

Locality Preserving Loss: Neighbors that *Live* together, *Align* together

Ashwinkumar Ganesan, Francis 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 embeddings while separating uncorrelated representations. Given two pretrained embedding manifolds, LPL optimizes a model to project an embedding and maintain its local neighborhood while aligning one manifold to another. This reduces the overall size of the dataset required to align the two in tasks such as crosslingual word alignment. We show that the LPL-based alignment between input vector spaces acts as a regularizer, leading to better and consistent accuracy than the baseline, especially when the size of the training set is small. We demonstrate the effectiveness of LPL-optimized alignment on semantic text similarity (STS), natural language inference (SNLI), multi-genre language inference (MNLI) and cross-lingual word alignment (CLA) showing consistent improvements, finding up to 16% improvement over our baseline in lower resource settings.¹

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. Neural NLP models can be initialized with pretrained word embeddings learned using *word2vec* (Mikolov et al., 2013) or *GloVe* (Pennington et al., 2014) and fine-tuned sentence encoders like BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019). They show state-of-the-art performance in a number of tasks from part-of-speech tagging, named entity recognition, and machine translation to measuring textual similarity

(Wang et al., 2018). BERT’s success has spawned research dedicated to understanding (Rogers et al., 2020) and reducing parameters in transformer architectures (Lan et al., 2019). Despite these successes, supervised transfer learning is not a panacea. Models based on pretrained word embeddings (for bilingual induction) or BERT-based models require a large parallel corpus to train on. Can we reduce the number of training samples even further?

We approach this problem by proposing a framework that exploits the inherent relationship between word or sentence representations in their *pretrained manifolds*. These relationships help train the model with fewer samples, since each training sample represents a group of instances.

We consider three types of tasks: (1) a regression task such as semantic text similarity; (2) a classification task (e.g., natural language inference (NLI)); and (3) vector space alignment, where the purpose is to learn a mapping between two independently trained embeddings (e.g., crosslingual word alignment) 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.² 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

¹Code Source: <https://github.com/codehacken/locality-preservation>

²*Low resource* has two interpretations i.e. one where corpus size to generate unsupervised pretrained embeddings is small and the other where the parallel corpus for alignment is minimal. We experiment with the later here.

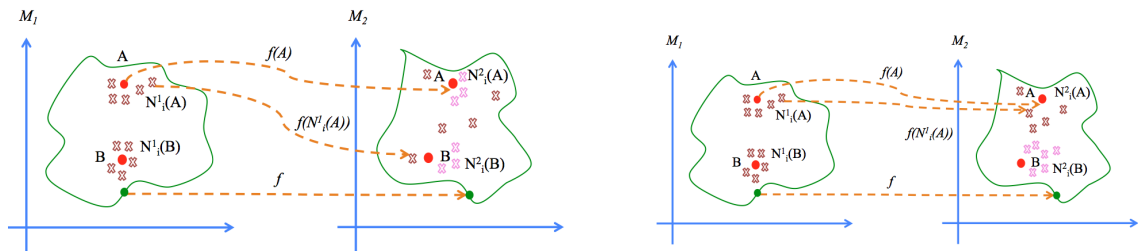


Figure 1: Quality of alignment with different types of losses. A, B are two words in two word embedding manifolds M_1 and M_2 . f is the manifold alignment function while $N_i^1(A)$ and $N_i^1(B)$ are their respective neighbors in manifold M_1 . $N_i^2(A)$ and $N_i^2(B)$ are their neighbors in manifold 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 its local structure.

(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 and regression tasks where there are two inputs (e.g., NLI and STS-B), 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. LPL achieves this by augmenting existing text \leftrightarrow label pairs with pseudo-pairs constructed from their neighbors.

Specifically, our main contributions are:

- We propose a loss function called **locality preserving loss (LPL)** to improve vector space alignment and show how the **alignment acts as a regularizer** while performing language inference and semantic text similarity. LPL improves correlation or accuracy of linear and non-linear mapping (deep networks) while exploiting the inherent geometries of existing pretrained embedding manifolds to optimize an alignment model (Table 1a, Figure 3).
- We show LPL is flexible and can be optimized with SGD. Hence, it can be applied to both deep neural networks and linear transformations. In contrast, previous cross-lingual word alignment models that are a linear map between source and target language, are learned using singular value decomposition.
- We show an increase in correlation on semantic text similarity (STS-B) and accuracy on SNLI in comparison with the baseline when the models are trained with small datasets.

Training with LPL shows up to 16.38% (90.1% relative), on SNLI up to 8.9% (19.3% relative) improvement when trained with just 1000 samples (0.002% of the dataset). We train a crosslingual word alignment model giving up to 4.1% (13.8% relative) improvement in comparison to a MSE optimized mapping while reducing the size of the parallel corpus required to train the mapping by 40% (3K out of 5K pairs).

2 Background & Related Work

2.1 Dimensionality Reduction

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 similar 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 tool 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) (Roweis and Saul, 2000), the datapoints are assumed to have a linear relation with their neighbors. To project each point, first a reconstruction loss is utilized to

learn the linear relation between a point and their k neighbors. Then, the linear relation is used to learn the embeddings in the reduced dimension.³

Wang et al. (2014) extend autoencoders by modifying the reconstruction loss to use nearest neighbors of data points, leveraging neighborhood relationships between datapoints from non-linear dimension reduction methods like LLE and Laplacian Eigenmaps.

2.2 Manifold Alignment

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, Figure 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.

Our work is similar to Bollegala et al. (2017) where the meta-embedding (a common embedding space) for different vector representations is generated using a locally linear embedding (LLE) which preserves the locality. One drawback, though, is that LLE does not learn a singular functional mapping between the source and target vector spaces. A linear mapping between a word and its neighbor is learned for each new word. Hence, the meta-embedding must be retrained 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. Simi-

³See Appendix C for an in-depth explanation of LLE.

lar to NORMA, LPL uses a modified locally linear representation of each embedding but, unlike it, LPL uses actual nearest neighbors in order to learn an embedding. This is important as NNs may not be present in annotated parallel corpus. **Hence, using NNs of annotated pairs in the corpus in LPL expands the size of the training dataset.** LPL is optimized with gradient descent and can be easily added to a deep neural network (as seen in §3.5).

3 Incorporating Locality Preservation into Task-based Learning

In this section, we describe locality preserving loss, the assumption underlying the loss function and objective functions (eq. (2), eq. (3)) that are optimized. The cumulative loss function while training the model is defined in eq. (4).

3.1 Locality Preservation Criteria

The locality preserving loss (LPL, eq. (2)) 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).

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

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 $V^p = \{w_1^p \dots w_c^p\}$ be the set of the unit pairs; we consider V^p a supervised training set (though it could be weakly supervised, e.g., 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 ($c \ll m$ and $c \ll n$). The mapping (manifold alignment) function is f .

