GARI: Graph Attention for Relative Isomorphism of Arabic Word Embeddings

Muhammad Asif Ali,[1] **Maha Alshmrani,**[1] **Jianbin Qin,**[2] **Yan Hu,**[1] **Di Wang**[1]
[1] King Abdullah University of Science and Technology, KSA
[2] Shenzhen University, China
{muhammadasif.ali; maha.shmrani; yan.hu; di.wang}@kaust.edu.sa; qinjianbin@szu.edu.cn

Abstract

Bilingual Lexical Induction (BLI) is a core challenge in NLP, it relies on the relative isomorphism of individual embedding spaces. Existing attempts aimed at controlling the relative isomorphism of different embedding spaces fail to incorporate the impact of semantically related words in the model training objective. To address this, we propose GARI that combines the distributional training objectives with multiple isomorphism losses guided by the graph attention network. GARI considers the impact of semantical variations of words in order to define the relative isomorphism of the embedding spaces. Experimental evaluation using the Arabic language data set shows that GARI outperforms the existing research by improving the average P@1 by a relative score of up to 40.95% and 76.80% for in-domain and domain mismatch settings respectively. We release the codes for GARI at `https://github.com/asif6827/GARI`.

1 Introduction

Bilingual Lexical Induction (BLI) is a key task in natural language processing. It aims at the automated construction of translation dictionaries from monolingual embedding spaces. BLI plays a significant role in multiple different natural language processing applications. For instance, the automated construction of lexical dictionaries plays a key role in the development of linguistic applications for low-resource languages, especially in cases where hand-crafted dictionaries are non-existent. Automated construction of high-quality dictionaries also helps in augmenting the end performance of multiple down-streaming tasks, including but not limited to: machine translation (Lample et al., 2018), information retrieval (Artetxe et al., 2018), cross-lingual transfers (Artetxe and Schwenk, 2019).

Earlier methods aimed at the construction of cross-lingual embeddings use linear and/or non-linear mapping functions in order to map the monolingual embeddings in a shared space. Some examples in this regard include retrieval criteria for bilingual mapping by Joulin et al. (2018) and BLI in non-isomorphic spaces by Patra et al. (2019).

These methods rely on the approximate isomorphism assumption, i.e., they assume that underlying monolingual embedding spaces are geometrically similar, which severely limits their use to closely related data sets originating from similar domains and/or languages exhibiting similar characteristics. The limitations of the mapping-based methods, especially their inability to handle data sets originating from different domains and languages exhibiting different characteristics has been identified by (Conneau et al., 2017; Søgaard et al., 2018; Glavas et al., 2019; Patra et al., 2019).

Some other noteworthy aspects identified in the literature that limit the end performance of the BLI systems, include: (a) algorithmic mismatch for independently trained monolingual embeddings, (b) different parameterization, (c) variable data sizes, (d) linguistic difference, etc., (Marie and Fujita, 2020; Marchisio et al., 2022).

In the recent past, there has been a shift in the training paradigm for the BLI models, i.e., instead of relying on pre-trained embeddings trained independently of each other, they use explicit isomorphism metrics along with the distributional training objective (Marchisio et al., 2022). However, a key limitation of these models is their inability to incorporate the impact of semantically related tokens (including their lexical variations) in controlling the relative isomorphism of different spaces. This is illustrated in Figure 1, where the left half of the figure shows a set of semantically related English words, e.g., {strong, rugged, and robust}. These

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 181–190
December 7, 2023 ©2023 Association for Computational Linguistics

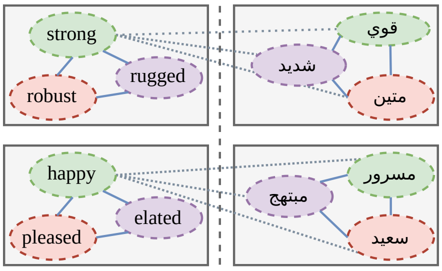

Figure 1: Some examples of semantically related tokens for English and their corresponding translations in the Arabic language.

words though lexically different share the same semantics. Correspondingly, their translations in the Arabic language: {شديد ، قوي ، متين} are also semantically related. We hypothesize that each language encompasses a list of such semantically related words that may be used interchangeably within a fixed context, and in order to control the relative isomorphism of corresponding embedding spaces the end model should be robust to incorporate these semantic variations in the model training objective.

