Locality Preserving Loss to Align Vector Spaces

Ashwinkumar Ganesan, Frank Ferraro, Tim Oates
Dept. Of Computer Science & Electrical Engineering (CSEE),
University Of Maryland Baltimore County (UMBC),
MD, USA - 21250
gashwin1@umbc.edu, ferraro@umbc.edu, oates@cs.umbc.edu

Abstract

We present a locality preserving loss (LPL) that improves the alignment between vector space representations (i.e., word or sentence embeddings) while separating (increasing distance between) uncorrelated representations as compared to the standard method that minimizes the mean squared error (MSE) only. The locality preserving loss optimizes the projection by maintaining the local neighborhood of embeddings that are found in the source, in the target domain as well. This reduces the overall size of the dataset required to the train model. We argue that vector space alignment (with MSE and LPL losses) acts as a regularizer in certain language-based classification tasks, leading to better accuracy than the baseline, especially when the size of the training set is small. We validate the effectiveness of LPL on a cross-lingual word alignment task, a natural language inference task, and a multilingual inference task.

1 Introduction

Over the last few years, vector space representations of words and sentences, extracted from encoders trained on a large text corpus, are primary components to model any natural language processing (NLP) task, especially while using neural or deep learning methods. This is because training stable word embeddings requires words to have high frequency in the corpus (Sahin et al., 2017). Hence, word embeddings generated from resource (corpus) constrained languages have a limited vocabulary. Similarly, corpora collected from domains such as healthcare or computational social sciences (Foulds, 2017) are typically small, reducing the model's capacity to generalize while learning to perform a task. Thus, it is common to initialize neural NLP models with pretrained word embeddings learned using word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) and fine tune sentence encoders like BERT (Devlin et al., 2018) in a number of tasks from part-of-speech tagging, named entity recognition, and machine translation to measuring textual similarity.

Let us consider two types of tasks, namely, vector space alignment where the purpose is to learn a mapping between two independently trained embeddings (e.g., crosslingual word alignment) and a classification task (e.g., natural language inference (NLI)). Learning bilingual word embedding models alleviates low resource problems by aligning embeddings from a source language that is rich in available text to a target language with a small corpus with limited vocabulary. Largely, recent work focuses on learning a linear mapping to align two embedding spaces by minimizing the mean squared error (MSE) between embeddings of words projected from the source domain and their counterparts in the target domain (Mikolov et al., 2013; Ruder et al., 2017). Minimizing MSE is useful when a large set of translated words (between source and target languages) is provided, but the mapping overfits when the parallel corpus is small or may require non-linear transformations (Søgaard et al., 2018). In order to reduce overfitting and improve word alignment, we propose an auxiliary loss function called locality preserving loss (LPL) that trains the model to align two sets of word embeddings while maintaining the local neighborhood structure around words in the source domain.

With classification tasks where there are two inputs (e.g., NLI), we show how the alignment between the two input subspace acts as regularizer, improving the model's accuracy on the task with MSE alone and when MSE and LPL are combined together.

Specifically, our main contributions are:

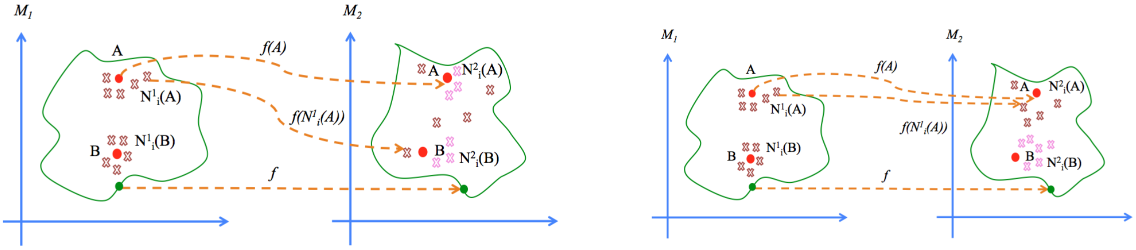

Figure 1: **Quality of alignment with different types of losses.** A, B are two words in two word embedding manifolds M_1 and M_2. f is the manifold alignment function while $N_i^1(A)$ and $N_i^1(B)$ are their respective neighbors in manifold M_1. $N_i^2(A)$ and $N_i^2(B)$ are their neighbors in manifold M_2. Figure (a) shows the alignment when trained with a MSE loss. The neighbors are distributed across the manifold due to overfitting. (b) shows alignment with a locality preserving loss (LPL) that reconstructs the original manifold in the target domain M_2 maintaining the its local structure.