In this paper, we experiment with three types of tasks: cross-lingual word alignment (mapping), natural language inference (classification), and semantic text similarity (regression). In cross-lingual word alignment, V^s and V^t represent the source and target vocabularies, V^p is a bilingual dictionary, and M^t and M^s are the target and source manifolds. f with θ_f parameters is a linear projection with a single weight matrix W . For NLI, V^t and V^s are target and source sentences with M^t and M^s being their manifolds. f is a 2-layer FFN.

3.3 Locality Preserving Loss (LPL)

We use a mapping function $f : M^s \rightarrow M^t$ to align 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 2-layer neural network (non-linear mapping) for NLI. The mapping is optimized using three loss functions: an orthogonal transform (Xing et al., 2015) $\mathcal{L}_{\text{ortho}}$ (constrain $W^{-1} = W^T$); mean squared error \mathcal{L}_{mse} (eq. 1); and locality preserving loss (LPL) \mathcal{L}_{lpl} (eq. 2).

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

$$\mathcal{L}_{\text{mse}} = \sum_{i \in V^p} \mathcal{L}_{\text{mse}}^i = \sum_{i \in V^p} \overbrace{\|f(m_i^s) - m_i^t\|_2^2}^{\mathcal{L}_{\text{mse}}^i}, \quad (1)$$

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. 2) 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 $f(m_i^s)$ be a weighted average of all the projected vectors of its neighbors $f(m_j^s)$. Formally, for a particular common item i , LPL at i minimizes

$$\mathcal{L}_{\text{lpl}}^i = \left\| m_i^t - \sum_{m_j^s \in N_k(m_i^s)} W_{ij} f(m_j^s) \right\|^2 \quad (2)$$

with $\mathcal{L}_{\text{lpl}} = \sum_{m_i^s, m_i^t \in V^p} \mathcal{L}_{\text{lpl}}^i$. 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

$$\mathcal{L}_{\text{lle}}^i = \left\| m_i^s - \sum_{m_j^s \in N_k(m_i^s)} W_{ij} m_j^s \right\|^2 \quad (3)$$

with $\mathcal{L}_{\text{lle}} = \sum_{m_i^s \in V^p} \mathcal{L}_{\text{lle}}^i$. 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 $\mathcal{L}_{\text{ortho}} = WW^T - I$. 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 .

3.4 Model Training with LPL

The total supervised loss becomes:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{mse}}(\theta_f) + \beta * \mathcal{L}_{\text{lpl}}(\theta_f, W) + \mathcal{L}_{\text{lle}}(W) + \mathcal{L}_{\text{ortho}}(\theta_f) \quad (4)$$

We introduce a constant β to allow control over the contribution of LPL to the total loss. Although we minimize total loss (4), shown explicitly with variable dependence, the optimization can be unstable

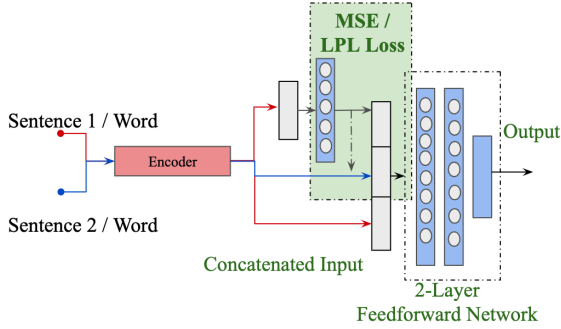


Figure 2: Use of alignment loss for the NLI and STS-B task. The pipeline consists of a 2-layer FFN used to classify sentence pairs. The output layer is of size 3 to classify the input into *entailment*, *contradiction* and *neutral*. It has a size of 1 to generate a continuous value between 0 and 1 for STS-B. The premise / sentence 1 and hypothesis / sentence 2 subspaces are aligned using a MSE and LPL loss that are then added as a concatenated input to train the classifier / regressor. A δ hyperparameter configured for each label provides the ability to perform *alignment* for *entailment* and *contradiction* while performing *divergence* for *neutral* input pairs.

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 \mathcal{L}_{lle} alone and the weights are frozen. Then, \mathcal{L}_{mse} and \mathcal{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 be 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. **Euclidean distance** between embeddings is used to find NNs.

3.5 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. Figure 2 shows an example architecture where alignment is used as a regularizer for the NLI task. The architecture contains a two layer FFN used to perform language inference, i.e., to predict if the given sentence pairs are *entailed*, *contradictory* or *neutral*. The in-

put 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:

$$\mathcal{L}_{total} = \gamma \sum_i \delta_{y_i} (\mathcal{L}_{mse}^i + \mathcal{L}_{lpl}^i + \mathcal{L}_{lle}^i(W)) + CE_{y_i} \quad (5)$$

where γ is a hyperparameter that controls the impact of the loss (learning rate). Thus, the loss (eq. (5)) is an extension of eq. (4) for a classification task but without \mathcal{L}_{ortho} , which is not applied as f is a 2-layer FFN (non-linear mapping) and the $WW^T = I$ 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 were set when optimizing network hyper-parameters. As the optimizer minimizes the loss, a divergence loss tends to $-\infty$; in practice, we clip the negative loss value at -1 .

4 Experiment Results & Analysis

We demonstrate the effectiveness of the locality preserving alignment on three types of tasks: semantic text similarity (regression), natural language inference (text classification) and crosslingual word alignment (mapping / regression). In order to compute local neighborhoods, as needed for, e.g., (2), we build a standard KD-Tree and find the nearest neighbors using Euclidean distance.

4.1 Semantic Text Similarity (STS)

In semantic text similarity (STS), we use the STS-B dataset (Cer et al., 2017) that is widely utilized as a part of the GLUE benchmark (Wang et al., 2018). The baseline model is the same as Siamese-BERT

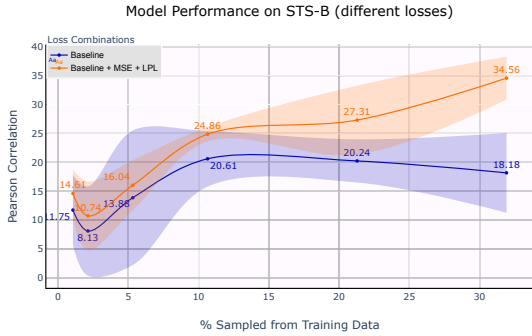


Figure 3: Pearson correlation on semantic text similarity with STS-B dataset. Locality Preserving Loss (LPL) improves sentence pair similarity in all data sizes from small (100 samples or 2% of total data) to 50% of the dataset.

