Enhancing Bilingual Lexicon Induction with Dynamic Translation

Michaela Denisová[1][a] and Pavel Rychlý[1,2][b]
[1]*Natural Language Processing Centre, Masaryk University, Brno, Czech Republic*
[2]*Lexical Computing, Brno, Czech Republic*
fi

Keywords: Cross-Lingual Embedding Models, Parameter *K*, Classification Neural Network.

Abstract: Bilingual lexicon induction (BLI) has been a popular task for evaluating cross-lingual word embeddings (CWEs). The prevalent metric employed in the evaluation is precision at k, where k represents the number of target words retrieved for each source word. However, establishing a fixed k for the entire evaluation dataset proves challenging due to varying target word counts for each source word. This leads to limited results, compromising either precision or recall. In this paper, we present a novel classification-based approach with dynamic k for bilingual lexicon induction that aims to identify all relevant target words for each source word by exploiting the information derived from the aligned embeddings while offering a balanced trade-off between precision and recall. On top of that, it enables the evaluation of the existing CWEs using dynamic k. Compared to the standard baseline systems and evaluation procedures, it provides competitive results.

1 INTRODUCTION

Retrieving translations of individual words is an intrinsic evaluation task referred to as bilingual lexicon induction (BLI). This task has been a commonly used method for evaluating cross-lingual word embeddings (CWEs), which aim to align two (or more) sets of individually trained monolingual word embeddings (MWEs) into a shared cross-lingual space where similar words are represented by similar vectors (Ruder et al., 2019).

Owing to this property, they have shown to be beneficial in many NLP applications, e.g., machine translation (Artetxe et al., 2018c; Duan et al., 2020; Zhou et al., 2021; Wang et al., 2022), cross-lingual information retrieval (Vulić and Moens, 2015), language acquisition and learning (Yuan et al., 2020).

In the BLI task, the method aims to generate a list of target words for each source word, ranking them based on the cosine similarities between their respective embeddings. Afterwards, top-k target words for each source word are selected, and the word pairs are compared to the evaluation dataset (Ruder et al., 2019).

However, the k is not determined by the method and often is set by the evaluation metrics or derived from the evaluation data. This limitation makes the approach less reflective of real-world translation scenarios, where the number of target words corresponding to a source word cannot be estimated and is vital to be determined by the model to successfully fulfil the BLI task's objective.

In many papers, the preferred metric is precision at k (P@k), where k is fixed, usually $k = \{1, 5, 10\}$ (e.g., (Mikolov et al., 2013; Conneau et al., 2017; Li et al., 2022; Tian et al., 2022)). What the papers actually report is HitRatio@k, where HitRatio@1 = P@1 and P@k_1 > P@k_2 as long as $k_1 > k_2$ (Conneau et al., 2017). This is problematic for two reasons.

Firstly, the majority of the source words are likely to have more than one target word, and the number of target words differs for each source word. For example, the most popular evaluation datasets MUSE (Conneau et al., 2017) consist of word-to-many lists: the English-French evaluation dataset contains 2943 word pairs from which are only 1.5K unique English words. As Table 1 shows, the English source words exhibit various numbers of target words (e.g., *compact* - *compact*, *compacte*, *compactes*, *compacts*, *compresser*, *pacte*; *admit* - *admet*, *admets*, *admettre*; *subway* - *métro*).

Secondly, since HitRatio@k assumes that every source word has only one target word, in cases where $k > 1$, the metric may yield results exceeding 100%, which leads to distortions in the results.

While prior work suggested replacing P@k with

[a] https://orcid.org/0009-0001-8402-504X
[b] https://orcid.org/0000-0001-5097-4610

Denisová, M. and Rychlý, P.
Enhancing Bilingual Lexicon Induction with Dynamic Translation.
DOI: 10.5220/0013346000003890
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 735-744
ISBN: 978-989-758-737-5; ISSN: 2184-433X
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

Table 1: The number of target words (TGW) in four MUSE evaluation datasets and the dynamic *k* values (NN *k*) that were predicted by VM-S+NN model trained on English-Spanish.

TGW	en-fr	en-cs	en-ko	en-es	NN *k*
1	698	820	1085	663	801
2	409	398	310	435	465
3	210	207	63	211	175
4	123	61	7	146	52
5	55	13	0	45	6
6+	5	1	0	0	1

Mean Average Precision (MAP) to address this issue, they evaluated their models with one-to-one datasets (Glavaš et al., 2019). We argue that in a real-language scenario, the source word is improbable to have only one target word, and even the less frequent words bear multiple meanings. For example, specific-domain-related words also occur in regular texts (*string* - sequence of characters vs a piece of rope, series of events, etc.).

Another attempt to advocate the MAP metric appeared at the BUCC 2022 conference (Adjali et al., 2022; Laville et al., 2022). While MAP is a valuable metric for assessing a model based on the ranking of target words and the quality of the embeddings' alignment, it fails to consider the parameter *k*, which is set in advance according to the evaluation dataset.

To relax from the constraint of having a fixed number of target words, dynamic translation was introduced in the shared task of the BUCC 2020 conference together with alternative evaluation metrics, such as recall and F1 scores (Rapp et al., 2020). While the participants advocated computing a threshold for similarity scores between the source and target word embeddings instead, they had to tailor it for each language pair individually.