1. We propose a new loss function called locality preserving loss (LPL) to improve vector space alignment and show how it can improve performance on crosslingual word alignment giving up to 4.1% (13.8% relative) improvement and multiple down-stream tasks such as SNLI with up to 8.9% (19.3% relative) improvement when trained with just 1000 samples.

2. We demonstrate how LPL reduces the size of the supervised set of labeled items required to train the model while maintaining equivalent performance.

3. We show how manifold alignment acts as a regularizer while performing natural language inference and that LPL when combined with MSE leads to higher overall accuracy.

2 Background & Related Work

Our work is inspired by a generalized autoencoder and locally linear embedding model.

2.1 Dimensionality Reduction & Manifold Alignment

Manifold learning methods represent these high dimensional datapoints in a lower dimensional space by extracting important features from the data, making it easier to cluster and search for like data points. The methods are broadly categorized into linear, such as Principal Component Analysis (PCA), and non-linear algorithms. Non-linear methods include multi-dimensional scaling (Cox and Cox, 2000, MDS), locally linear embedding (Roweis and Saul, 2000, LLE) and Laplacian eigenmaps (Belkin and Niyogi, 2002, LE). He and Niyogi (2004) compute the euclidean distance between points to construct an adjacency graph and create a linear map that preserves the neighborhood structure of each point in the manifold. Another popular approach to learn manifolds is an autoencoder where a self-reconstruction loss is used to train a neural network (Rumelhart et al., 1985). Vincent et al. (2008) design an autoencoder that is robust to noise by training it with a noisy input and then reconstructing the original noise-free input.

In locally linear embedding (LLE), the datapoints are assumed to have a linear relation with their neighbors. There are various ways to compute the neighbors of a datapoint like using Euclidean distance. The projection of each point is computed in a two step process. First, a reconstruction loss is utilized to learn the linear relation between a point and k neighbors (Roweis and Saul, 2000),

$$L_{\text{reconstruct}} = \sum_i \left\| X_i - \sum_{j \in N_i} W_{ij} X_j \right\|^2, \qquad (1)$$

where X_i is the datapoint and the X_js represent the neighbors. An additional constraint is imposed on the weights ($\sum_{ij} W_{ij} = 1$) to make the transform scale invariant. In (1) the weights W are an $N \times K$ matrix in a dataset of N points (i.e., each point has its own weights). Learning a transformation ϕ therefore requires learning W:

$$\phi(Y) = \sum_i \left\| Y_i - \sum_{j \in N_i} W_{ij} Y_j \right\|^2, \qquad (2)$$

Y_i is a projection for X_i (typically with reduced dimensions). Wang et al. (2014) extend the autoencoder model by modifying the reconstruction loss to use neighbors similar to the non-linear methods described above:

$$\rho(X_i) = \sum_{j \in N_i} S_{ij} L(X_i, X_j) \tag{3}$$

$L(X_i, X_j)$ is the loss between point X_i and associated point (i.e., neighbor) X_j and S_{ij} is a weight that represents the relationship between points. For example, S_{ij} can be 1 when they are nearest neighbors and 0 when they are not. Depending on the type of non-linear method (described above) retrofitted into the model, L can be various functions.

Benaim and Wolf (2017) utilize a GAN to learn a unidirectional mapping. The total loss applied to train the generator is a combination of different losses, namely, an adversarial loss, a cyclic constraint (inspired by Zhu et al. (2017)), MSE and an additional distance constraint where the distance between the point and its neighbors in the source domain are maintained in the target domain. Similarly, Conneau et al. (2017) learn to translate words without any parallel data with a GAN that optimizes a cross domain similarity scale to resolve the hubness problem (Dinu et al., 2014).

These methods are the foundation to learn a mapping between two lower dimensional spaces (manifold alignment, fig. 1). Wang et al. (2011) propose a manifold alignment method that preserves the local similarity between points in the manifold being transformed and the correspondence between points that are common to both manifolds. Boucher et al. (2015) replace the manifold alignment algorithm that uses the nearest neighbor graph with a low rank alignment. Cui et al. (2014) align two manifolds without any pairwise data (unsupervised) by assuming the structure of the lower dimension manifolds are similar.