(Reimers and Gurevych, 2019) where sentence embeddings are extracted individually using a BERT model (Devlin et al., 2018). An attached connected feedforward network (FFN) is trained to predict a sentence similarity score between 0 and 1 (optimized with squared error loss).⁴

To analyze the impact of various loss functions, the BERT model parameters are frozen so that sentence embeddings remain the same and only the parameters of the FFN are optimized. As described in §3.4, the baseline model is additionally trained with a MSE loss that aligns one sentence manifold to another ($m_1 \rightarrow m_2$) and the third model also trains with LPL. δ is set to mimic the normalized label thus generating the largest loss while aligning two sentences that are same. The contrastive loss separates the two sentence embeddings when they are dissimilar. The maximum margin is set to 0.1.

In order to study the impact on model performance when the training data is small, we limit the sampled training data size to upto 50% of the original (total dataset size is 5k).⁵ Figure 3 shows the Pearson correlation between the baseline model and the same regularizer with MSE and LPL (eq. 4). As observed, the correlation of models trained with LPL is higher for every training dataset size. The relative increase in correlation is in the range of 10.83 to 90.1%. We note here that the correlation cannot be compared with the original BERT model as we do not fine-tune the entire network but only the FFN in order to measure the improvements with the addition of LPL.

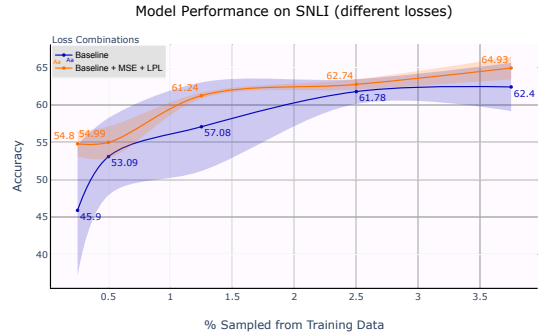


Figure 4: Accuracy of alignment regularization on SNLI. The graph shows the accuracy, averaged across 3 runs, for differing size of training samples up to 5% of the training dataset only (total: 500K).

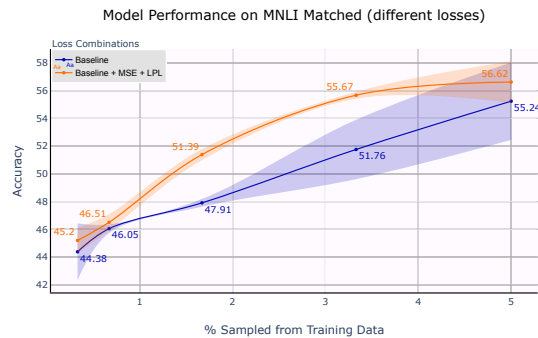


Figure 5: Accuracy of alignment regularization on MNLI dataset with a varying number of *matched* in-genre samples, up to 5% of the training dataset only (total: 300K samples).

4.2 Natural Language Inference

To test the effectiveness of alignment as a regularizer, a 2-layer FFN 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 with the impact of the loss function on two datasets: the Stanford natural language inference (SNLI) (Bowman et al., 2015a) 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 4, 5, and 6 show the accuracy of the models when optimized with a standard cross-entropy loss (baseline) and with MSE and LPL combined. The accuracy is measured when the size of the

⁴Appendix A.1 provides details about the dataset and FFN.

| 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 |
| LPL+MSE: CSLS | S0, S2 | SGD | 43.33 | 46.07 | 33.50 | 35.13 |

(a) We compare our method (bottom row: LPL) 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. The *method* column 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 Table 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.

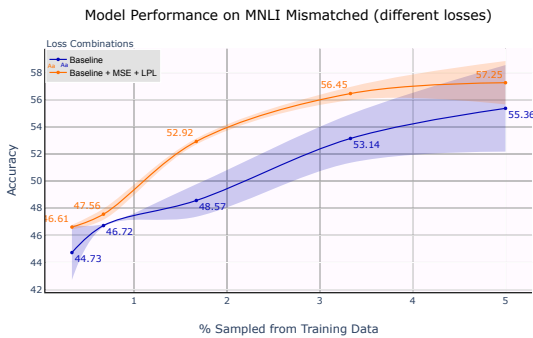


Figure 6: Accuracy of alignment regularization on MNLI dataset with a varying number of *mismatched* out-of-genre samples, up to 5% of the training dataset only (total: 300K samples).

training set is reduced.⁵ 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. The difference in accuracy (in Figure 4) is larger initially, it 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

⁵ Model accuracy using MSE and MSE + LPL with 100% of the training data for STS-B is provided in Appendix A.1 and A.2 for SNLI / MNLI.

be part of the training pairs themselves.

4.3 Crosslingual Word Alignment

The cross lingual word alignment dataset is from Dinu et al. (2014). The dataset is extracted from the Europarl corpus and consists of word pairs split into training (5k pairs) and test (1.5k pairs) respectively.⁶ From the 5K word pairs available for training only 3K pairs are used to train the model with LPL 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.⁴

Compared to previous methods that look at explicit mapping of points between the two spaces, LPL tries to maintain the relations between words and their neighbors in the source domain while projecting them into the target domain. Along with the mapping methods in Table 1a, previous methods also apply additional pre/post processing transforms 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.

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 preprocess-

⁶<http://opus.lingfil.uu.se/>

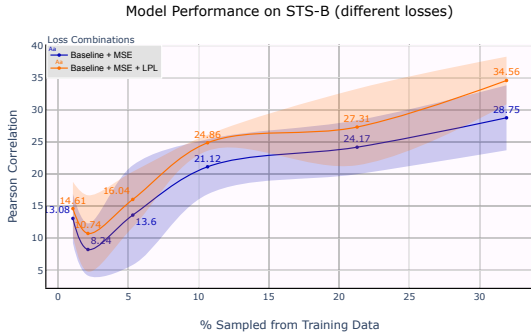


Figure 7: Pearson correlation of alignment regularization on STS-B up to 50% of the training dataset only (total: 4700 samples). The models are trained with MSE and MSE + LPL.

ing 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) under performs in comparison to the original baseline while giving improvements with LPL. 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 to use a two-step training process.

4.4 Ablation Study

Although we introduce LPL in this paper, models in §4.1 and §4.2 are both trained with a combination of MSE and LPL. This raises the question: how much does LPL contribute to the overall performance of the model? We analyze this question by training a model separately on the STS-B dataset with MSE only and then comparing it with the prior model trained with the combined losses. In Figure 7, we see that the model trained with MSE + LPL performs better with a maximum of 20.2% relative improvement (sampled dataset size is 30%) over one trained with MSE alone. Additionally when the dataset size is small (less than 10% of the training data), it is observed that variation in accuracy is smaller for the model with the combined loss.