Another existing work proposes classification-based approaches (Heyman et al., 2017; Severini et al., 2020a). In line with the previous research, framing the BLI task as a classification problem not only allows for dynamic *k* but also leads to additional improvements in the models' performance (Irvine and Callison-Burch, 2017; Karan et al., 2020). However, these methods suffer from computational inefficiencies, applying deep neural network to each word pair that is being classified.

Motivated by these insights, we implement a novel, simple classification-based approach, allowing for a dynamic *k* while exploiting various features derived from the aligned embeddings. The aim is to identify as many relevant target words as possible for each source word and report P, recall, and F1 scores not constrained by a predefined set of *k* nearest neighbours while balancing P and recall and maintaining high performance.

Different to previous endeavours introducing classification-based approaches (Heyman et al., 2017; Severini et al., 2020a) and similar approaches establishing dynamic *k* and alternative evaluation metrics (Rapp et al., 2020), our method is more straightforward to implement and more computationally efficient, as we demonstrate in this paper. It builds up a solution for existing CWEs to relax the constraint of having a fixed *k*, making them comparable with methods using dynamic *k* and improving their performance.

We evaluate our approach on the widely used evaluation datasets MUSE for various languages: English (en) to German (de), French (fr), Spanish (es), Russian (ru), Czech (cs), Dutch (nl), Finnish (fi), and Korean (ko), and on manually annotated data for Estonian (et) to Slovak (sk).

Our contribution is manifold.

1. We present a classification-based framework for bilingual lexicon induction that dynamically determines *k*, addressing the limitations of fixed *k* in traditional methods.
2. We propose a new solution that enables a more accurate evaluation of existing CWEs by balancing P and recall without predefining the number of nearest neighbours.
3. We provide a rigorous evaluation across diverse language pairs, demonstrating consistent improvements over state-of-the-art baselines.
4. To encourage reproducibility and further research, we make our datasets, code, and models publicly available. [1]

2 RELATED WORK

The pioneering work introducing the embedding-based method evaluated on the BLI task was proposed by (Mikolov et al., 2013). In their work, the authors reported results using metrics P@1 and P@5. Since then, the BLI task has enjoyed popularity among researchers as the mainstream task for the CWE evaluation and precision as the main reported metric, including the most cited baseline methods such as MUSE (Conneau et al., 2017) and VECMAP (Artetxe et al., 2018a; Artetxe et al., 2018b).

The more comprehensive evaluation study suggesting an alternative evaluation metric was proposed in (Glavaš et al., 2019). They criticised the lack of consistency and statistical significance testing in

[1] https://github.com/x-mia/Word_pair_classifier

existing evaluations, hampering thorough comparisons. (Glavaš et al., 2019) recommended using MAP, claiming this metric to be more informative since it does not treat all models that rank the correct translation below k equally. The same argument was brought up later at the BUCC 2022 conference (Adjali et al., 2022; Laville et al., 2022), which employed MAP metric in the shared task. In this paper, we concentrate on two submissions: CUNI (Požár et al., 2022) and IJS (Repar et al., 2022).

In these papers, the CUNI team implemented three approaches: static embeddings with posthoc alignment (CUNI$_{muse}$), unsupervised phrase-based machine translation using the Monoses pipeline (CUNI$_{mono}$), and contextualized multilingual embeddings from pre-trained models like BERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019) (CUNI$_{comb}$). In contrast, the IJS approach integrated linguistic, neural, and sentence-transformer features into an SVM binary classifier consisting of three settings (IJS$_1$, IJS$_2$, IJS$_3$). While CUNI focuses on multiple alignment techniques, IJS prioritizes comprehensive feature integration for precise term alignment.

Another attempt to bring different metrics into BLI evaluation appeared in the shared task of the BUCC 2020 conference (Rapp et al., 2020). The authors were instructed to report on recall and F1 scores, in addition to traditional precision, without setting a fixed k. In this paper, we focus on two models: LMU (Severini et al., 2020b) and LS2N (Laville et al., 2020).

In both papers, distinct methodologies were employed to ascertain a dynamic k. (Severini et al., 2020b) calculated a local threshold value for each source word instead of using a global threshold. The score of each candidate word T for a given source word S is determined by a function that considers the margin between the similarity of S and T and the average similarity of S with its most similar candidates. Each target candidate is considered a translation if its score exceeds the threshold value. The threshold value is tuned individually for each language pair.

(Laville et al., 2020) exploited scores from cosine similarity-based measure CSLS (Conneau et al., 2017). Then, they employed two criteria to limit the scores: i) setting a maximum number of candidates to retain for each source word and ii) establishing a minimum CSLS value to validate candidates. Each language pair had its specific threshold value.

A new line of the BLI research introduced classification-based approaches (e.g., (Irvine and Callison-Burch, 2017; Heyman et al., 2017; Severini et al., 2020a)), which relax the constraint of having a fixed k and offered an alternative evaluation metrics, such as recall and F1 score demonstrating the balance in the performance.