2.2 Cross Embedding Word Alignment

One way to alleviate the problem of limited text is to align words between two languages that have similar meanings to initialize the embeddings for unknown words. Mikolov et al. (2013) learn a linear mapping by optimizing the MSE between the source and target language. Xing et al. (2015) improve the mapping by adding an orthogonal constraint to the weights. In BilBOWA (Gouws et al., 2015), the cross-lingual mappings are learned by training monolingual representations for the source and target language and additional training with a cross-lingual objective on a sentence aligned corpus. Faruqui et al. (2014) use external information to adjust the existing word embeddings. Artetxe et al. (2017) reduce the need for a parallel word corpus by iteratively inducing a dictionary. Faruqui and Dyer (2014) learn to map the embeddings to a joint space with canonical correlation analysis (CCA). Lu et al. (2015) extend the prior using deep canonical correlation analysis. Our work is similar to Bollegala et al. (2017) where the meta-embedding (a common embedding space) for different vector representations in generated using a locally linear embedding (LLE) which preserves the locality. One drawback though is that LLE does not learn a single mapping between the source and target vector spaces. A linear mapping between a word and its neighbor is learned for each new word and the meta-embedding for each word in vocabulary is learned every time new words are added to the vocabulary. Nakashole (2018) propose NORMA that uses neighborhood sensitive maps where the neighbors are learned rather than extracted from the existing embedding space.

3 Locality Preserving Alignment (LPA)

3.1 Locality Preservation Criteria

The locality preserving loss (LPL, (5)) is based on an important assumption about the source manifold: for a pre-defined neighborhood of k points (k is chosen manually) in the source embedding space we assume points are "close" to a given point such that it can be reconstructed using a linear map of its neighbors. This assumption is similar to that made in locally linear embedding (Roweis and Saul, 2000). The above principle can be applied to the target space in order to learn a reverse mapping too.

3.2 Preliminaries

As individual embeddings can represent words or sentences, we call each individual embedding a *unit*. Consider two manifolds $M^s \in \mathbb{R}^{n \times d}$ (source domain) and $M^t \in \mathbb{R}^{m \times d}$ (target domain), that are vector space representations of units. We do not make assumptions on the methods used to learn each manifold; they may be different. We also do not assume they share a prima facie common lexical vocabulary. For example, M^s can be created using a standard distributed representation method like word2vec (Mikolov et al., 2013) and consists of English word embeddings while M^t is created using GloVe (Pennington et al., 2014) and contains Italian embeddings. Let V^s and V^t be the respective vocabularies (collection of units)

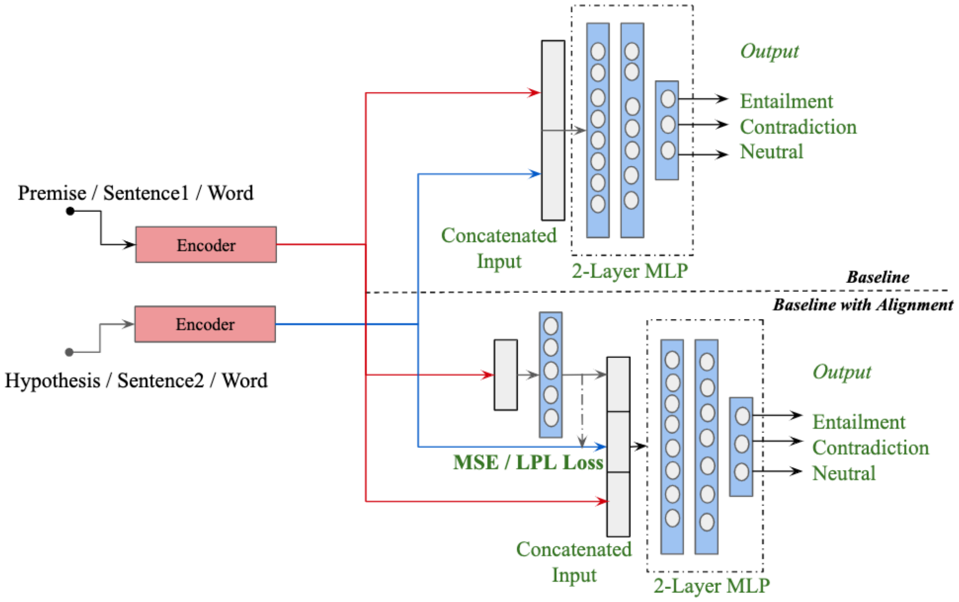

Figure 2: **Use of alignment loss for the NLI task.** The pipeline consists of a 3-layer MLP used to classify sentence pairs into *entailment*, *contradiction* and *neutral*. The premise and hypothesis subspaces are aligned using a MSE and LPL loss that is then added to the concatenated input to train the classifier. A δ hyperparameter configured for each label controls provides the ability to perform *alignment* for *entailment* and *contradiction* while performing *divergence* for *neutral* input pairs.

of the two manifolds. Hence $V^s = \{w_1^s \ldots w_n^s\}$ and $V^t = \{w_1^t .. w_m^t\}$ are sets of units in each vocabulary of size n and m. The distributed representations of the units in each manifold are $M^s = \{m_1^s \ldots m_n^s\}$ and $M^t = \{m_1^t \ldots m_n^t\}$.