To address these challenges, in this paper, we propose Graph Attention for Relative Isomorphism (GARI). GARI combines the distributional training objective with the isomorphism loss in a way that it incorporates the impact of semantically related words using graph attention, required to perform the end-task in a performance-enhanced way. We outline the key contributions of this work as follows:

1. We propose GARI that combines the distributional loss with graph attention-based isomorphism loss functions for effective BLI.

2. The graph attention part of the GARI leverages self-attention mechanism in order to attend over words that are semantically related to a given word.

3. We prove the effectiveness of GARI by comprehensive experimentation. Experimental evaluation shows, for the Arabic dataset, the GARI outperforms the existing research on relative isomorphism by 40.95% and 76.80% for in-domain and out-of-domain settings.

2 Related Work

There is an immense literature on BLI and controlling the relative isomorphism of the embedding spaces. In order to save space, we primarily limit the related work of this paper to one that is more relevant to our problem settings. We classify the related work into the following categories: (i) mapping pre-trained embeddings, (ii) combined training.

Mapping Pre-trained Embeddings. These methods rely on the use of linear and/or non-linear mappings to map the mono-lingual embeddings to a shared space.

Earlier works in this regard include principled bilingual dictionaries by Artetxe et al. (2016) that aim to learn bilingual mappings while preserving invariance for the monolingual analogy tasks. Artetxe et al. (2017) introduced a self-learning approach to relax the requirements for bilingual training seeds and/or parallel corpora. Alvarez-Melis and Jaakkola (2018) formulate the alignment as an optimal transport problem and employ Gromov-Wasserstein distance to compute the similarity of word pairs across different languages. Doval et al. (2018) propose additional transformation on top of the alignment step to force the synonyms towards a middle point for a better cross-lingual integration of the vector spaces. Jawanpuria et al. (2019) introduced language-specific rotations followed by a language-independent similarity in a common space. Similar to the word embedding methods, the application of the mapping-based methods to the contextualized embeddings include context-aware mapping by Aldarmaki and Diab (2019) and alignment of contextualized embeddings by Schuster et al. (2019).

Combined Training. On contrary to the mapping-based methods that rely on pre-trained embeddings, these methods use parallel data as input in order to jointly minimize the mono-lingual as well as cross-lingual training objectives. Duong et al. (2017) introduced methods for cross-lingual word embeddings for multiple languages in a unified vector space aimed to combine the strengths of different languages. Wang et al. (2019) addressed the limitations of joint training methods by combining them with mapping-based schemes for model training. For more details on the joint training methods refer to the survey paper by Ruder et al. (2019). Marchisio et al. (2022) introduced IsoVec which uses multiple different isomorphism metrics with skip-gram as the distributional training objective to control the isomorphism.

Nevertheless, we observe that existing methods for controlling the relative isomorphism ignore the impact of words that are semantically related to a given word, severely limits the ability of these methods to control the relative isomorphism of the embedding spaces.

3 Background

In this section, we first introduce the mathematical notation being used throughout the paper and formulate our problem definition. Later, we provide a quick background of the VecMap (Artetxe et al., 2018), a toolkit for mapping across different embedding spaces.

3.1 Notation

For this work, we use $\mathbf{X} \in \mathbf{R}^{m \times d}$ and $\mathbf{Y} \in \mathbf{R}^{n \times d}$ to represent the embedding matrices for the source and target languages with vocab size m and n respectively. d refers to the dimensionality of the embedding space. The embedding vectors for words, e.g., $\{x, y\}$ are represented by $\{\vec{\mathbf{x}}, \vec{\mathbf{y}}\}$. Like existing supervised works on controlling the relative isomorphism, e.g., IsoVec by Marchisio et al. (2022), we assume the availability of training seeds pairs for the source and target languages, denoted by: $\{(x_0, y_0), (x_1, y_1), ...(x_s, y_s)\}$.

3.2 The problem

In this work, we address a core challenge in BLI, i.e., we control the relative isomorphism of the embedding spaces. Specifically, we learn the distributional embeddings for the source language (i.e., Arabic) in a way:

1. The source embeddings $\mathbf{X}$ are geometrically isomorphic to the target embeddings $\mathbf{Y}$ (i.e., English language).
2. While learning isomorphic embeddings the $\mathbf{X}$ should incorporate the impact of the semantically related tokens (also their lexical variations) in $\mathbf{Y}$ in order to perform the end task in a performance-enhanced way.

3.3 VecMap toolkit

We use VecMap toolkit[1] for mapping across different embedding spaces. For this, we pre-process the embeddings using a process flow outlined by Zhang et al. (2019). The embeddings are unit-normed, mean-centered followed by another round of unit-normalization. For bi-lingual induction, we follow (Artetxe et al., 2018), i.e., whitening the spaces, and solving Procrustes. Later, we perform re-weighting, de-whitening, and mapping of translation pairs via nearest-neighbor retrieval (Artetxe et al., 2018).