(Irvine and Callison-Burch, 2017) leveraged temporal word variation, normalised edit distance, and word burstiness, among other inputs, to train a linear classifier using a set of training translation pairs. Contrarily, (Heyman et al., 2017) suggested incorporating word-level and character-level representations within a deep neural network architecture instead. They provided experiments with various models. In this paper, we mention the models exploiting word-level representations (CLASS_SGNS) utilising word-level representations from the SGNS model (Mikolov et al., 2013), character-level representations (CHAR-LSTM$_{joint}$), and both in a combined model.

Additionally, they set a threshold t for the classification scores instead of fixed k, which they further fine-tuned on a validation set. This enabled them to enhance the model's performance evaluated with F1 scores.

Finally, (Severini et al., 2020a) proposed a novel approach, enabling the languages with different scripts to exploit orthographic features via transliteration. They integrated semantic and orthographic information using a transliteration system, seq2seqTr (m+BOEs). In contrast to (Heyman et al., 2017), the reported metric was HitRatio@ with a fixed set of retrieved target words for each source word.

3 METHODOLOGY

We introduce a classification neural network, leveraging its ability to enable dynamic k. Each source word w^s is processed by the network separately. Let V_s and V_t be the sets of all source and target words, respectively. Given a list of target candidates C, which is a list of the top 10 most similar target words, where $C \subset V_t$, it could also be denoted as $C = \{w^t_i | i = 1...10\}$ and let w^t_1 be the most similar top target candidate, the aim is to learn a function:

$$0,1 \leftarrow f(C), \tag{1}$$

where the input C is not a target candidate (or its embedding) directly but a vector of features derived from the similarity of the candidate vector and similarities of other candidates, and it can be formulated as follows:

$$0,1 \leftarrow f(sim, S^d, S^r, R^t, R^s, F^s) \tag{2}$$

The classification neural network produces the output of either 0 or 1. When the neural network identifies a target candidate w^t_i as the corresponding translation, it assigns a value of 1. The count of 1 associated with a source word w_n^s is equal to the value of k.

The training of the classification neural network requires sets of positive and negative examples to make a correct prediction. For that purpose, we exploit the evaluation datasets MUSE and the baseline CWEs. The data is described in Section 4.1 in further detail. Then, the neural network is trained by minimising the binary cross-entropy loss, defined as follows:

$$-\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log(\hat{y}_i) + (1-y_i)\cdot\log(1-\hat{y}_i)\right], \tag{3}$$

where N is the length of the training data, y_i is the true label for the i-th instance (either 0 or 1), and $\hat{y}_i$ is the predicted probability that the i-th instance belongs to class 1.

The key component of the classification neural network is the input layer, consisting of a vector of features representing a target candidate. This vector contains six features, i.e., cosine similarity score, absolute difference score, ratio score, and normalised target's word position, source word's rank and corpus frequency log values. The first four are computed from the CWEs using cosine similarity.

Let $X^s \times X^t$ be the aligned word embedding matrices of the $V_s \times V_t$. The cosine similarity score between the word embeddings (x_s, x_i) corresponding to the word pair (w_s, w_i) is defined as:

$$sim_i = sim(x_s, x_i), \tag{4}$$

where the $sim()$ function represents the dot product between the source and target word embeddings.

The absolute difference score S^d is then computed as:

$$S_i^d = sim_1 - sim_i, \tag{5}$$

where sim_1 denotes the similarity between the closest target word embedding and the source word embedding.

Similarly, the ratio score S^r is calculated by the following formula:

$$S_i^r = sim_i / sim_1 \tag{6}$$

Furthermore, we can derive the normalised target's word position log value as $R_i^t = Norm(|\{i|sim_j < sim_i\}|)$. F_i^s normalised frequency of source word w_s in a corpus and R_i^s normalised rank of the source word w_s in MWE, while the $Norm()$ function is defined as:

$$Norm(x,C) = (log(x) - min_C)/(max_C - min_C) \tag{7}$$

These features are then combined as an input vector $Z \in \mathbb{R}^6$, which is fed into the neural network c_h:

$$c_{h_0} = \tanh(W_{h_0} \cdot Z + b_{h_0}) \tag{8}$$

$$c_{h_i} = \tanh(W_{h_i} \cdot c_{h_{i-1}} + b_{h_i}) \tag{9}$$

$$prediction = \sigma(W_p \cdot c_{h_T} + b_p), \tag{10}$$

where tanh and σ represent tanh and sigmoid activation functions, respectively, and T expresses the number of hidden layers implemented in the neural network.

The weight matrices W_{h_i}, W_p and bias terms b_{h_i}, b_p are learned during the training process through backpropagation. The activation functions tanh and σ are applied after each transformation to introduce nonlinearity, which is essential for capturing complex relationships in the data.

4 EXPERIMENTAL SETUP

In this section, we outline the key components of the experiments that were conducted.

4.1 Data

To train the classification neural network for all languages in combination with English, we utilised the widely used evaluation datasets MUSE (Conneau et al., 2017) and treated them as positive examples, all labelled as 1.

Afterwards, we generated negative examples by retrieving the ten most similar target candidates C and their vectors of features for each source word from the evaluation dataset using different CWE models described in Section 4.2. All retrieved word pairs that did not occur in the evaluation dataset received label 0.