While we do not assume that V^t and V^s must have common items, we do assume that there is some set of unit pairs that are connected by some consistent relationship. Let $V^p = \{w_1^p \ldots w_c^p\}$ be the set of the unit pairs; we consider V^p a supervised training set, or perhaps derived from a parallel corpus. For example, in crosslingual word alignment this consistent relationship is whether one word can be translated as another; in natural language inference, the relationship is whether one sentence entails the other (the second must logically follow from the first). We assume this common set V^p is much smaller than the individual vocabularies ($c << m$ and $c << n$). The mapping (manifold alignment) function is f.

In this paper, we experiment with two types of tasks i.e. cross-lingual word alignment and natural language inference. In cross-lingual word alignment, V^t and V^s represent the source and target vocabularies, V^p bilingual dictionary, M^t and M^s are the target and source manifold. f with θ_f parameters is a linear projection with a single weight matrix W. For NLI, V^t and V^s target and source sentences with M^t and M^s being their manifolds. f is 3-layer MLP.

3.3 Locality Preserving Loss (LPL)

We use a mapping function $f : M^s \rightarrow M^t$ to align the manifold M^s to M^t. The exact structure of f is task-specific: for example, in our experiments f is a linear function for crosslingual word alignment and it is a single layer neural network (non-linear mapping) for NLI. The mapping is optimized using three loss functions: an orthogonal transform (Xing et al., 2015) represented as L_{ortho} (i.e. constrain $W^{-1} = W^T$ to be ; mean squared error L_{mse} (eq. 4); and locality preserving loss (LPL) as L_{lpl} (eq. 5).

The standard loss function to align two manifolds is mean squared error (MSE) (Ruder et al., 2017; Artetxe et al., 2016),

$$L_{\text{mse}} = \sum_{i \in V^p} L_{\text{mse}}^i = \sum_{i \in V^p} \overbrace{\|f(m_i^s) - m_i^t\|_2^2}^{L_{\text{mse}}^i}, \quad (4)$$

which minimizes the distance between the unit's representation in M_t (the target manifold) and projected vector from M_s. The function $f(m_i^s)$ has learnable parameters θ_f. MSE can lead to an optimal alignment when there is a large number of units in the parallel corpus to train the mapping between the two manifolds (Ruder et al., 2017). However, when the parallel corpus V^p is small, the mapping is prone to overfitting (Glavas et al., 2019).

Locality preserving loss (LPL: eq. 5) optimizes the mapping f to project a unit together with its neighbors. For a small neighborhood of k units, the source representation of unit w_i^s is assumed to be a linear combination of its source neighbors. We represent this small neighborhood (of the source embedding m_i^s of word w_i^s) with $N_k(m_i^s)$, and we compute the local linear reconstruction using W_{ij}, a learned weight associated with each word in the neighborhood of the current word $N_k(m_i^s)$. LPL requires that the projected source embedding $f(m_i^s)$ is an average of all the projected vectors of its neighbors $f(m_j^s)$. Formally, for a particular common item i, LPL at i minimizes

$$L_{\text{lpl}}^i = \left\| m_i^t - \sum_{m_j^s \in N_k(m_i^s)} W_{ij} \cdot f(m_j^s) \right\|^2 \tag{5}$$

with $L_{\text{lpl}} = \sum_{m_i^s, m_i^t \in V^p} L_{\text{lpl}}^i$. Intuitively, W represents the relation between a word and its neighbors in the source domain. We learn it by minimizing the LLE-inspired loss. For a common i this is

$$L_{\text{lle}}^i = \left\| m_i^s - \sum_{m_j^s \in N_k(m_i^s)} W_{ij} \cdot m_j^s \right\|^2 \tag{6}$$

with $L_{\text{lle}} = \sum_{m_i^s \in V^p} L_{\text{lle}}^i$. The weights W are subject to the constraint $\sum W_{ij} = 1$, making the projected embeddings invariant to scaling (Roweis and Saul, 2000). We can formalize this with an objective $L_{\text{ortho}} = WW^\intercal - I$. LPL reduces overfitting because the mapping function f does not simply learn the mapping between unit embeddings in the parallel corpus: it also optimizes for a projection of the unit's neighbors that are not part of the parallel corpus—effectively expanding the size of the training set by the factor k.