Additionally, we show the results of ablation studies on SNLI, MNLI (matched) and MNLI (mismatched). As seen in figures 8, 9, 10, LPL’s contribution to the model’s accuracy in these tasks is lower in comparison to its contribution STS-B, but the variation in accuracy is smaller with it. Thus, we can conclude that LPL makes the model’s performance consistent across experiments.

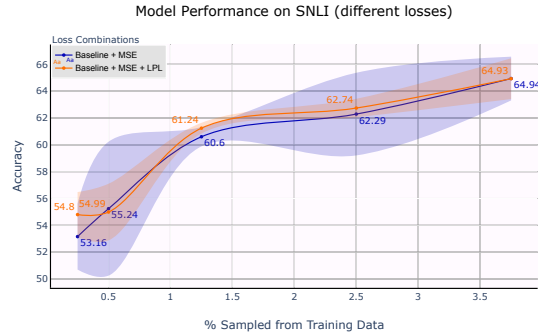


Figure 8: Accuracy of alignment regularization on SNLI up to 5% of the training dataset only. The models are trained with MSE and MSE + LPL.

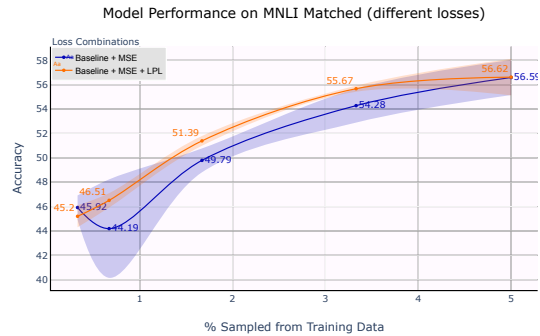


Figure 9: Accuracy of alignment regularization on STS-B up to 5% of the training dataset only. The models are trained with MSE and MSE + LPL.

4.5 Discussion

Table 2 shows the 2 nearest neighbors for a premise-hypothesis pair (P, H) taken from each classifier, i.e., baseline (B), baseline + MSE (BM), and baseline + MSE + LPL (BML) 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 BM and BML (NNs 3 and 4 for BM, NNs 5 and 6 for BML). The predicted labels of the NNs are clustered into entailment only. The sentence pair representations cluster 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

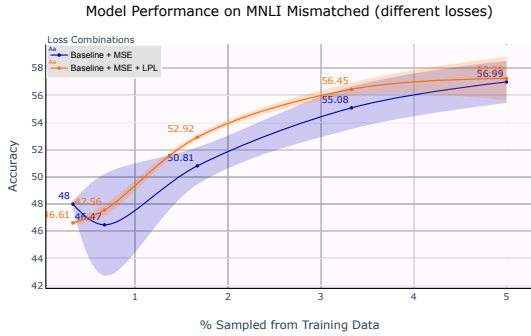


Figure 10: Accuracy of alignment regularization on STS-B up to 5% of the training dataset only. The models are trained with MSE and MSE + LPL.

| Model Type | ID | Sentence |
|------------|----|---|
| B | 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. |
| BM | 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. |
| BML | 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 2: 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 (B), baseline + MSE (BM), and baseline + MSE + LPL (BML). 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 initial alignment (or divergence) between the premise and hypothesis. Also, we observe that a model regularized with MSE and LPL are more likely to reach optimal parameters consistently.⁷

5 Conclusion

In this paper, we introduce a new 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. The models are trained with SGD as compared to others that learn with SVD. Additionally, the same alignment loss is applied as a regularizer in a classification task (NLI) and a regression task

⁷Check appendix section B for more details.

(STS-B) to demonstrate how it can improve the accuracy of the model over the baseline.

Broader Impact

In this section, we discuss the potential benefits and risks of using a locality preserving loss in the NLP tasks described in this paper.

Benefits of LPL. The main motivation of our work is to train models with limited data and showcase the effectiveness of locality preserving loss. When the dataset is small, models overfit the training data unable to generalize and exacerbate language biases. The main benefit of using LPL is that it maintains relationships between a datapoint and its neighbor in the target embedding space, restricting the model from overfitting.

Risks of Utilizing LPL. LPL’s ability to maintain relationships between points that are present in the source manifold after projection, can also be a risk. The weights W in equation 3 are learned by constructing a linear map between a datapoint and its neighbors using embeddings extracted from a pretrained model. Thus, any biased relationships prevalent in the pretrained model will become of a component of LPL and ultimately a part of the downstream fine-tuned model too. Hence it is necessary to evaluate the pretrained model thoroughly prior its use with LPL.

Acknowledgements

We would like to thank our anonymous reviewers from the Adapt-NLP Workshop and previous NLP conferences for their constructive reviews. We thank Prof. Konstantinos Kalpakis for his insights about manifold alignment and locality preservation methods. The hardware used in our computational studies is part of the UMBC HPCF facility and UMBC’s CARTA lab. This material is also based on research that is in part supported by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Air Force Research Laboratory (AFRL), DARPA, or the U.S. Government.

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. 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. 2015a. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015b. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.
- 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 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.
- 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*.
- Xiaofei He and Partha Niyogi. 2004. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- 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.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.
- Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2017. A survey of cross-lingual embedding models. *CoRR*, abs/1706.04902.
- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.
- David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, DTIC Document.
- 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.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- 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.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- 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.

Appendix

A Experimentation Details

This section discusses in detail the various experiments conducted in this paper. For each set of experiments, we provide an overview of the dataset used, a description of the dataset (with samples), information about the training set and list of any hyper-parameters that are optimized. The computing infrastructure used is a single Tesla P100-SXM2 GPU to train a single model. The GPU consumption is driven by the base text / word encoding model. GPU usage while training the feedforward network (FFN) is limited. In practice, once the embeddings are extracted from an language encoder (like BERT), multiple FFs can be simultaneously trained on a single GPU. Additional GPUs are used to scale experiments while training models with different loss function combinations and dataset sizes.

A.1 Semantic Text Similarity (STS)

In order to understand the impact of Locality Preserving Loss (LPL) while finetuning a model to measure how similar two sentences are, we use the STS-B dataset (Cer et al., 2017) that is widely utilized as a part of the GLUE benchmark (Wang et al., 2018). STS-B consists of a total of 8628 pairs of sentences of which 5749 are for training, 1500 pairs are part of the dev set (for hyper-parameter tuning) and 1379 are test pairs. The labels are a continuous value between 0 – 5 where 0 represents sentences that are not similar while 5 represents sentences that have the same syntax and meaning. While training various models, the labels are normalized between 0 and 1. Table 3 shows a few examples of sentence pairs in the dataset.

| Sentence 1 | Sentence 2 | Label |
|--|--|-------|
| A plane is taking off. | An air plane is taking off. | 5.0 |
| A man is spreading shredded cheese on a pizza. | A man is spreading shredded cheese on an uncooked pizza. | 3.8 |
| Three men are playing chess. | Two men are playing chess. | 2.6 |

Table 3: Sample sentence pairs from STS-B (Cer et al., 2017) dataset with their corresponding labels. The labels represent subjective human judgements of how similar the sentences are. It is a continuous variable.