Each MUSE dataset consists of 1.5K source words, meaning we obtained 15K word pairs for each CWE model. We randomly sampled 8K word pairs from each dataset and split them into 5K training data, 1.5K testing data, and 1.5K validation data. We train our model using train and test data and report our results on validation data.

For the Estonian-Slovak language pair, we exploited the manually compiled and annotated data from (Denisová, 2022). These datasets were much smaller than the ones obtained from MUSE, resulting in 600 training and 100 testing word pairs for each model.

4.2 Implementation Details

4.2.1 CWE

To retrieve aligned monolingual word embeddings we utilised two state-of-the-art CWE frameworks, MUSE and VECMAP (VM) in a supervised (MUSE-S, VM-S), unsupervised (MUSE-U, VM-U) mode and mode that relies on identical strings (MUSE-I, VM-I).

The default settings closely followed the MUSE training described in (Conneau et al., 2017), and VM-S and VM-I in (Artetxe et al., 2018a), and VM-U settings in (Artetxe et al., 2018b). We used pre-trained fastText (Grave et al., 2018) on Wikipedia with dimension 300. We induced the first 200K aligned embeddings. To train supervised systems (MUSE-S; VM-S), we utilise MUSE training datasets.

4.2.2 Classification Neural Network

The classification neural network was implemented in Python using TensorFlow (Abadi et al., 2016). We utilised three hidden layers with 24-12-8 nodes. We used Adam optimizer for training with a 0.001 learning rate. The training for each language pair ran for 500 epochs.

4.3 Baselines

We compare our model with the results from classification-based approaches presented in (Heyman et al., 2017) (CLASS_SGNS, CHAR-LSTM$_{joint}$, combined) and (Severini et al., 2020a) (m+BOEs), the best outcomes submitted by CUNI (CUNI$_{muse}$) (Požár et al., 2022) and IJS (IJS$_2$) (Repar et al., 2022) at the BUCC 2022 conference in a shared task (Adjali et al., 2022), and the results obtained by models LMU (Severini et al., 2020b) and LS2N (Laville et al., 2020) at the BUCC 2020 conference in a shared task using dynamic k (Rapp et al., 2020).

Since the codes of these models are not publicly available, we directly juxtapose our system's performance against the outcomes reported in the papers.

4.4 Metrics

Precision: at k (P@k) computes the ratio of true positives (TP) to the sum of true positives and false positives (FP). In other words, it is the ratio of the positive target words to the number of all target words that the model found (positive and negative). In this case, k represents the number of the source word's nearest neighbours that were extracted.

Recall: (R) is calculated using the standard formula.

F1: score summarises the model's performance by capturing both metrics: P and R, showing the balance between them, and it is computed in a standard way as well.

5 EVALUATION

This section reports the main results obtained with our classification neural network. It is split into two parts to distinguish when our model is being used as a novel approach for BLI and as an extension for existing baseline CWE models, enabling the dynamic k. In the first one, we assess the outcomes concerning our model and discuss them against baseline models stated in Section 4.3.

In the second one, we compare the results of the state-of-the-art CWE models described in Section 4.2, evaluated using fixed k and classification neural network with dynamic k.

5.1 Classification Neural Network

5.1.1 Efficiency

We implemented a minimalist 3-hidden-layer classification neural network. This network efficiently predicts 1.5K source words within a short span of approximately 1.01 seconds. According to the (Heyman et al., 2017), their method has a time complexity of $O(|V_S| \times |V_T|)$ multiplied by the complexity of g [2], making it a computationally intensive process, especially for extensive vocabularies, where it becomes impractically costly. Our approach offers greater efficiency when compared to the RNN neural network presented in (Heyman et al., 2017).

5.1.2 Overall Results

The results across four language pairs (English to French, Spanish, Russian, and German) compared to the baselines CUNI$_{muse}$, IJS$_2$, LMU, LS2N, and m+BOEs are provided in Table 2. The comparison of the F1 scores with the models presented in (Heyman et al., 2017) trained on the English-Dutch language pair is displayed in Table 3.

Tables 2 and 3 indicate that our classification approach outperforms almost all baselines within a margin of approximately 1% to 20%. In particular, the model MUSE-S+NN stands out, achieving the highest results for English-Russian and English-Dutch and

[2] Where V_S and V_T denote source and target words and g denotes classifier.

Table 2: P, R, and F1 score using dynamic *k* (*CWE*+NN) compared to the baselines CUNI$_{muse}$ (Požár et al., 2022), IJS$_2$ (Repar et al., 2022), LMU (Severini et al., 2020b), LS2N (Laville et al., 2020), and m+BOEs (Severini et al., 2020a).
[*]In the article, the authors present their findings on HitRatio@*k*, which cannot be directly compared to our results. Consequently, we only make comparisons using the reported HitRatio@1, equivalent to P@1.