[1] https://github.com/artetxem/vecmap

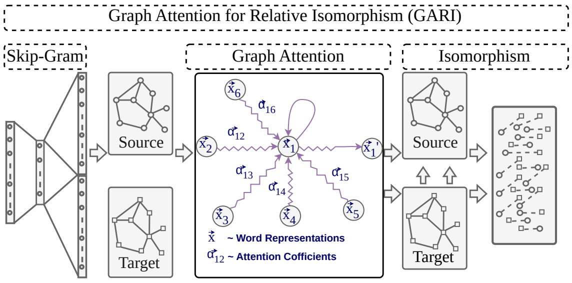

Figure 2: Graph Attention for Relative Isomorphism (GARI), the framework proposed in this work. It combines skip-gram and isomorphism loss (guided by graph attention).

4 Proposed Approach

In this paper, we address a core challenge in controlling the geometric isomorphism for source word embeddings relative to the target word embeddings, i.e., incorporate the impact of semantically coherent words in order to perform the end task in a performance augmented fashion. For this, we propose Graph Attention for Relative Isomorphism (GARI), shown in Figure 2. Details about the individual components of GARI are provided in the following subsections.

4.1 GARI

4.1.1 Overview

GARI aims to learn the source distributional embeddings $\mathbf{X}$ in a way that: (a) $\mathbf{X}$ is geometrically isomorphic to the target embeddings $\mathbf{Y}$, (b) $\mathbf{X}$ incorporates the impact of semantic variations of words in $\mathbf{Y}$. In order to control the geometric isomorphism of the embedding spaces in a robust way, GARI uses graph attention mechanism (to incorporate the impact of semantically related tokens) prior to using the isomorphism loss functions. Finally, it combines the distributional training objective and the isomorphism loss as the training objectives of the complete model.

4.1.2 Distributional Representation Learning

In order to learn the distributional embeddings for GARI, we use skip-gram with negative sampling (Mikolov et al., 2013). Its formulation is shown in Equation 1, i.e, embed a word close to its neighboring words within a fixed contextual window, while at the same time pushing it away from a list of random words selected from a noisy distribution.

$$\mathcal{L}_{Dis} = \log \sigma(\vec{\mathbf{x}}'^{\mathsf{T}}_{c_O} \vec{\mathbf{x}}_{c_I}) + \sum_{i=1}^{k} \mathbf{E}_{c_i \sim P_n(c)} \big[\log \sigma(-\vec{\mathbf{x}}'^{\mathsf{T}}_{c_i} \vec{\mathbf{x}}_{c_I}) \big] \tag{1}$$

Here $\vec{\mathbf{x}}_{c_O}$ and $\vec{\mathbf{x}}_{c_I}$ correspond to the output and input vector representations of the word c. k is the number of noisy samples and $\vec{\mathbf{x}}'_{c_i}$ is the embedding vector for the noisy word selected from the noisy distribution $P_n(c)$.

4.1.3 Semantic Relatedness

To incorporate the impact of semantically related words in controlling the relative isomorphism of the embedding spaces, GARI uses graph attention mechanism. The graph attention part of GARI works as follows: (a) create a graph $\mathbf{G}$ such that semantically related words end up being neighbors in the graph, (b) use graph attention mechanism for information sharing among neighbors in $\mathbf{G}$. The details about individual components are as follows:

(a) Graph Construction. The end goal of the graph construction step is to unite and/or combine the semantically related words helpful in controlling the relative isomorphism. Inputs for the graph construction process include: (i) pre-trained word2vec embeddings[2], and (ii) seed words corresponding to the target language, i.e., $\{y_0, y_1, ..., y_s\}$. The graph construction process proceeds as follows:

(a) Organize all seed words for the target language as a set of pairs: $\mathbf{P} = \{(y_0, y_1), (y_0, y_2), ..., (y_s, y_s)\}$, i.e., combinations of two words at a time.

(b) For each pair compute the cosine similarity score between the corresponding word2vec embedding vectors, and retain only the subset ($\mathbf{P}_{sub}$) with the cosine similarity score greater than a threshold (η).

(c) Finally, for the word pairs in $\mathbf{P}_{sub}$ construct a graph $\mathbf{G}$ by formulating edges between the word pairs.

