Learning Geometric Word Meta-Embeddings

Pratik Jawanpuria[1], **N T V Satya Dev**[2], **Anoop Kunchukuttan**[1], **Bamdev Mishra**[1]
[1]Microsoft, India [2]Vayve Technologies, India
[1]{pratik.jawanpuria,ankunchu,bamdevm}@microsoft.com
[2]tvsatyadev@gmail.com

Abstract

We propose a geometric framework for learning meta-embeddings of words from different embedding sources. Our framework transforms the embeddings into a common latent space, where, for example, simple averaging or concatenation of different embeddings (of a given word) is more amenable. The proposed latent space arises from two particular geometric transformations - source embedding specific orthogonal rotations and a common Mahalanobis metric scaling. Empirical results on several word similarity and word analogy benchmarks illustrate the efficacy of the proposed framework.

1 Introduction

Word embeddings have become an integral part of modern NLP. They capture semantic and syntactic similarities and are typically used as features in training NLP models for diverse tasks like named entity tagging, sentiment analysis, and classification, to name a few. Word embeddings are learned in an unsupervised manner from large text corpora and a number of pre-trained embeddings are readily available. The quality of the word embeddings, however, depends on various factors like the size and genre of training corpora as well as the training method used. This has led to ensemble approaches for creating meta-embeddings from different original embeddings (Yin and Shutze, 2016; Coates and Bollegala, 2018; Bao and Bollegala, 2018; O'Neill and Bollegala, 2020). Meta-embeddings are appealing because they can improve quality of embeddings on account of noise cancellation and diversity of data sources and algorithms.

Various approaches have been proposed to learn meta-embeddings and can be broadly classified into two categories: (a) simple linear methods like averaging or concatenation, or a low-dimensional projection via singular value projection (Yin and Shutze, 2016; Coates and Bollegala, 2018) and (b) non-linear methods that aim to learn meta-embeddings as shared representation using auto-encoding or transformation between common representation and each embedding set (Muromägi et al., 2017; Bollegala et al., 2018; Bao and Bollegala, 2018; O'Neill and Bollegala, 2020).

In this work, we focus on simple linear methods such as averaging and concatenation for computing meta-embeddings, which are very easy to implement and have shown highly competitive performance (Yin and Shutze, 2016; Coates and Bollegala, 2018). Due to the nature of the underlying embedding generation algorithms (Mikolov et al., 2013; Pennington et al., 2014), correspondences between dimensions, e.g., of two embeddings $x \in \mathbb{R}^d$ and $z \in \mathbb{R}^d$ of the same word, are usually not known. Hence, averaging may be detrimental in cases where the dimensions are negatively correlated. Consider the scenario where $z := -x$. Here, simple averaging of x and z would result in the zero vector. Similarly, when z is a (dimension-wise) permutation of x, simple averaging would result in a sub-optimal meta-embedding vector compared to averaging of *aligned* embeddings. Therefore, we propose to align the embeddings (of a given word) as an important first step towards generating meta-embeddings.

To this end, we develop a geometric framework for learning meta-embeddings, by aligning different embeddings in a common latent space, where the dimensions of different embeddings (of a given word) are in coherence. Mathematically, we perform different orthogonal transformations of the source embeddings to learn a latent space along with a Mahalanobis metric that scales different features appropriately. The meta-embeddings are, subsequently, learned in the latent space, e.g., using averaging or concatenation. Empirical results on the word similarity and the word analogy tasks

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 39–44

show that the proposed geometrically aligned meta-embeddings outperform strong baselines such as the plain averaging and the plain concatenation models.

2 Proposed Geometric Modeling

Consider two (monolingual) embeddings $x_i \in \mathbb{R}^d$ and $z_i \in \mathbb{R}^d$ of a given word i in a d-dimensional space. As discussed earlier, embeddings generated from different algorithms (Turian et al., 2010; Mikolov et al., 2013; Pennington et al., 2014; Dhillon et al., 2015; Bojanowski et al., 2017) may express different characteristics (of the same word). Hence, the goal of learning a meta-embedding w_i (corresponding to word i) is to generate a representation that inherits the properties of the different source embeddings (e.g., x_i and z_i).

Our framework imposes orthogonal transformations on the given source embeddings to enable alignment. To allow a more effective model for comparing similarity between different embeddings of a given word, we additionally induce this latent space with the Mahalanobis metric. The Mahalanobis similarity generalizes the cosine similarity measure, which is commonly used for evaluating the relatedness between word embeddings. Unlike cosine similarity, the Mahalanobis metric does not assume uncorrelated feature and it incorporates the feature correlation information from the training data (Jawanpuria et al., 2019). The combination of orthogonal transformation and Mahalanobis metric learning allows to capture any *affine* relationship that may exist between word embeddings. Mathematically, this relates to the singular value decomposition of a matrix (Bonnabel and Sepulchre, 2009; Mishra et al., 2014).

Overall, we formulate the problem of learning geometric transformations – the orthogonal rotations and the metric scaling – via a binary classification problem (discussed later). The meta-embeddings are subsequently computed using these transformations. The following sections formalize the proposed latent space and meta-embedding models.

2.1 Learning the Latent Space

In this section, we learn the latent space using geometric transformations.