	en-de			en-fr			en-es			en-ru		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CUNI$_{muse}$	-	-	-	39.8	31.7	35.3	-	-	-	-	-	-
IJS$_2$	-	-	-	2.87	**80.0**	5.55	-	-	-	-	-	-
LMU	40.2	59.8	48.1	-	-	-	-	-	-	33.9	37.8	35.8
LS2N	54.3	54.8	54.5	61.2	69.7	65.1	63.8	61.4	62.6	32.6	38.7	35.4
m+BOEs	-	-	-	-	-	-	-	-	-	36.0*	-	-
MUSE-S+NN	56.8	**73.8**	**64.2**	63.4	61.1	62.2	**77.6**	59.0	67.0	**66.4**	**75.8**	**70.8**
MUSE-I+NN	49.3	58.1	53.3	62.7	61.3	62.0	67.6	63.6	65.5	43.2	56.4	48.9
MUSE-U+NN	57.8	56.8	57.3	52.7	63.1	57.4	73.9	63.5	**68.3**	37.3	40.7	39.0
VM-S+NN	**60.7**	67.1	63.7	64.7	64.7	64.7	72.5	62.2	67.0	42.3	63.9	50.9
VM-I+NN	54.5	63.7	58.8	66.4	70.3	68.3	57.3	**71.1**	63.5	39.3	59.7	47.4
VM-U+NN	55.3	62.1	58.5	**68.8**	68.5	**68.6**	71.8	61.3	66.1	41.6	50.0	45.4

performing well across English-Spanish and English-German. The only exception is the English-French language pair, where the baseline IJS$_2$ surpassed our best model VM-I+NN by almost 10 %.

Table 3: Comparison of F1 using dynamic *k* (*CWE*+NN) to the three models presented in (Heyman et al., 2017) evaluated on English-Dutch.

	F1
CLASS_SGNS	19.8
CHAR-LSTM$_{joint}$	34.9
combined	36.6
MUSE-S+NN	**71.5**
MUSE-I+NN	58.4
MUSE-U+NN	62.4
VM-S+NN	61.6
VM-I+NN	67.3
VM-U+NN	63.1

5.1.3 Setting the Dynamic *K*

Table 4 provides a sample of analysed source word *admit* along with target candidates and their vectors of features derived from the VM-S model trained using English-Spanish. Fig. 1 visualises the correlation between the features *sim* and R^t derived from the same model across the entire English-Spanish validation data and assigned values of 1 or 0.

The English word *admit* has in the English-Spanish MUSE evaluation dataset four target words, i.e., *admita*, *admite*, *admiten*, *admitir*. The classification neural network assigned to three of them the value of 1 but found an additional target word *admitirlo* and did not include *admiten* that was found at rank 1326. Thus, the *k* was set to 4 for the source word *admit*.

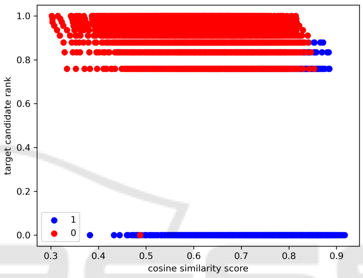

Figure 1: Correlation between *sim* (cosine similarity score) and R^t (target candidate rank) features across the VM-S+NN model trained using English-Spanish labelled as 0 or 1.

The analysis of the feature vectors of target candidates displayed in Table 4 suggests a strong correlation between the scores' magnitudes, various ranks, and assigned labels, i.e., the higher *sim* value and the lower R^t value increase the probability of a target candidate being labelled as 1. Since the classification neural network learns patterns using information derived from CWEs and ranks from MWEs and corpus data, it plays a more crucial role than the linguistic aspects of the language, highlighting the significance of the quality of the MWEs and CWEs' alignment method.

Additionally, Table 1 compares the number of target words in the evaluation dataset and the values of *k* set by the classification neural network. For example, in the evaluation dataset, 435 source words have 2 target words and the classification neural network set *k* = 2 for 465 source words.

5.2 Extension

In the second part of the evaluation, we evaluated the performance of the CWE models MUSE-S, MUSE-I,

Table 4: Example from the VM-S+NN model trained on the English-Spanish language pair. SRC = source word, TGT = target word, ED = evaluation dataset, C = correct.

SRC	**rank**	**TGT**	**NN**	**ED**	**C**	*sim*	S^d	S^r	R^t	R^s	F^s
admit	0	admitir	1	✓	✓	0.799	0.0	1.0	0.0	0.922	0.692
k = 4	1	admitirlo	1	×	✓	0.724	0.074	0.906	0.758	0.922	0.692
	2	admita	1	✓	✓	0.720	0.078	0.901	0.834	0.922	0.692
	3	admite	1	✓	✓	0.710	0.089	0.888	0.879	0.922	0.692
	4	admito	0	×	✓	0.709	0.090	0.886	0.910	0.922	0.692
	5	admitiendo	0	×	✓	0.703	0.096	0.879	0.935	0.922	0.692
	6	entenderla	0	×	×	0.688	0.110	0.861	0.955	0.922	0.692
	7	creer	0	×	×	0.683	0.115	0.855	0.972	0.922	0.692
	8	admitan	0	×	✓	0.668	0.131	0.835	0.987	0.922	0.692
	9	ignorarla	0	×	×	0.666	0.133	0.833	1.0	0.922	0.692
	1326	admiten	-	✓	✓	-	-	-	-	-	-

MUSE-U, VM-S, VM-I, and VM-U by employing the P and F1 scores with fixed and dynamic *k*. [3]

For the fixed *k*, we chose values in {1, 3, 5}. The dynamic *k* is set by the classification neural network acting as an extension for the CWEs [4], and we denote P and F1 scores using dynamic *k* as P@NN and F1@NN, respectively.