Note, this setting for the graph construction allows each word to be surrounded by a set of semantically related neighbors which provides GARI with the provision to allow the propagation of information by using graph attention, as explained below.

(b) Graph Attention. The graph attention part of GARI follows a similar approach as proposed by Veličković et al. (2017). For a graph $\mathbf{G}$, the inputs to a single attention layer of the graph attention network include the source word representations $\{\vec{\mathbf{x}}_0, \vec{\mathbf{x}}_1, ..., \vec{\mathbf{x}}_s\}, \vec{\mathbf{x}}_i \in \mathbf{R}^d$, where s represent the number of words and d represents the dimensionality of the feature. It generates a new set of word representations $\{\vec{\mathbf{x}}'_0, \vec{\mathbf{x}}'_1, ..., \vec{\mathbf{x}}'_s\}, \vec{\mathbf{x}}'_i \in \mathbf{R}^{d'}$ as output. Its process flow is explained as follows:

Initially, a linear transformation is applied to all the words in $\mathbf{G}$ parameterized by a shared matrix $\mathbf{W} \in \mathbf{R}^{d \times d'}$. This is followed by using a shared attention mechanism $z : \mathbf{R}^{d'} \times \mathbf{R}^{d'} \rightarrow \mathbf{R}$ to compute the intermediate attention coefficients β_{ij} that incorporates the importance of word j on word i.

$$\beta_{ij} = z(\mathbf{W}\vec{\mathbf{x}}_i, \mathbf{W}\vec{\mathbf{x}}_j) \tag{2}$$

where the attention mechanism z is simply a single-layered feed-forward neural network with a weight vector $\vec{\mathbf{z}} \in \mathbf{R}^{d'}$ and ReLU non-linearity, as shown below:

$$z = \text{ReLU}\Big(\vec{\mathbf{z}}^T [\mathbf{W}\vec{\mathbf{x}}_i || \mathbf{W}\vec{\mathbf{x}}_j]\Big) \tag{3}$$

where $||$ is the concatenation operator. Note, the computation for β_{ij} implies each word will have an impact on every other word in $\mathbf{G}$, which is computationally inefficient and may inject noise in the model training. In order to avoid this, we perform masked attention, i.e., compute the attention weight β_{ij} for a fixed neighborhood of word i, i.e., $j \in \mathcal{N}_i$. We use the softmax function to compute the normalized attention coefficients α_{ij}, shown as follows:

$$\alpha_{ij} = \text{softmax}(\beta_{ij}) = \frac{\exp(\beta_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(\beta_{ik})} \tag{4}$$

Finally, we use the normalized coefficients in order to compute a linear combination of the corresponding word representations as the final output representation of each word as follows:

$$\vec{\mathbf{x}}'_i = \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{\mathbf{x}}_i\Big) \tag{5}$$

where σ is a nonlinearity.

[2]https://code.google.com/archive/p/word2vec/, trained using Google-News Corpus of 100 billion words.

Though Veličković et al. (2017) extend their work to a multi-head attention setting, but for GARI, we resort to one attention layer in order to avoid the computational overhead.

The intuitive explanation for the graph attention part of GARI is to surround each word by a set of semantically related words by forming edges in the graph and re-compute the representation of each word by propagating information from the neighbors in a way that it accommodates the impact of semantic variations of each word in an attentive way.

4.1.4 Isomorphism Loss

Finally, we use the output of the graph attention layer ($\mathbf{X}^{'}$) to compute the isomorphism loss for GARI relative to the target embeddings $\mathbf{Y}$. For this, we analyze the impact of multiple different variants of isomorphism loss functions referred to as $\mathcal{L}_{Iso}$. The details about different variants of the isomorphic loss functions are as follows:

L2 Loss ($\mathcal{L}_2$). We use L2-norm averaged over the number of words as our isomorphism metric. For N words, $\mathcal{L}_2$ is computed as:

$$\mathcal{L}_2 = \frac{1}{N}||\mathbf{X}^{'} - \mathbf{Y}||_2 \tag{6}$$

Orthogonal Procrustus Loss ($\mathcal{L}_{proc}$). The orthogonal Procrustes problem aims to find a linear transformation $\mathbf{W}_p$ to solve the following metric:

$$\mathcal{L}_{proc} = \underset{\mathbf{W}_p \in \mathbf{R}^{d \times d}, \mathbf{W}_p^T \mathbf{W}_p = I}{\arg\min} \frac{1}{N}||\mathbf{X}^{'}\mathbf{W}_p - \mathbf{Y}||_2 \tag{7}$$

For this, we use an existing solution $\mathbf{W}_p = \mathbf{QP}^T$ proposed by Schönemann (1966), where $\mathbf{P}\Sigma\mathbf{Q}^T$ is the singular value decomposition of the matrix $\mathbf{Y}^T\mathbf{X}^{'}$.