A BERT model (Devlin et al., 2018) with an additional 2-layer feedforward network (FFN) (P_n) predicts the similarity score between sentences and

is optimized with MSE (between predicted score and label). The FFN consists of 2 hidden layers, each of size 1024. The baseline model (trained only with MSE) has a concatenated input (as shown in Figure 2 - *baseline*). While using an additional alignment loss (MSE or LPL), an additional single hidden layer FFN (A_n) is attached that aligns the two sentence manifolds. The aligned projection of sentence 1 is then concatenated as input to P_n (as shown in Figure 2 - *baseline with alignment*). Although this increases the number of parameters in the P_n (by 768×1024) for models trained with alignment, we experimented with increasing the baseline model with the same number of hidden layer parameters and found the baseline’s performance to decrease or remain constant. Hence, the size of the layers is maintained at 1024.

A.1.1 Crossvalidation

Figure 3 shows the pearson correlation between the predicted and actual similarity score. For each sample size on the X-axis, 3-fold cross validation is performed. Each time cross-validation is performed, different training pairs are randomly sampled (without replacement) from the complete dataset and the seed for initializing weights of each layer in the FFN is changed. To maintain consistency across experiments, the seed is maintained constant across trained models.

A.1.2 Hyperparameters

As described in §3.5, the hyperparameters include γ and δ . γ is a learning rate that defines how much of the alignment loss functions contribute to the overall loss. In practice, γ is set to 1.0.

Equation 5 uses LPL as a regularizer. We implement this as a contrastive divergence loss.

$$\mathcal{L}_{\text{mcd}}^i = \gamma \sum_i \max(-1.0, \delta_{y_i} * \mathcal{L}_{\text{mse}}^i) \quad (6)$$

$$\mathcal{L}_{\text{lcd}}^i = \gamma \sum_i \max(-1.0, \delta_{y_i} * \mathcal{L}_{\text{lpl}}^i) \quad (7)$$

The above equations change \mathcal{L}_{mse} and \mathcal{L}_{lpl} to a contrastive divergence (max margin) loss. We set the maximum margin to 0.1. The margin is manually tuned after optimizing it on validation pairs. The overall learning rate to train the model is 0.0001. The optimizer is RMSProp.

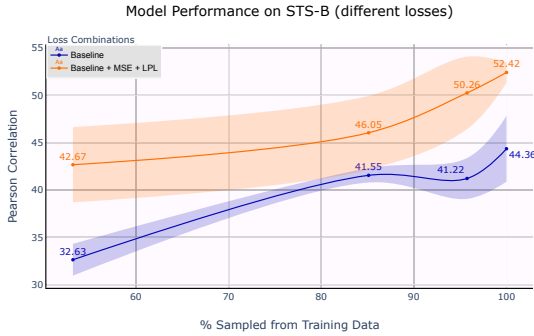


Figure 11: Pearson correlation on semantic text similarity with STS-B dataset for training sample size greater than 50% of the original dataset.

δ is the label specific hyper-parameter that defines how much loss can be contributed by a specific label. As the STS label is a continuous variable, δ is equal to the label value. This forces sentence pairs with scores that are 5.0 to have maximum loss while pairs with scores that tend towards 0 create a negative squared loss (this moves the embeddings apart) limited up-to the maximum margin.

A.1.3 Performance on Larger Dataset

In Figure 11, we see that the LPL + MSE loss model performs better than the baseline when dataset size is increased. The lowest bound in performance is higher than the upper bound of the baseline. LPLs improvement over the baseline when the entire dataset is used shows that as a whole STS-B may benefit from LPL irrespective of the dataset size. This is because LPL explicitly models a relationship between sentences present in a given sample text’s neighborhood, ensuring that those relationships are maintained while computing the similarity between a given sentence pair.

A.2 Natural Language Inference (NLI)

§4.2 discusses in detail the experiments with natural language inference. Experiments are conducted on SNLI (Bowman et al., 2015b) and MNLI (Williams et al., 2017) datasets. Stanford’s natural language inference dataset contains sentence pairs consisting of a premise and hypothesis. The model predicts if the hypothesis *entails* or *contradicts* the premise or if they have a neutral relationship (i.e., the two sentences are not related). The dataset contains 500K training pairs, 10K pairs in the dev set and 10K pairs of sentences for testing. Table 4 shows a few sentence pairs from the dataset.

MNLI (Williams et al., 2017) is an extension of the SNLI dataset where the sentences are from mul-

| Premise | Hypothesis | Label |
|--|---|----------------------|
| A person on a horse jumps over a broken down airplane. | A person is at a diner, ordering an omelette. | <i>Contradiction</i> |
| A person on a horse jumps over a broken down airplane. | A person is outdoors, on a horse. | <i>Entailment</i> |
| Children smiling and waving at camera | They are smiling at their parents | <i>Neutral</i> |

Table 4: Samples from the SNLI (Bowman et al., 2015b) dataset. Each pair consists of two sentences and a label with one of three values *entailment*, *neutral*, *contradiction*.

iple genres such as letters promoting non-profit organizations, government reports and documents as well as fictional books. This expands the variation in language used in sentences, reducing the model’s ability to memorize sentence pairs and their labels. Table 5 contains example pairs from the MNLI training data.

| Premise | Hypothesis | Label |
|---|--|-------------------|
| Conceptually cream skimming has two basic dimensions - product and geography. | Product and geography are what make cream skimming work. | <i>Neutral</i> |
| How do you know? All this is their information again. | This information belongs to them. | <i>Entailment</i> |

Table 5: Samples from the SNLI (Bowman et al., 2015b) dataset. Each pair consists of two sentences and label with one of three values *entailment*, *neutral*, *contradiction*

The model’s architecture is described in Figure 2. The sentence embeddings are extracted from BERT (Devlin et al., 2018) and the NLI classification layers are trained separately. For experiments in §4.2, the trained model is a 2-layer FNN, each hidden layer with a size of 4096. The activation function is a Leaky RELU. Similar to the experiments in appendix A.1.1, we perform 3-fold crossvalidation where the training dataset is resampled and the results presented in §4.2 are an average over these 3 runs.