The overall results across all language pairs displaying F1 scores are provided in Table 7, and P metrics in Table 8, both placed in Appendix. We can observe that although in nearly all cases, the P@1 evaluation yields better results, almost all models offer a significantly better balance between P and R when dynamic *k* is employed, improving F1 scores by a margin rising up to almost 58%.

Table 5: F1 score of the best model when evaluated with fixed *k* (F1@1) vs. the best model when evaluated with dynamic *k* (F1@NN).

	Best F1@1		Best F1@NN	
en-cs	VM-S	39.0	VM-I	51.4
en-fi	VM-I	32.8	VM-U	48.3
et-sk	VM-S	26.7	VM-U	68.0

On top of that, Table 5 shows how the models' ranking changes when evaluated using fixed and dynamic *k* across three language pairs. For example, when evaluating models on the Estonian-Slovak language pair with a fixed *k*, the VM-S model achieves the highest performance. However, when using a dynamic *k* metric, the VM-U model outperforms all others by more than 41%. This can be illustrated using the example of the Estonian word *sõdur* (*soldier*) in Table 6. Using *k* = 1 for the evaluation would yield poorer performance, as the top-1 induced target word is absent from the evaluation dataset despite its correctness. In contrast, the classification neural network identified the correct target word from the evaluation dataset, as well as an additional correct word not present in the dataset. This approach not only enables efficient selection of the optimal model but also accurately identifies target words despite biases in the evaluation datasets (see Section 6).

Table 6: Example *sõdur* - *soldier* from the VM-U+NN model trained on the Estonian-Slovak language pair. SRC = source word, TGT = target word, ED = evaluation dataset, C = correct.

SRC	**rank**	**TGT**	**NN**	**ED**	**C**
sõdur	0	vojaka	1	×	✓
k = 2	1	vojak	1	✓	✓
	2	bojovník	0	×	×
	3	voj	0	×	×
	4	bojovníka	0	×	×
	5	lukostrelec	0	×	×
	6	civilista	0	×	×
	7	voják	0	×	×
	8	delostrelec	0	×	×
	9	pechota	0	×	×

6 LIMITATIONS

Over the years, the MUSE datasets have been frequently used for the BLI evaluation. Despite their popularity, several concerns have emerged. (Kementchedjhieva et al., 2019) revealed that a significant portion of the word pairs are comprised of proper nouns, which do not reflect the performance reliably. Later, (Laville et al., 2022) pointed out more serious issues, such as the fact that the datasets contain over 30% identical word pairs and around 40% graphically close word pairs.

Another problem is the bias that occurs when creating positive and negative examples for the evalua-

[3] While HitRatio@*k* is often favoured, in this paper, we opt to utilise P@*k* for the reasons outlined in the Introduction.

[4] *Model-X*+NN means that the classification neural network was trained and evaluated on the output from *model-X*.

tion. For example, the Spanish word *admitir* can have over 40 valid translations for the English word *admit*, depending on the context. Moreover, the verb *admit* could also be translated by *reconocer* or *confesar*, which convey similar meanings. As a result, the top 10 words identified by the proposed method might include some of these over 100 suitable translations, which are classified as negative pairs. This has a negative impact on the accuracy of the labels.

When demonstrated using our model, Table 4 shows that there were only four forms of the verb *admitir* in the evaluation dataset, whereas the model generated 7 viable word forms, four of them missing in the evaluation dataset (*admitirlo*, *admito*, *admitiendo*, *admitan*). Moreover, alternative translations like *reconocer* or *confesar* were not captured, indicating areas for improvement in contextual understanding.

7 CONCLUSION

In this paper, we have presented a novel classification-based approach to BLI, addressing the limitations of traditional evaluation metrics by introducing dynamic *k* for enhanced P, R, and F1 scores. We evaluated our approach across diverse language pairs, showing its benefits as a new approach for the BLI task and as an extension for existing CWE approaches, enabling dynamic *k*.

To summarise, the evaluation of CWE models using P@1 yields seemingly impressive results. However, it only assesses a small part of the evaluation dataset. Therefore, employing dynamic *k* provides a more accurate picture of the model's performance while balancing P and R. Additionally, the results suggest that for determining the correct target candidates, not only the absolute numbers are important, but also the ranks and the scores relative to the highest score achieved.

Moreover, we have demonstrated that our approach is computationally efficient and produces competitive results when compared to the current baseline systems.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv*, abs/1603.04467.

Adjali, O., Morin, E., Sharoff, S., Rapp, R., and Zweigenbaum, P. (2022). Overview of the 2022 BUCC Shared Task: Bilingual Term Alignment in Comparable Specialized Corpora. In *BUCC, 15th Workshop on Building and Using Comparable Corpora*, pages 67–76.

Artetxe, M., Labaka, G., and Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019.

Artetxe, M., Labaka, G., and Agirre, E. (2018b). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2018c). Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642. Association for Computational Linguistics.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and J'egou, H. (2017). Word translation without parallel data. *ArXiv*, abs/1710.04087.