A variant of Procrustus Loss ($\mathcal{L}_{proc_{src}}$). For this, we follow the same process flow as outlined above for the Procrustus loss. The only difference is that we use pre-trained embeddings for the target words to initialize the corresponding embeddings for the source words for a given set of translation seed pairs. The end goal of this setting is to analyze the contribution of the pre-trained embeddings to guide the overall isomorphism of the source embeddings. Note that the initialized embeddings for the source words are updated during the model training.

4.2 The Complete Model

Finally, we combine the loss for the skip-gram distributional training objective with the isomorphism loss in order to come up with the loss function of GARI, as shown below:

$$\mathcal{L}_{GARI} = \gamma\mathcal{L}_{Dis} + (1-\gamma)\mathcal{L}_{Iso} \tag{8}$$

where, γ is the hyper-parameter controlling the contribution of individual losses in the model.

5 Experiments and Results

5.1 Datasets

For comparative analysis, we use the same data settings as primarily used by recent work, i.e., IsoVec by Marchisio et al. (2022). For the main experiments (section 5.4), we use the first 1 million lines of the newscrawl-2020 data set for the English and Arabic languages (Barrault et al., 2020). For the domain mismatch settings (section 6.1), we use 33.8 million lines of web-crawl data for the English language and newscrawl-2020 data for the Arabic language. For data pre-processing, we use Moses scripts[3] to process the English language data. For the Arabic language, we use NLTK tokenizer[4]. For performance evaluation, we used publically available train, dev, and test splits provided by MUSE (Conneau et al., 2017). We use word pairs numbered: 0-5000, 5001-6500, and 6501-8000 as train, test, and dev splits respectively. The train split is used for model training, and dev split for parameter tuning. The final results are computed over the test split.

5.2 Baseline Models

We use independently trained distributional embeddings for the source and target languages (without the isomorphism loss) as an immediate baseline. Other than this, we compare GARI against the existing best-performing model on relative isomorphism, i.e., IsoVec by Marchisio et al. (2022). Note, IsoVec follows a similar approach as that of GARI with the distinction that GARI uses graph attention as an additional layer to control the relative isomorphism of semantically relevant words. For IsoVec, we used publicly available implementation provided by the authors to generate the results for the Arabic language.

[3] github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer

[4] https://www.nltk.org/api/nltk.tokenize.html

Methodology	Avg. P@1
Baseline	15.58 ($\pm$ 0.8)
IsoVec (L2)	19.59 ($\pm$ 0.7)
IsoVec (Proc-L2)	20.03 ($\pm$ 0.5)
IsoVec (Proc-L2-Init)	<u>22.10 ($\pm$ 0.5)</u>
GARI ($\mathcal{L}_2$)	29.32 ($\pm$ 0.09)
GARI ($\mathcal{L}_{proc_{src}}$)	**31.15 ($\pm$ 0.07)**
GARI ($\mathcal{L}_{proc}$)	30.60 ($\pm$ 0.21)

Table 1: The results for the proposed model compared against the baseline model and existing state-of-the-art work on relative isomorphism, i.e., IsoVec (Marchisio et al., 2022).

Methodology	Avg. P@1
Baseline	14.70 ($\pm$ 0.7)
IsoVec (L2)	18.49 ($\pm$ 0.6)
IsoVec (Proc-L2)	18.80 ($\pm$ 0.7)
IsoVec (Proc-L2-Init)	<u>19.14 ($\pm$ 0.7)</u>
GARI ($\mathcal{L}_2$)	29.69 ($\pm$ 0.18)
GARI ($\mathcal{L}_{proc_{src}}$)	32.27 ($\pm$ 0.17)
GARI ($\mathcal{L}_{proc}$)	**33.84 ($\pm$ 0.02)**

Table 2: The results for the proposed approach under domain mismatch settings compared against the baseline model and existing state-of-the-art work on relative isomorphism, i.e., IsoVec (Marchisio et al., 2022).

5.3 Experimental Settings

In order to train the proposed model, i.e., GARI, we use Adam optimizer (Kingma and Ba, 2014) with learning rate = 0.001. In Equation 1, we set the value of k = 10. In Equation 8, we use the value of γ = 0.333. For the graph construction process, $\eta = 0.4$. We use English as the target language, and Arabic as the source language. Similar to the baseline models, we use VecMap toolkit (explained in Section 3.3) for mapping across different embedding spaces. We use average precision (i.e., P@1) as our evaluation metric, and report the mean (μ) and standard deviation (σ) of the results averaged over 5 runs of the experiment. All the experiments are performed using Intel Core-i9-10900 CPU and Nvidia 1080Ti GPUs.