3.3.1 Model Training with Locality Preserving Alignment

The total supervised loss becomes:

$$L_{\text{sup}} = L_{\text{mse}}(\theta_f) + \beta * L_{\text{lpl}}(\theta_f, W) + L_{\text{ortho}}(W) \tag{7}$$

We introduce a constant β to allow control over the contribution of LPL to the total loss.

Although we minimize total loss (7), shown explicily with variable dependence, the optimization can be unstable as there are two sets of independent parameters W and θ_f representing different relationships between datapoints. To reduce the instability, we split the training into two phases. In the first phase, W is learned by minimizing L_{lle} alone and the weights are frozen. Once W is learned, L_{mse} and L_{lpl} are minimized while keeping W fixed.

One key difference between our work and Artetxe et al. (2016) is that they optimize the mapping function by taking the singular vector decomposition (SVD) of the squared loss while we use gradient descent to find optimal values of θ_f. As our experimental results show, while both can empirically advantageous, our work allows LPL to be easily added as just another term in the loss function: with the exception of the alternating optimization of W, our approach does not need special optimization updates to be derived.

3.4 Alignment as Regularization

MSE and LPL can be used to align two vector spaces: in particular, we show that the objectives can align two subspaces in the *same* manifold. When combined with cross entropy loss in a classification task, this subspace alignment effectively acts as a regularizer.

Fig. 2 shows an example architecture where alignment is used as a regularizer for the NLI task. The architecture contains a two layer MLP used to perform language inference, i.e., to predict if the given sentence pairs are *entailed*, *contradictory* or *neutral*. The input to the network is a pair of sentence vectors. The initial representations are generated from any sentence/language encoder, e.g., from BERT. The source/sentence1/premise embeddings are first projected to the hypothesis space. The projected vector is then concatenated with the original pair of embeddings and given as input to the network. The alignment losses (MSE and LPL) are computed between the projected premise and original hypothesis embeddings. If the baseline network is optimized with cross entropy (CE) loss to predict label y_i, the total loss becomes:

$$L_{\text{total}} = \gamma \sum_i \delta_{y_i}(L_{\text{mse}}^i + L_{\text{lpl}}^i) + CE_{y_i} \tag{8}$$

where γ is an empirical hyperparameter that controls the impact of the loss (learning rate). Thus, the loss 8 is an extension of 7 for a classification task but without L_{ortho}, which is not applied as f is a 3-layer MLP (non-linear mapping) and the

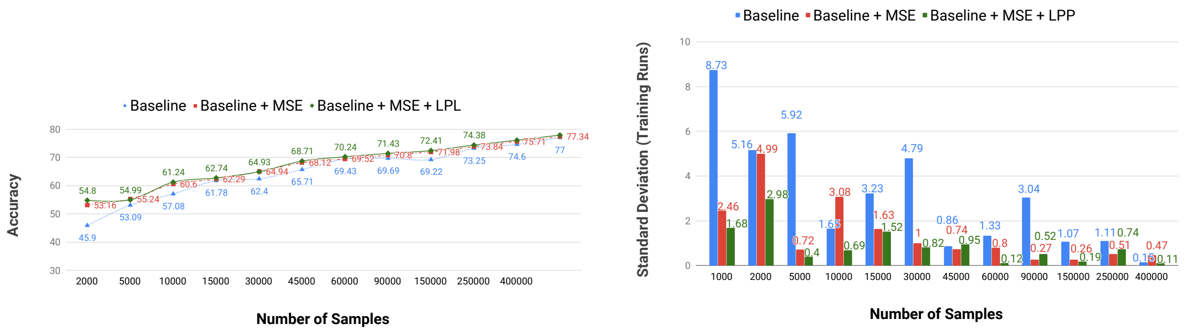

Figure 3: Accuracy of alignment regularization on SNLI. The left graph shows the accuracy, averaged across 3 runs, for differing size of training samples (total: 500K). The right chart shows accuracy standard deviation for the baseline, baseline + MSE and baseline + MSE + LPL models: LPL yields more consistently optimal systems.