Denisová, M. (2022). Parallel, or comparable? that is the question: The comparison of parallel and comparable data-based methods for bilingual lexicon induction. In *Proceedings of the Sixteenth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2022*, pages 4–13. Tribun EU.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Duan, X., Ji, B., Jia, H., Tan, M., Zhang, M., Chen, B., Luo, W., and Zhang, Y. (2020). Bilingual dictionary based neural machine translation without using parallel sentences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1570–1579. Association for Computational Linguistics.

Glavaš, G., Litschko, R., Ruder, S., and Vulić, I. (2019). How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721. Association for Computational Linguistics.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).*

Heyman, G., Vulić, I., and Moens, M.-F. (2017). Bilingual lexicon induction by learning to combine word-level and character-level representations. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1085–1095. Association for Computational Linguistics.

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

Karan, M., Vulić, I., Korhonen, A., and Glavaš, G. (2020). Classification-based self-learning for weakly supervised bilingual lexicon induction. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6915–6922. Association for Computational Linguistics.

Kementchedjhieva, Y., Hartmann, M., and Søgaard, A. (2019). Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3336–3341. Association for Computational Linguistics.

Laville, M., Hazem, A., and Morin, E. (2020). TALN/LS2N participation at the BUCC shared task: Bilingual dictionary induction from comparable corpora. In Rapp, R., Zweigenbaum, P., and Sharoff, S., editors, *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 56–60. European Language Resources Association.

Laville, M., Morin, E., and Langlais, P. (2022). About evaluating bilingual lexicon induction. In Rapp, R., Zweigenbaum, P., and Sharoff, S., editors, *Proceedings of the BUCC Workshop within LREC 2022*, pages 8–14. European Language Resources Association.

Li, Y., Liu, F., Vulić, I., and Korhonen, A. (2022). Improving bilingual lexicon induction with cross-encoder reranking. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4100–4116. Association for Computational Linguistics.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168.*

Požár, B., Tauchmanová, K., Neumannová, K., Kvapilíková, I., and Bojar, O. (2022). CUNI submission to the BUCC 2022 shared task on bilingual term alignment. In Rapp, R., Zweigenbaum, P., and Sharoff, S., editors, *Proceedings of the BUCC Workshop within LREC 2022*, pages 43–49. European Language Resources Association.

Rapp, R., Zweigenbaum, P., and Sharoff, S. (2020). Overview of the fourth BUCC shared task: Bilingual dictionary induction from comparable corpora. In Rapp, R., Zweigenbaum, P., and Sharoff, S., editors, *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 6–13. European Language Resources Association.

Repar, A., Pollak, S., Ulčar, M., and Koloski, B. (2022). Fusion of linguistic, neural and sentence-transformer features for improved term alignment. In Rapp, R., Zweigenbaum, P., and Sharoff, S., editors, *Proceedings of the BUCC Workshop within LREC 2022*, pages 61–66. European Language Resources Association.

Ruder, S., Vulić, I., and Søgaard, A. (2019). A survey of cross-lingual word embedding models. *The Journal of Artificial Intelligence Research*, 65:569–631.

Severini, S., Hangya, V., Fraser, A., and Schütze, H. (2020a). Combining word embeddings with bilingual orthography embeddings for bilingual dictionary induction. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6044–6055. International Committee on Computational Linguistics.

Severini, S., Hangya, V., Fraser, A., and Schütze, H. (2020b). LMU bilingual dictionary induction system with word surface similarity scores for BUCC 2020. In Rapp, R., Zweigenbaum, P., and Sharoff, S., editors, *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 49–55. European Language Resources Association.

Tian, Z., Li, C., Ren, S., Zuo, Z., Wen, Z., Hu, X., Han, X., Huang, H., Deng, D., Zhang, Q., and Xie, X. (2022). RAPO: An adaptive ranking paradigm for bilingual lexicon induction. *ArXiv*, abs/2210.09926.

Vulić, I. and Moens, M.-F. (2015). Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 363–372.

Wang, X., Ruder, S., and Neubig, G. (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. Association for Computational Linguistics.

Yuan, M., Zhang, M., Van Durme, B., Findlater, L., and Boyd-Graber, J. (2020). Interactive refinement of cross-lingual word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5984–5996. Association for Computational Linguistics.

Zhou, Y., Geng, X., Shen, T., Zhang, W., and Jiang, D. (2021). Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5822–5834. Association for Computational Linguistics.

APPENDIX

Tables 7 and 8 present the outcomes of the models MUSE-S+NN, MUSE-I+NN, MUSE-U+NN, VM-S+NN, VM-I+NN, and VM-U+NN evaluated using F1 scores (F@5, 3, 1, NN), and P metrics (P@5, 3, 1, NN) across all language pairs, respectively.

Table 7: Reported F1 score (F1@5, F1@3, F1@1, and F1@NN (*k* selected by the classification neural network)) for the MUSE-S, MUSE-I, MUSE-U, VM-S, VM-I, and VM-U models evaluated with the MUSE evaluation datasets.