5.4 Main Results

The results of GARI compared against the baseline models are shown in Table 1. We bold-face overall best scores and underline the previous state-of-the-art.

These results show that GARI outperforms the baseline models by a significant margin. The results of GARI with different isomorphism loss functions show that almost all the loss functions exhibit a similar performance with the loss ($\mathcal{L}_{proc_{src}}$) yielding overall best scores. Compared with the best performing baseline scores, GARI($\mathcal{L}_{proc_{src}}$) improves the average P@1 by approximately 40.95%. For the variants of GARI with loss functions $\mathcal{L}_2$ and $\mathcal{L}_{proc}$ the improvement in performance is 32.67% and 38.46% respectively. A relatively higher performance for the loss $\mathcal{L}_{proc_{src}}$ compared to $\mathcal{L}_{proc}$ shows that initializing the source embeddings with corresponding translation pairs from the target embeddings had a beneficial impact on the model training. Analyzing the variance of the results, we observe the variance of GARI is much lower compared to the variance of the baseline models. The worst-case variance of GARI is even less than half of the variance of the baseline models, which shows that GARI yields an overall stable performance across multiple re-runs of the experiments.

To summarize, these experiments show the essence of using the graph attention layers on controlling the relative isomorphism of the embedding spaces for BLI. We attribute the performance gained by GARI to the ability of the self-attention mechanism to appropriately accumulate information from semantically related words, which in turn plays a significant role in controlling the relative isomorphism of the embedding spaces.

6 Discussion

In this section, we perform a detailed analysis of GARI under different settings. For this, we perform analyses encompassing: (i) domain mismatch settings, (ii) correlation with isometric metrics, and (iii) error analysis.

6.1 Domain Mis-match

The results of our model for domain mismatch settings are shown in Table 2. Similar to the results for the main experiments, we also compare these results against the baseline models. We boldface the overall best scores with existing state-of-the-art underlined. These results show that GARI yields higher performance compared to the baseline models. The variants of GARI with loss $\mathcal{L}_2$, $\mathcal{L}_{proc}$ and $\mathcal{L}_{proc_{src}}$ outperform the best performing baseline model by 55.12%, 76.80%, and 68.60% respectively.

Comparing these results to the results for the main experiments (reported in Table 1), we observe that GARI yields a better performance for the domain mismatch settings relative to the in-domain setting. We attribute this performance improvement to: (a) the ability of GARI to capture and consolidate information from semantically relevant

	ES (↓)	ρ (↑)
GARI ($\mathcal{L}_2$)	80.99	0.46
GARI ($\mathcal{L}_{proc}$)	99.89	**0.56**
GARI ($\mathcal{L}_{proc_{src}}$)	**76.89**	0.45

Table 3: Analysis of different isometry metrics for GARI, i.e., , Eigenvector Similarity (ES) and Pearson's Correlation (ρ).

words even from different domains, (b) a relatively larger corpus for the target language (English) for domain mismatch settings. We notice that in contrast to the main experiments, for the domain mismatch settings loss the model GARI($\mathcal{L}_{proc}$) yields a better performance compared to GARI($\mathcal{L}_{proc_{src}}$). This shows that with the increase in the size of the data, the capability of the graph attention part of GARI to accumulate information about the semantically related words augments in a way that it even surpasses the model training with seed embeddings initialized.

Note, as illustrated in Section 1, domain mismatch is a key challenge for the BLI systems. Earlier research by Søgaard et al. (2018) shows that the majority of existing BLI systems perform poorly in inferring bilingual information from embeddings trained on different data domains. One key challenge that hinders the performance of these BLI systems is their inability to incorporate the impact of semantically related keywords and/or jargons peculiarly related to different domains. These words though belonging to different data domains have similar meanings and BLI systems should appropriately use this information for the model training. This makes GARI a better alternate, especially because of its provision to accumulate information about multiple different semantically related words using graph attention layers, as is also evident by a relatively higher performance of GARI compared to the baseline models.