A.2.1 Hyperparameters

The models are trained with a learning rate of 0.0001 and the optimizer is RMSProp. In order to train the model with LPL and MSE, δ is configured. In comparison to a continuous δ variable used in the STS task, in NLI, the δ is constant for each label. For each dataset, the δ parameter is set after manual testing on the dev set. Sentence pairs that have a *neutral* label tend to be sentences

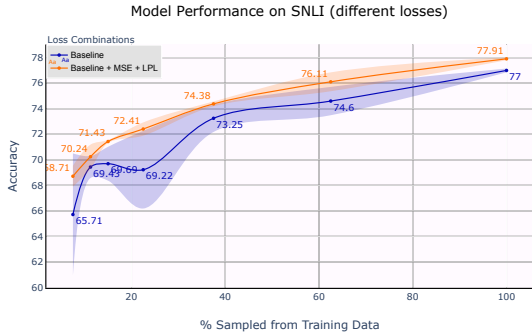


Figure 12: Accuracy of alignment regularization on SNLI. The graph shows the accuracy, averaged across 3 runs, for differing size of training samples from 5% to 100% of the training dataset only (total: 500K).

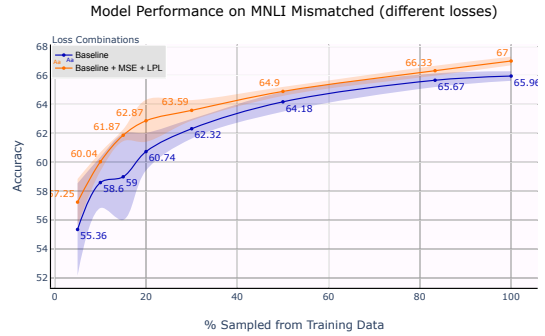


Figure 14: Accuracy of alignment regularization on MNLI dataset with a varying number of *mismatched* out-of-genre samples, from 5% to 100% of the training dataset only (total: 300K samples).

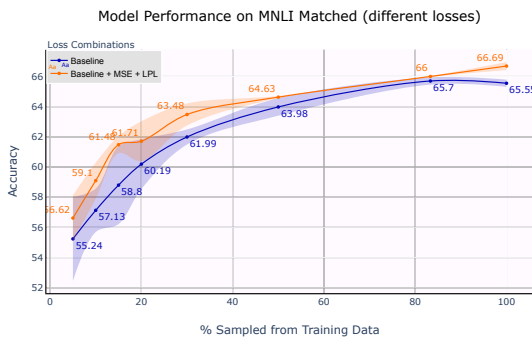


Figure 13: Accuracy of alignment regularization on MNLI dataset with a varying number of *matched* in-genre samples, from 5% to 100% of the training dataset only (total: 300K samples).

that are dissimilar. The semantic difference can be based on differing subjects, predicates or objects in the sentence, the content / topic or even genre of the sentence. Hence while training the model, these projections are **separated** with the alignment loss (and have a negative δ) rather than converged.

From our initial experiments, we found that lower δ on *entailment* and *contradiction* yielded no change in accuracy from baseline. The accuracy increases when a higher positive δ scalar multiple is attached to *entailment* & *contradiction*, and a negative scalar is multiplied to the loss generated for *neutral* label samples.

| Dataset | Entailment | Neutral | Contradiction |
|---------|------------|---------|---------------|
| SNLI | 100.0 | -5.0 | 0.0 |
| MNLI | 250.0 | 1.0 | -10.0 |

Table 6: Samples from the SNLI (Bowman et al., 2015b) dataset. Each pair consists of two sentences and label with one of three values *entailment*, *neutral*, *contradiction*

A.2.2 Performance on Larger Dataset

Apart from better accuracy when the training dataset is small, in Figures 12, 13, and 14, we observe the accuracy of models trained with the alignment loss using MSE alone and another in combination with LPL converge as the number of training samples increases. 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. Because 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). 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 unrelated text corpus reconstruct local manifolds.

A.3 Crosslingual Word Alignment

For experiments with cross lingual word alignment, we use the parallel corpus from Dinu et al. (2014). Table 7 compares our method with other methods such as Xing et al. (2015) and Faruqi and Dyer

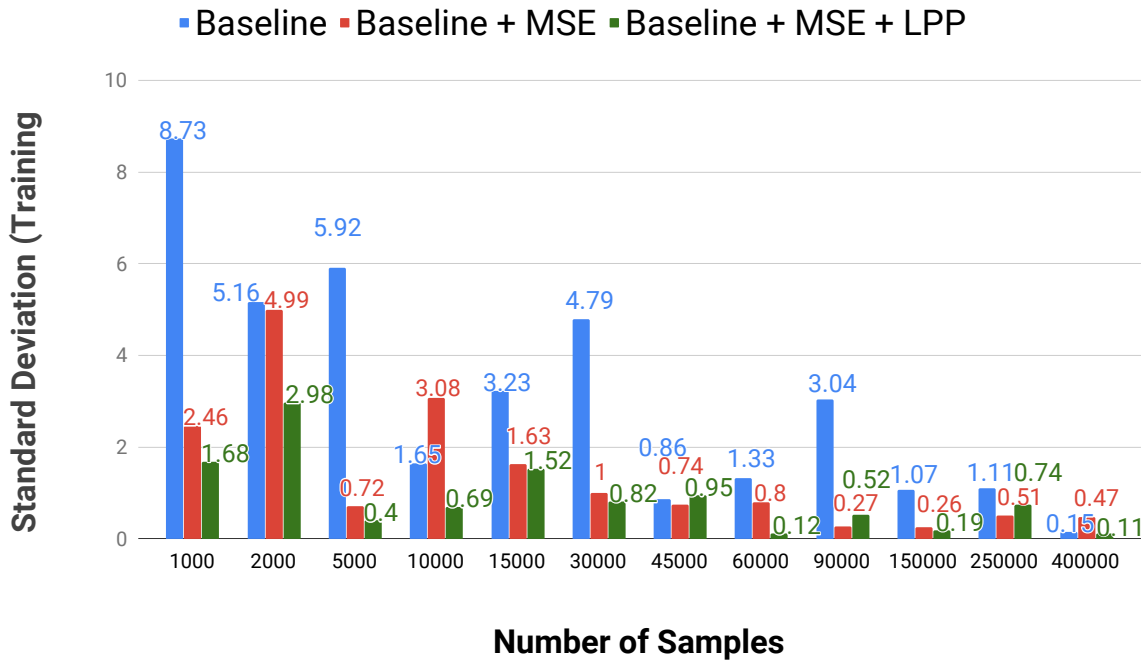


Figure 15: Standard deviation in performance. The graphs show the standard deviation in performance over 3 runs when the SNLI training dataset size varies. In each case, the model trained with LPL + MSE + task loss has the least variation in performance, while the model trained with task loss + MSE or the task loss alone having a higher variance in performance.