	MUSE-S				MUSE-I				MUSE-U			
F1@	5	3	1	NN	5	3	1	NN	5	3	1	NN
en-de	42.1	50.4	48.8	64.2	34.5	40.7	41.1	53.3	34.8	40.8	41.6	57.3
en-fr	37.2	46.5	52.9	62.2	37.1	46.5	52.9	62.0	37.1	46.2	53.0	57.4
en-es	37.7	47.5	53.0	67.0	37.6	47.3	53.2	65.5	37.8	47.8	53.1	68.3
en-ru	39.7	51.9	60.2	70.8	25.5	30.4	31.8	48.9	23.7	28.1	27.2	39.0
en-cs	27.5	33.2	34.8	45.2	26.1	31.7	34.2	41.9	25.1	30.1	32.3	46.8
en-nl	33.2	40.6	42.9	71.5	24.7	30.4	32.4	58.4	31.2	40.5	51.0	64.6
en-fi	24.3	29.5	29.3	38.8	21.2	25.7	27.7	41.6	19.2	23.0	23.5	31.7
en-ko	12.1	14.3	15.1	22.9	11.4	13.6	15.2	20.0	9.6	11.3	12.2	17.7
et-sk	9.8	11.2	12.4	66.0	9.2	10.4	11.6	50.0	7.3	8.7	9.5	61.0
	VM-S				VM-I				VM-U			
F1@	5	3	1	NN	5	3	1	NN	5	3	1	NN
en-de	36.9	43.4	42.6	63.7	36.3	42.6	42.9	58.8	36.3	42.5	42.8	58.5
en-fr	38.9	48.4	53.7	64.7	38.7	48.0	54.4	68.3	38.5	48.1	54.4	68.6
en-es	39.8	49.7	53.3	67.0	38.9	48.9	54.0	63.5	38.8	49.0	54.1	66.1
en-ru	29.8	37.3	38.4	50.9	28.4	34.8	34.8	47.4	25.0	30.5	29.2	45.4
en-cs	31.3	38.7	39.0	47.0	29.5	36.4	36.7	51.4	29.2	35.7	36.6	46.4
en-nl	34.1	44.1	54.2	61.6	33.6	43.7	55.5	67.3	33.6	43.6	55.5	63.1
en-fi	28.2	34.4	32.6	45.0	26.0	31.6	32.8	46.2	25.7	31.6	32.6	48.3
en-ko	19.8	24.7	30.4	42.2	13.6	16.2	18.8	22.6	11.1	13.3	14.2	9.3
et-sk	13.4	15.2	26.7	67.5	10.8	12.6	14.9	53.0	6.3	7.7	10.4	68.0

Table 8: Reported P (@5, @3, @1, and @NN (*k* selected by the classification neural network)) for the MUSE-S, MUSE-I, MUSE-U, VM-S, VM-I, and VM-U models evaluated with the MUSE evaluation dataset.

	MUSE-S				MUSE-I				MUSE-U			
P@	5	3	1	NN	5	3	1	NN	5	3	1	NN
en-de	31.3	45.7	83.9	56.8	25.7	36.9	70.7	49.3	25.9	37.0	71.5	57.8
en-fr	25.9	38.5	78.4	63.4	25.9	38.5	78.3	62.7	25.8	38.2	78.5	52.7
en-es	26.4	39.4	79.1	77.6	26.3	39.3	79.3	67.6	26.6	39.7	79.1	73.9
en-ru	26.3	40.1	79.2	66.4	16.9	23.5	41.9	43.2	15.7	21.7	35.8	37.3
en-cs	18.5	26.0	47.1	41.3	17.5	24.9	46.1	34.8	16.8	23.6	43.6	46.8
en-nl	26.8	41.0	87.1	70.3	19.8	31.3	68.4	63.0	21.2	31.4	67.7	62.4
en-fi	16.2	23.0	39.2	32.6	14.18	20.0	37.1	42.2	12.9	18.0	31.5	31.4
en-ko	7.6	10.3	17.5	28.9	7.2	9.8	17.6	50.0	6.0	8.1	14.1	28
et-sk	6.1	9.3	14.1	52.4	5.2	7.9	13.9	36.5	5.2	7.8	12.2	47.8
	VM-S				VM-I				VM-U			
P@	5	3	1	NN	5	3	1	NN	5	3	1	NN
en-de	27.5	39.4	73.3	60.7	27.0	38.6	73.7	54.5	27.0	38.5	73.7	55.3
en-fr	27.1	40.0	79.5	64.7	27.0	39.7	80.6	66.4	26.8	39.8	80.5	68.8
en-es	27.8	41.3	79.5	72.5	27.2	40.6	80.6	57.3	27.1	40.7	80.7	71.8
en-ru	19.8	28.8	50.5	42.3	18.8	26.8	45.8	39.3	16.6	23.6	38.5	41.6
en-cs	21.0	30.3	52.7	44.4	19.8	28.5	49.6	46.3	19.6	28.0	49.5	36.8
en-nl	22.7	34.2	71.9	57.5	22.8	33.9	73.6	61.6	22.4	33.8	73.6	62.8
en-fi	18.8	26.8	43.7	39.5	17.4	24.6	43.9	42.6	17.2	24.6	43.7	43.5
en-ko	12.5	17.8	35.2	46.2	8.6	11.7	21.8	24.6	7.0	9.6	16.5	13.9
et-sk	8.0	11.7	24.3	73.0	7.7	9.9	18.2	48.9	7.2	9.4	12.7	53.2