Method	Trans.	Optim.	EN-IT	EN-DE	EN-FI	EN-ES
MSE, Train T→S (Shigeto et al., 2015)	S1, S2	Linear	41.53	43.07	31.04	33.73
MSE (Artetxe et al., 2016)	S0, S2	Linear	39.27	41.87	30.62	31.40
MSE: IS (Smith et al., 2017)	S0, S2, S5	Linear	41.53	43.07	31.04	33.73
MSE: NN (Artetxe et al., 2018)	S0-S5	Linear	44.00	44.27	32.94	36.53
MSE: IS (Artetxe et al., 2018)	S0-S5	Linear	**45.27**	44.13	32.94	**36.60**
MSE	S0, S2	SGD	39.67	45.47	29.42	35.3
LPA+MSE: CSLS	S0, S2	SGD	43.33	**46.07**	**33.50**	35.13

(a) We compare our method (bottom row: LPA) on cross-lingual word alignment. In comparison to Artetxe et al. (2018), we use cross-domain similarity local scaling (CSLS) (Conneau et al., 2017) to retrieve the translated word. Method lists different losses/methods used to learn the projection: *NN* is nearest neighbor search while *IS* is inverted softmax. Many mapping methods use additional transformation steps.

Trans.	Desc.	Backprop?
S0	Embedding normalization (unit / center)	Yes
S1	Whitening	No
S2	Orthogonal Mapping	Yes
S3	Re-weighting	No
S4	De-Whitening	No
S5	Dimensionality Reduction	No

(b) A map of various transformations that can be performed as described in Artetxe et al. (2018). We indicate which steps can easily be combined with backpropagation.

Table 1: The accuracy of the locality preserving method. Table 1a lists 6 high-performing supervised/semi-supervised baselines; table 1b lists the transformations used in these methods and how easily those transformations can be used with back-propagation. Notice that our method uses transformations amenable with back-propagation. In 1a, the first five baselines rely on algebraic updates while our method works nicely with SGD: we include the sixth row (MSE via SGD) to illustrate the comparative performance gain we obtain.

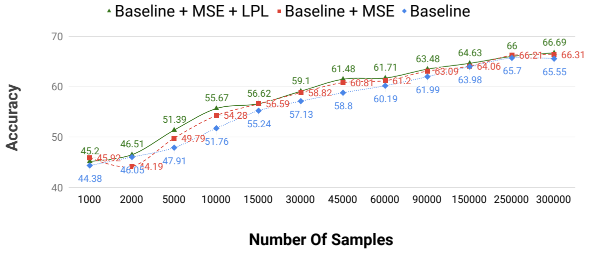

Figure 4: Accuracy of alignment regularization on MNLI dataset with a varying number of *matched* in-genre samples (total: 300K samples).

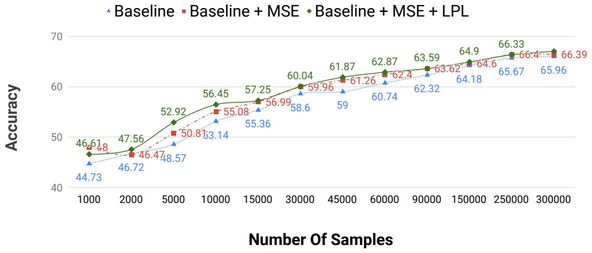

Figure 5: Accuracy of alignment regularization on MNLI dataset with a varying number of *mismatched* out-of-genre samples (total: 300K samples).

$WW^\intercal = I$ constraint for each layer's weights cannot be guaranteed. The alignment loss becomes a vehicle to bias the model based upon our knowledge of the task, forcing a specific behavior on the network. The behavior can be controlled with δ, which can be a positive or negative value specific to each label. A positive δ optimizes the network to align the embeddings while a negative δ is a divergence loss. In NLI we assign a constant scalar to all samples with a specific label (i.e., 100 for entailment, 1.0 for contradiction and -5.0 for neutral). The scalars have been assigned while optimizing network hyper-parameters. As the optimizer minimizes the loss, a divergence loss tends to ∞; in practice, the negative loss has a threshold.

4 Experiment Results & Analysis

We demonstrate the effectiveness of the locality preserving alignment (LPA) on two types of tasks: natural language inference and crosslingual word alignment. In order to compute local neighborhoods, as needed for, e.g., (5), we build a standard KD-Tree with Euclidean distance.

4.1 Natural Language Inference

To test the effectiveness of alignment as a regularizer, a 2-layer MLP is used as shown in Figure 2, we measure the change in accuracy with respect to this baseline. An additional single layer network is utilized to perform the alignment with premise and hypothesis spaces. We experiment the impact of the loss function on two datasets: the Stanford natural language inference (SNLI) (Bowman et al., 2015) and the multigenre natural language inference dataset (MNLI) (Williams et al., 2018). SNLI consists of 500K sentence pairs while MNLI contains about 433k pairs. The MNLI dataset contains two test datasets. The *matched* dataset contains sentences that are sampled from the same genres as the training samples while *mismatched* samples test the models accuracy for out of genre text.