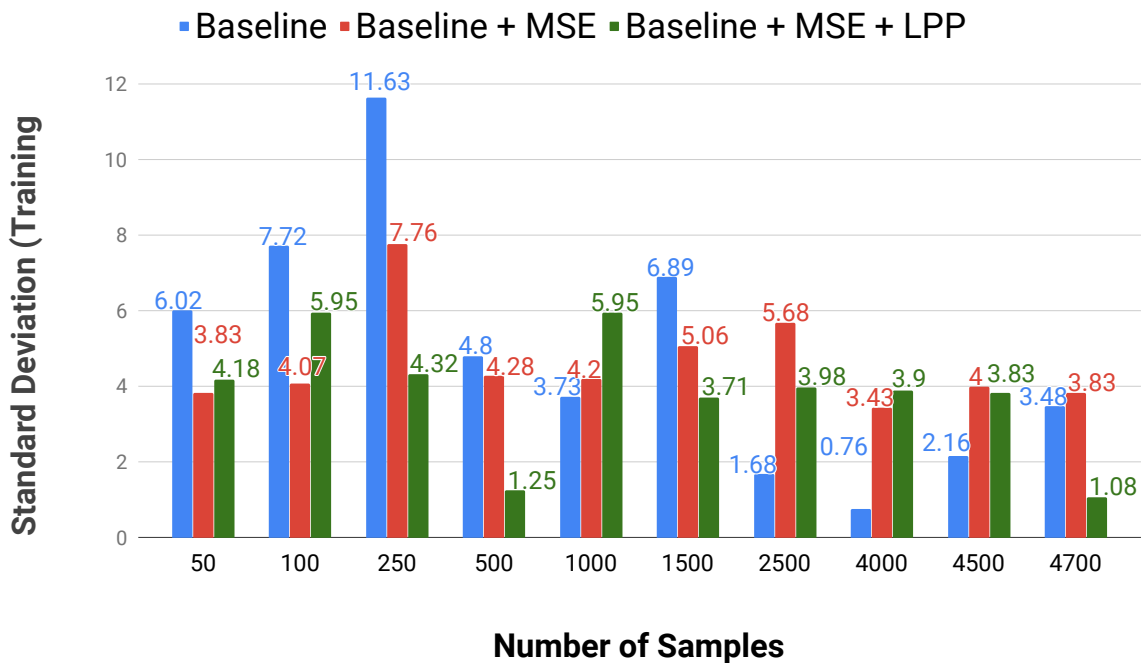


Figure 16: Standard deviation in pearson correlation. The graphs show the standard deviation in correlation over 3 runs when the STS-B training dataset size varies. In each case, the model trained with LPL + MSE + task loss has the least variation in performance, while the model trained with task loss + MSE or the task loss alone having a higher variance in performance.

(2014) and extends results in Table 1a. The dataset contains word pairs collected from the Europarl corpus. The bilingual dictionary has 5000 word pairs for training and 1500 word pairs for testing and evaluation. The language pairs in the dictionary include English \leftrightarrow Italian, English \leftrightarrow German, English \leftrightarrow Finnish and English \leftrightarrow Spanish. Figure 18 shows the pipeline to perform cross-lingual word alignment.

| Method | EN - IT |
|-------------------------------------|--------------|
| Mikolov et al. (2013) | 34.93 |
| Xing et al. (2015) | 36.87 |
| Faruqui and Dyer (2014) | 37.80 |
| Artetxe et al. (2016) | 39.27 |
| Locality Preserving Loss (Our Work) | 43.33 |

Table 7: Accuracy of various models predicting word translated from English to Italian.

Once the initial pretrained word vectors are selected, preprocessing can be applied. Preprocessing functions include unit normalization, whitening and z-normalization (Table 1b). Ruder et al. (2019) observe that deep neural networks are perform word alignment poorly in comparison to a linear transformation ($Y = WX$). The linear transformation matrix W is learned by minimizing the sum of squared loss between Y and WX . Optimal parameters for W are learned by minimizing the loss with singular value decomposition (SVD) (Artetxe et al., 2016).

Locality Preserving Alignment combines preprocessing functions and training W . Also, parameters of W are learned using SGD, showcasing that LPL can be added to both linear and non-linear transformations. To find the translated word in the target embedding space, multiple inference mechanisms are available such as nearest neighbor (NN), inverted softmax (Smith et al., 2017), and cross-domain similarity scaling (CSLS) (Conneau et al., 2017). We use CSLS to find the translated word in the target language.

In our experiments, we use original pretrained word embeddings obtained from the dataset (Dinu et al., 2014). The models are trained with a learning rate of 0.001. The β parameter is the learning rate specific to LPL (from equation 4) and is set to 0.7 and is manually tuned against the validation dataset.

Table 8 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

| Manifold | Nearest Neighbors of “Windows” |
|----------------------|---|
| 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, ntf5, mac, 95/98/nt, nt, osx, msdos |

Table 8: 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.

between the two spaces, LPL 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.

B Measuring Performance Consistency

Figure 16 and 17 chart the variation in evaluated performance on each model on tasks STS-B and MNLI when the size of the training dataset varies. In each case, a model optimized with the task specific loss has higher variation in performance than models that are optimized with additional losses MSE and LPL. The variation for baseline models are higher when the size of the dataset is small. Figure 15 shows the variation in test accuracy across runs on the SNLI dataset. The variation in accuracy is highest for the baseline model when the size of the dataset is small. For example, when the baseline is trained with 1000 samples (0.002% of the dataset), the variation in accuracy is 8.73%. Similarly, when the baseline is trained with 50 samples (0.01% of the dataset) in STS-B task, the variation in accuracy is 6.02% (11.63% when sample size is 250). The variation in accuracy reduces as the sample size increases.

This occurs because when the subset of data randomly sampled, the quality of instances sampled has a large impact on the final performance of the model. Thus, measuring the consistency of the model’s performance over multiple runs is a vital evaluation criteria (as much as the accuracy itself).