6.2 Correlation with isometric metrics

Similar to the existing works on controlling the relative isomorphism of the embedding spaces (Marchisio et al., 2022), we compute isomorphism metrics for the results of GARI. We use two widely used metrics, namely: (i) Eigenvector similarity, (ii) Pearson's correlation. The computation details, and results of GARI for these metrics are as follows:

Eigenvector Similarity (ES). In order to compute the eigenvector similarity between the embedding spaces, we compute the Laplacian spectra of corresponding k-nearest neighbour graphs. We expect the graphs with similar structures to have similar eigenvalue spectra. For this, we follow the same settings as that of Søgaard et al. (2018). Given the seed pairs $\{x_0, x_1, ..., x_s\}$ and $\{y_0, y_1, ..., y_s\}$, we proceed as follows: (i) compute unweighted k-nearest neighbour graphs (i.e., G_X and G_Y), (ii) compute the graph Laplacians L_{G_X} and L_{G_Y}, where $L_{\mathrm{G}} = D_{\mathrm{G}} - A_{\mathrm{G}}$, (iii) compute the eigenvalues for each graph Laplacian, i.e., $\{\lambda_{L_{\mathrm{G}_X}}(i); \lambda_{L_{\mathrm{G}_Y}}(i)\}$ (iv) select $r = min(r_X, r_Y)$ where r_X is the maximum r such that the first r eigenvalues of L_{G_X} sum to less than 90% of the total sum of the eigenvalues. (v) depending upon the value of r, compute the eigenvector similarity as: $\sum_{i=1}^{r}(\lambda_{L_{\mathrm{G}_X}}(i) - \lambda_{L_{\mathrm{G}_Y}}(i))^2$.

The results for the eigenvector similarity measures should have an inverse correlation (↓) with the P@1. The results in the left column of Table 3 show that the variant of GARI with loss $\mathcal{L}_{proc_{src}}$ yields a higher performance which aligns with our findings for the main experiments in Table 1. However, the ES scores for the model with $\mathcal{L}_2$ and $\mathcal{L}_{proc}$ show irregular behavior. We expect the model with the loss $\mathcal{L}_{proc}$ to have a lower value for the ES score compared to $\mathcal{L}_2$, which is in contrast to our findings in Table 3.

Pearson's Correlation(ρ). In order to calculate the Pearson's correlation, we first compute the pairwise cosine similarity scores for the seed translation pairs, i.e., $\{\cos(x_0, x_1), \cos(x_0, x_2), ..., \cos(x_s, x_s)\}$, and $\{\cos(y_0, y_1), \cos(y_0, y_2), ..., \cos(y_s, y_s)\}$. Later, we compute the Pearson's correlation between the lists of cosine similarity scores. We expect the Pearson's correlation score to correlate positively (↑) with the average P@1.

The results in the right half of Table 3 show the Pearson's correlation scores for all the variants of GARI. These results show an unclear behaviour, with $\mathcal{L}_{proc}$ showing better performance compared to $\mathcal{L}_2$ and $\mathcal{L}_{proc_{src}}$. This is in contrast to the results for P@1 reported in Table 1, where $\mathcal{L}_{proc_{src}}$ shows a better performance compared to other models.

To summarize our findings for the isometric metrics, we observe that these results do not truly correlate with the average P@1. These findings are consistent with the earlier study IsoVec (Marchisio et al., 2022) that also emphasized the need for better isomorphism metrics in order to portray the correct picture of the degree of relative isomorphism of the

GARI (w/o Graph Attention)		
source	target$'$	target
الليزر	infrared	laser
الفهم	pronunciation	understanding
السجل	database	register
تلفاز	keyboards	tv
صدى	elated	echo
اربعة	three	four
زرقاء	foreboding	blue

Table 4: Example error cases for the model: GARI (w/o Graph Attention). The "target$'$" represents the model predictions, "target" represents the ground truth.

GARI ($\mathcal{L}_{proc_{src}}$)		
source	target$'$	target
ضرورية	vital	necessary
شمل	includes	included
القلم	pencil	pen
سماع	hear	hearing
المواهب	talents	talent
الفولاذ	metal	steel
المواصفات	certifications	specs

Table 5: Example error cases for GARI using the loss function $\mathcal{L}_{proc_{src}}$. The "target$'$" represents the model predictions, and the "target" represents the ground truth.

embedding spaces.