Figures 3(a), 4, and 5 show the accuracy of the models when optimized with a standard cross-entropy loss (baseline), with additional MSE for alignment and finally with MSE and LPL combined. The accuracy is measured when the size of the training set is reduced. The reduced datasets are created by randomly sampling the required number from the entire dataset. The graphs show that an alignment loss consistently boosts accuracy of the model with respect to the baseline. It also shows that LPL, when combined with MSE, is able to provide higher gains as compared to the model being optimized with MSE alone. Also, the difference in accuracy is larger (as compared to baseline) when the number of training samples are small and reduces as the training set becomes larger. This is because we calculate the neighbors for each premise from the training dataset only rather than any external text like Wikipedia (i.e., generate embeddings for Wikipedia sentences and then use them as neighbors). As the training size increases LPL has diminishing returns, as the neighbors tend to be part of the training pairs themselves. Figure 3(b) is a plot of the standard deviation of accuracy across 3 runs (training data being randomly sampled each time). We clearly observe that a model regularized with MSE and LPL are more likely to reach optimal parameters consistently.

Manifold	Nearest Neighbors
Source (M_1)	nt4, 95/98/nt, nt/2000, nt/2000/xp, windows98
Target (M_2)	winzozz, mac, nt, osx, msdos
Aligned ($f(M_1)$)	winzozz, nt4, ntfs, mac, 95/98/nt, nt, osx, msdos

Table 2: Neighbors of the word “windows” in source domain (English), target domain (Italian) and the combined vector space with both English & Italian vocabulary. The **Aligned** neighborhood contains a mix of the English and Italian words, not just the translation.

4.2 Crosslingual Word Alignment

The cross lingual word alignment dataset is from Dinu et al. (2014). The dataset is extracted from the Europarl corpus[1] and consists of word pairs split into training (5k pairs) and test (1.5k pairs) respectively. From the 5K word pairs available for training only 3K pairs are used to train the model with LPA and an additional 150 pairs are used as the validation set (in case of Finnish 2.5K pairs are used). This is a reduced set in comparison to the models in table 1a that are trained with all pairs.

Table 2 shows the neighbors for the word “windows” from the source embedding (English) and the target embedding (Italian). Compared to previous methods that look at explicit mapping of points between the two spaces, LPA tries to maintain the relations between words and their neighbors in the source domain while projecting them into the target domain. In this example, the word “nt/2000” is not a part of the supervised pairs available and will not have an explicit projection in the target domain to be optimized without a locality preserving loss.

Along with the mapping methods in Table 1a, previous methods also apply additional pre/post processing tranforms on the word embeddings as documented in Artetxe et al. (2018) (described in table 1b). Cross-domain similarity local scaling (CSLS) (Conneau et al., 2017) is used to retrieve the translated word. Table 1a shows the accuracy of our approach in comparison to other methods.

[1] http://opus.lingfil.uu.se/

ID	Sentence
P	Family members standing outside a home.
H	A family is standing outside.
1P	People standing outside of a building.
1H	One person is sitting inside.
2P	Airline workers standing under a plane.
2H	People are standing under the plane.
3P	A group of four children dancing in a backyard.
3H	A group of children are outside.
4P	People standing outside of a building.
4H	One person is sitting inside.
5P	A family doing a picnic in the park.
5H	A family is eating outside.
6P	Airline workers standing under a plane.
6H	People are standing under the plane.

Table 3: **Nearest neighbors extracted from SNLI classifier for a sentence pair representation.** P and H are the sample premise and hypothesis pair. The original label is *Entailment*. (nP, nH) are the nearest neighbors of this sentence pair's representation from the penultimate layer of each classifier i.e. baseline, MSE and MSE+LPL. 1 & 2 are nearest neighbors from the baseline, 3 & 4 are when trained with MSE only and 5 & 6 are when trained with MSE and LPL.

The accuracy of our proposed approach is better or comparable to previous methods that use similar numbers of transforms. It is similar to Artetxe et al. (2018) while having fewer preprocessing steps. This is because we choose to optimize using gradient descent as compared to a matrix factorization approach. Thus, our implementation of Artetxe et al. (2016) (MSE Loss only) underperforms in comparison to the original baseline while giving improvements with LPA. Gradient descent has been adopted in this case because the loss function can be easily adopted by any neural network architecture in the future as compared to matrix factorization methods that will force architectures in the future to use a two-step training process.