Hence, training the model with an alignment loss, i.e., with locality preservation (\mathcal{L}_{mse} and \mathcal{L}_{lpl}), empirically guarantees that the model reaches near-optimal performance when the size of the supervised set is limited and that it has a narrow bound as compared to the baseline model trained without

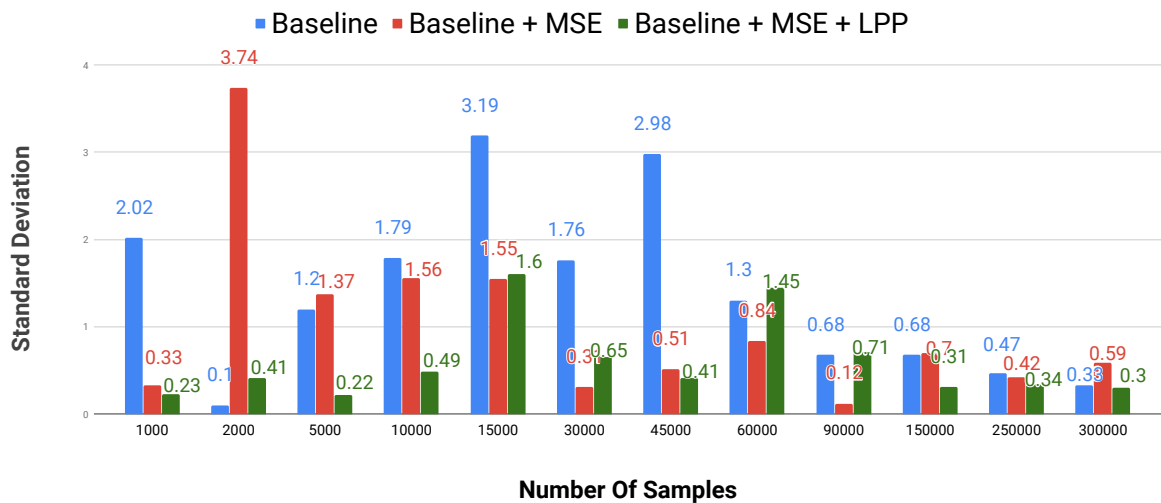
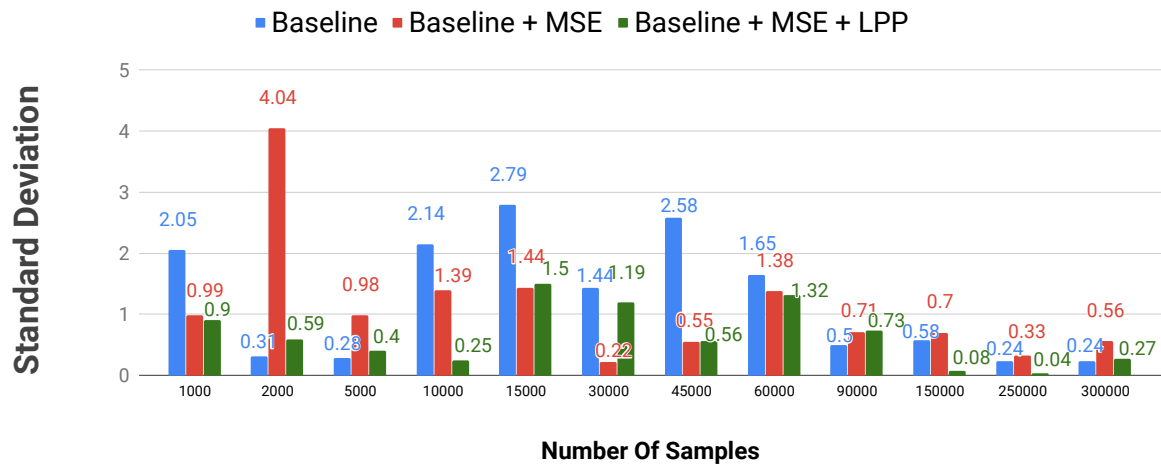


Figure 17: Standard deviation in accuracy of alignment on MNLI. The graph shows standard deviation in accuracy when size of the training sample set differs (total: 300K) for the baseline, baseline + MSE and baseline + MSE + LPL models: LPL yields more consistently optimal systems. (a) Standard deviation in accuracy when tested with in-genre sentence pairs (matched MNLI) (b) Standard deviation in accuracy when tested with out-of-genre sentence pairs (mismatched MNLI)

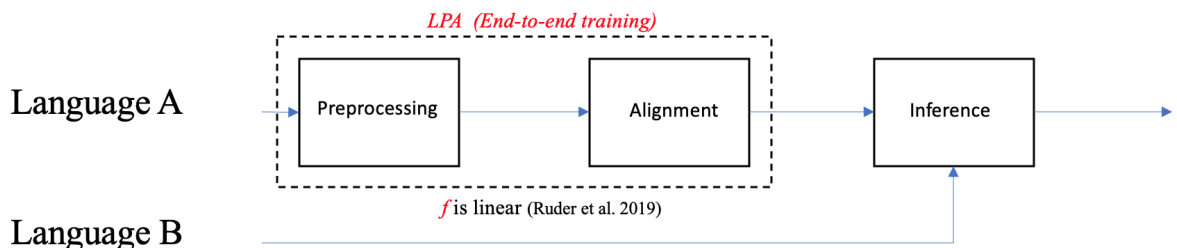


Figure 18: Crosslingual Word Alignment Pipeline (CLA). As described in Ruder et al. (2019), the CLA pipeline includes choosing an existing set of pretrained vectors that can be preprocessed. Table 1b contains a list of preprocessing functions that can be applied. Artetxe et al. (2018) evaluate each preprocessing method in detail. Alignment involves learning the transform matrix W and inference involves finding the translated word.

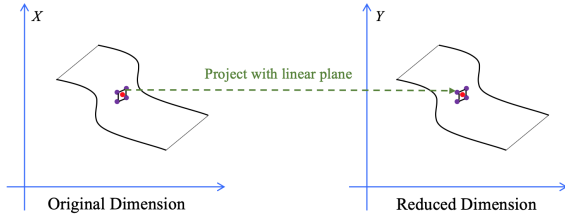


Figure 19: Locally Linear Embedding (LLE). The red point is a vector X_i that is reconstructed by a plane of its neighbors in blue. This linear plane is functionally represented by $W \in \mathbb{R}^{n \times k}$ for n points, each having k neighbors.

them. While performing these experiments, not only are the vocabularies randomly initialized, but also parameters too, making the model less dependent on how the training pairs are sampled from the dataset.

C Locally Linear Embedding

Our work is inspired from locally linear embedding (LLE) (Roweis and Saul, 2000). As discussed in §2.1, in LLE, the datapoints are assumed to have a linear relation with their neighbors (Figure 19). The reduced dimension projection of each vector is learned through a two step process: (a) Learn the linear relationship through a reconstruction loss (b) Use relation to learn low dimension representation. Assume each point in the manifold has k neighbors N_i . The reconstruction loss is:

$$\mathcal{L}_{\text{reconstruct}} = \sum_i \left\| X_i - \sum_{j \in N_i} W_{ij} X_j \right\|^2, \quad (8)$$

where X_i is the datapoint and the X_j represents each neighbor. An additional constraint is imposed on the weights ($\sum_{ij} W_{ij} = 1$) to make the transform scale invariant.

In (8) the weights W are an $N \times K$ matrix in a dataset of N points (i.e., each point has its own weights). Given a learned W from (8), we learn Y_i (a projection for X_i) by minimizing the following:

$$\sum_i \left\| Y_i - \sum_{j \in N_i} W_{ij} Y_j \right\|^2, \quad (9)$$

Y_i is typically with reduced dimensions.