6.3 Error Analysis

In this section, we perform a detailed analysis of the error cases of GARI in order to know: (i) the performance improvement attributable to the graph attention part of the model, (ii) limitations of the GARI, and room for potential improvement. For this, we perform error analysis on two variants of GARI, i.e., with and without graph attention layer. All experiments are performed using the in-domain settings using the best-performing model, i.e., GARI ($\mathcal{L}_{proc_{src}}$). Details are as follows:

GARI (w/o Graph Attention). We initially analyze the error cases for the basic variant of GARI (without the graph attention layer) that have been corrected by the complete model. The core focus of this analysis is to look for the translation instances that benefit especially due to the graph attention mechanism. Note, for this analysis, we only include error cases that have incorrect prediction for the basic model (i.e., without graph attention) and are correctly classified by the complete model GARI.

While the graph attention layer is able to correct approximately 11% of the errors made by the basic variant of GARI, we observe almost 72% of the error cases belong to the noun category. One possible explanation in this regard is that the phenomenon of multiple senses is more dominant among the nouns in contrast to other parts-of-the speech, e.g., verbs and adjectives, which makes it harder to control their relative isomorphism (Ali et al., 2019). Some examples in this regard have been shown in Table 4. We also observe that the majority of the predictions made by the basic variant of GARI are not semantically related to the true target words, which clearly indicates the need for information sharing among the semantically related words required to control the relative isomorphism of the embedding spaces.

GARI (The Complete Model). The end goal of performing error analysis on the complete model is to dig out the potential reasons and/or understanding of the limitations of the proposed model. Note, we perform this analysis for the best-performing variant of GARI, i.e., with the loss $\mathcal{L}_{proc_{src}}$.

We randomly select a subset of 50 error cases for quantification. To our surprise, most of the errors (approximately 65%) made by GARI are either semantically very close to the true target word or a lexical variant of the true target word. Some examples in this regard have been shown in Table 5. These results clearly show the current performance of GARI is underrated primarily due to the use of a very strict evaluation criterion, (i.e., P@1). This calls for the need for better and more sophisticated mechanisms for the BLI systems in order to measure the relative isomorphism of the geometric spaces.

To summarize, the error analysis shows the essence of using graph attention in order to control the relative isomorphism of the embedding spaces. It helps in incorporating and/or accumulating information across semantically related words in order to perform the end task in a robust way.

7 Conclusions and Future Research

In this work, we propose Graph Attention for Relative Isomorphism (GARI). GARI incorporates the impact of semantically related words in order to control the relative isomorphism of geometric spaces in a performance-enhanced way. Experimental evaluation using the Arabic data set shows that GARI outperforms the existing state-of-the-art research by 40.95% and 76.80% for in-domain and domain mismatch settings. In the future, we will extend this research to deep contextualized embeddings and non-euclidean geometries.

8 Limitations

Some of the core limitations of the proposed approach are outlined as follows: (i) all the techniques have been developed assuming a Euclidean geometry for the underlying embedding spaces, its extension to non-Euclidean spaces are still unaddressed, (ii) the existing problem formulation is not defined for the deep contextualized embeddings.

Acknowledgements. Di Wang, Yan Hu and Muhammad Asif Ali are supported in part by the baseline funding BAS/1/1689-01-01, funding from the CRG grand URF/1/4663-01-01, FCC/1/1976-49-01 from CBRC and funding from the AI Initiative REI/1/4811-10-01 of King Abdullah University of Science and Technology (KAUST). Di Wang is also supported by the funding of the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI).

References

Hanan Aldarmaki and Mona Diab. 2019. Context-aware cross-lingual mapping. *arXiv preprint arXiv:1903.03243*.

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-Wasserstein alignment of word embedding spaces. Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. *arXiv preprint arXiv:1808.08780*.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. Association for Computational Linguistics.

Goran Glavas, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.

Pratik Jawanpuria, Arjun Balgovind, Anoop Kunchukuttan, and Bamdev Mishra. 2019. Learning multilingual word embeddings in latent metric space: a geometric approach. *Transactions of the Association for Computational Linguistics*, 7:107–120.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *arXiv preprint arXiv:1804.07745*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Kelly Marchisio, Neha Verma, Kevin Duh, and Philipp Koehn. 2022. Isovec: Controlling the relative isomorphism of word embedding spaces. *arXiv preprint arXiv:2210.05098*.

Benjamin Marie and Atsushi Fujita. 2020. Iterative training of unsupervised neural and statistical machine translation systems. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. *arXiv preprint arXiv:1908.06625*.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *arXiv preprint arXiv:1902.09492*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. *arXiv preprint arXiv:1805.03620*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. 2019. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. *arXiv preprint arXiv:1910.04708*.

Mozhi Zhang, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. 2019. Are girls neko or sh\= ojo? cross-lingual alignment of non-isomorphic embeddings with iterative normalization. *arXiv preprint arXiv:1906.01622*.