4.3 Discussion

Table 3 shows the 2 nearest neighbors for a premise-hypothesis pair (P, H) taken from each classifier i.e. baseline, MSE only and MSE + LPL after they are trained (the dataset size is small at just 2000 samples). Since, NLI is a reasoning task, the sentence pair representations ideally will cluster around a pattern that represents *Entailment* or *Contradiction* or *Neutral*. Instead what is observed is that when the samples are limited, sentence pair representations have NNs that are syntactically similar (NNs 1 and 2) for the baseline model. The predicted labels for the NN pairs are not clustered into entailment but are a combination of all 3. This problem is reduced for models trained with MSE and MSE + LPL (NNs 3 and 4 for MSE, NNs 5 and 6 for MSE + LPL). The predicted labels of the NNs are clustered into entailment only. The sentence pair representations clusters containing a single label suggest the models are better at extracting a pattern for entailment (and improving the model's ability to reason). This semantic clustering of representations can be attributed to the initial alignment (or divergence) between the premise and hypothesis with additional locality preserving loss to increase the training size.

Apart from better accuracy when the training dataset is small, in figures 3, 4, and 5, we observe that accuracy of the models trained with alignment loss using MSE only and another in combination with LPL converge as number of training samples increase. This happens because of the way k-nearest neighbor (k-NN) is computed for each embedding in the source domain. We use BERT to generate the embedding of each sentence in the SNLI and MNLI dataset. But BERT itself is trained on millions of sentences from Wikipedia and Book Corpus. Searching for k-NN embeddings for each sentence from this dataset (for each sentence in the training sample) is computationally difficult. In order to make the k-NN search tractable, neighbors are extracted from the dataset itself (500K sentences in SNLI and 300K sentences in MNLI). This impacts the overall improvement in accuracy using LPL as it is not a perfect reconstruction of the datapoint (using its neighbors). Initially when the dataset is small the neighbors are unique. As the dataset size increases, the unique neighbors reduce and are subsumed by the overall supervised dataset (hence MSE begins to perform better). Thus, the impact of LPL reduces as the number of unique neighbors decreases and the entire dataset is used to train the model. This is unlikely to happen when NNs from a larger text corpus (unrelated to task) are used to reconstruct the local manifold.

5 Conclusion

In this paper, we introduce a new loss locality preserving loss (LPL) function that learns a linear relation between the given word and its neighbors and then utilizes it to learn a mapping for the neighborhood words that are not a part of the word pairs (parallel corpus). Also, we show how the results of the method are comparable to current supervised models while requiring a reduced set of word pairs to train on. Additionally, the same alignment loss is applied as a regularizer in a classification task like NLI to demonstrate how it can improve the accuracy of the model over the baseline.

References

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)*, volume 1, 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 Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591.

Sagie Benaim and Lior Wolf. 2017. One-sided unsupervised domain mapping. In *Advances in Neural Information Processing Systems*, pages 752–762.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2017. Think globally, embed locally—locally linear meta-embedding of words. *arXiv preprint arXiv:1709.06671*.

Thomas Boucher, CJ Carey, Sridhar Mahadevan, and Melinda Darby Dyar. 2015. Aligning mixed manifolds. In *AAAI*, pages 2511–2517.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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

Trevor F Cox and Michael AA Cox. 2000. *Multidimensional scaling*. CRC press.

Zhen Cui, Hong Chang, Shiguang Shan, and Xilin Chen. 2014. Generalized unsupervised manifold alignment. In *Advances in Neural Information Processing Systems*, pages 2429–2437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471.

James Foulds. 2017. Mixed membership word embeddings for computational social science. *arXiv preprint arXiv:1705.07368*.

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*.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *International Conference on Machine Learning*, pages 748–756.

Xiaofei He and Partha Niyogi. 2004. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160.

Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 250–256.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ndapandula Nakashole. 2018. Norma: Neighborhood sensitive maps for multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 512–522.

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.

Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

Sebastian Ruder, Ivan Vulic, and Anders Søgaard. 2017. A survey of cross-lingual embedding models. *CoRR, abs/1706.04902*.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, DTIC Document.

Cem Safak Sahin, Rajmonda S Caceres, Brandon Oselio, and William M Campbell. 2017. Consistent alignment of word embedding models. *arXiv preprint arXiv:1702.07680*.

Yutaro Shigeto, Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. 2015. Ridge regression, hubness, and zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 135–151. Springer.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Chang Wang, Peter Krafft, and Sridhar Mahadevan. 2011. Manifold alignment.

Wei Wang, Yan Huang, Yizhou Wang, and Liang Wang. 2014. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 490–497.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.