± 0.58
Mean	75	50 000	False	0.44 ± 0.17
Mean	100	50 000	True	1.42 ± 0.33
Mean	100	50 000	False	0.88 ± 0.40
Weighted	50	10 000	True	1.40 ± 0.41
Weighted	50	10 000	False	0.45 ± 0.16
Weighted	75	10 000	True	1.98 ± 0.49
Weighted	75	10 000	False	0.79 ± 0.25
Weighted	100	10 000	True	2.31 ± 0.46
Weighted	100	10 000	False	1.05 ± 0.43
Weighted	50	20 000	True	1.16 ± 0.54
Weighted	50	20 000	False	0.39 ± 0.29
Weighted	75	20 000	True	1.60 ± 0.50
Weighted	75	20 000	False	0.65 ± 0.32
Weighted	**100**	20 000	**True**	$\mathbf{2.37 \pm 0.46}$
Weighted	100	20 000	False	0.84 ± 0.23
Weighted	50	50 000	True	1.11 ± 0.33
Weighted	50	50 000	False	0.27 ± 0.19
Weighted	75	50 000	True	1.47 ± 0.36
Weighted	75	50 000	False	0.55 ± 0.20
Weighted	100	50 000	True	1.82 ± 0.55
Weighted	100	50 000	False	0.68 ± 0.34
Word	50	-	True	0.89 ± 0.47
Word	50	-	False	0.30 ± 0.15
Word	75	-	True	1.36 ± 0.52
Word	75	-	False	0.41 ± 0.26
Word	*100*	-	*True*	1.75 ± 0.47
Word	100	-	False	0.74 ± 0.38

Table 6: Parameter search results using the crawled Zulu Corpus

Emb. Type	#Seeds	Vocab.	P-End?	Score (Avg.)
Mean	50	10 000	True	13.13 ± 1.49
Mean	50	10 000	False	8.76 ± 2.59
Mean	75	10 000	True	14.72 ± 0.72
Mean	75	10 000	False	11.49 ± 1.81
Mean	100	10 000	True	15.89 ± 1.12
Mean	100	10 000	False	11.61 ± 1.71
Mean	50	20 000	True	14.89 ± 0.87
Mean	50	20 000	False	10.30 ± 2.51
Mean	75	20 000	True	15.55 ± 1.45
Mean	75	20 000	False	12.39 ± 1.30
Mean	*100*	*20 000*	*True*	18.26 ± 1.40
Mean	100	20 000	False	14.54 ± 1.39
Mean	50	50 000	True	9.62 ± 0.91
Mean	50	50 000	False	5.82 ± 1.90
Mean	75	50 000	True	13.84 ± 0.75
Mean	75	50 000	False	7.99 ± 1.90
Mean	100	50 000	True	15.58 ± 1.10
Mean	100	50 000	False	9.24 ± 1.25
Weighted	50	10 000	True	21.08 ± 0.77
Weighted	50	10 000	False	19.94 ± 1.34
Weighted	75	10 000	True	22.77 ± 0.87
Weighted	75	10 000	False	20.75 ± 1.70
Weighted	100	10 000	True	23.01 ± 1.06
Weighted	100	10 000	False	20.29 ± 1.40
Weighted	50	20 000	True	21.64 ± 1.33
Weighted	50	20 000	False	19.13 ± 2.89
Weighted	75	20 000	True	21.27 ± 2.29
Weighted	75	20 000	False	20.44 ± 1.35
Weighted	**100**	20 000	**True**	$\mathbf{23.35 \pm 0.63}$
Weighted	100	20 000	False	20.52 ± 1.88
Weighted	50	50 000	True	16.70 ± 2.42
Weighted	50	50 000	False	15.92 ± 2.35
Weighted	75	50 000	True	19.67 ± 1.92
Weighted	75	50 000	False	17.26 ± 3.66
Weighted	100	50 000	True	20.41 ± 2.21
Weighted	100	50 000	False	16.34 ± 3.22
Word	50	-	True	7.55 ± 1.70
Word	50	-	False	1.35 ± 0.48
Word	75	-	True	11.52 ± 1.69
Word	75	-	False	2.86 ± 0.88
Word	*100*	-	*True*	13.89 ± 0.99
Word	100	-	False	3.98 ± 0.77

Table 7: Parameter search results using the translated Zulu Corpus