Let $\mathbf{U} \in \mathcal{M}^d$ and $\mathbf{V} \in \mathcal{M}^d$ be orthogonal transformations for embeddings x_i and z_i, respectively, for all words $i = 1, \ldots, n$. Here $\mathcal{M}^d$ represents the set of $d \times d$ orthogonal matrices. The aligned embeddings in the latent space corresponding to x_i and z_i can then be expressed as $\mathbf{U}x_i$ and $\mathbf{V}z_i$, respectively. We next induce the Mahalanobis metric $\mathbf{B}$ in this (aligned) latent space, where $\mathbf{B}$ is a $d \times d$ symmetric positive-definite matrix. In this latent space, the similarity between the two embeddings x_i and z_i can be obtained by the following expression of their dot product: $(\mathbf{U}x_i)^\top \mathbf{B}(\mathbf{V}z_i)$. This expression may also be interpreted as the standard dot product (cosine similarity) between $\mathbf{B}^{\frac{1}{2}}\mathbf{U}x_i$ and $\mathbf{B}^{\frac{1}{2}}\mathbf{V}z_i$, where $\mathbf{B}^{\frac{1}{2}}$ denotes the matrix square root of the symmetric positive definite matrix $\mathbf{B}$.

The orthogonal transformations as well as the Mahalanobis metric are learned via the following binary classification problem: pairs of word embeddings $\{x_i, z_i\}$ of the same word i belong to the positive class while pairs $\{x_i, z_j\}$ belong to the negative class (for $i \neq j$). We consider the similarity between the two embeddings in the latent space as the decision function of the proposed binary classification problem. Let $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$ and $\mathbf{Z} = [z_1, \ldots, z_n] \in \mathbb{R}^{d \times n}$ be the word embedding matrices for n words, where the columns correspond to different words. In addition, let $\mathbf{Y}$ denote the label matrix, where $\mathbf{Y}_{ii} = 1$ for $i = 1, \ldots, n$ and $\mathbf{Y}_{ij} = 0$ for $i \neq j$. The proposed optimization problem employs the simple to optimize square loss function:

$$\min_{\substack{\mathbf{U},\mathbf{V} \in \mathcal{M}^d, \\ \mathbf{B} \succ 0}} \left\| \mathbf{X}^\top \mathbf{U}^\top \mathbf{B} \mathbf{V} \mathbf{Z} - \mathbf{Y} \right\|^2 + C \left\| \mathbf{B} \right\|^2, \quad (1)$$

where $\| \cdot \|$ is the Frobenius norm (which generalizes the 2-norm to matrices) and $C > 0$ is the regularization parameter.

2.2 Averaging and Concatenation in Latent Space

Meta-embeddings constructed by averaging or concatenating the given word embeddings have been shown to obtain highly competitive performance (Yin and Shutze, 2016; Coates and Bollegala, 2018). Hence, we propose to learn meta-embeddings as averaging or concatenation in the learned latent space.

Geometry-Aware Averaging

The meta-embedding w_i of a word i is generated as an average of the (aligned) word embeddings in the latent space. The latent space representation of x_i, as a function of orthogonal transformation $\mathbf{U}$ and metric $\mathbf{B}$, is $\mathbf{B}^{\frac{1}{2}}\mathbf{U}x_i$

(Jawanpuria et al., 2019). Hence, we obtain $w_i = \texttt{average}(\mathbf{B}^{\frac{1}{2}}\mathbf{U}x_i, \mathbf{B}^{\frac{1}{2}}\mathbf{V}z_i) = (\mathbf{B}^{\frac{1}{2}}\mathbf{U}x_i + \mathbf{B}^{\frac{1}{2}}\mathbf{V}z_i)/2$.

It should be noted that the proposed geometry-aware averaging approach generalizes the *plain averaging* method proposed in (Coates and Bollegala, 2018), which is now a particular case in our framework by choosing $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{B}$ as identity matrices.

Geometry-Aware Concatenation

We next propose to concatenate the aligned embeddings in the learned latent space. For a given word i, with x_i and z_i as different source embeddings, the meta-embeddings w_i learned by the proposed geometry-aware concatenation model is $w_i = \texttt{concatenation}(\mathbf{B}^{\frac{1}{2}}\mathbf{U}x_i, \mathbf{B}^{\frac{1}{2}}\mathbf{V}z_i) = [(\mathbf{B}^{\frac{1}{2}}\mathbf{U}x_i)^\top, (\mathbf{B}^{\frac{1}{2}}\mathbf{V}z_i)^\top]^\top$. The plain concatenation method studied in (Yin and Shutze, 2016) is a special case of the proposed geometry-aware concatenation (by setting $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{B}$ as identity matrices).

2.3 Optimization

The proposed optimization problem (1) employs square loss function and ℓ_2-norm regularization, both of which are well-studied in the literature. The search space is the Cartesian product of the set of d-dimensional symmetric positive definite matrices and the set of d-dimensional orthogonal matrices, both of which are smooth spaces. Such sets have well-known Riemannian manifold structure (Lee, 2003) that allows to propose computationally efficient iterative optimization algorithms. A manifold may be viewed as a generalization of the notion of surface to higher dimensions. We employ the popular Riemannian optimization framework (Absil et al., 2008) to solve (1). Recently, Jawanpuria et al. (2019) have studied a similar optimization problem in the context of learning cross-lingual word embeddings.

Our implementation is done using the Pymanopt toolbox (Townsend et al., 2016), which is a publicly available Python toolbox for Riemannian optimization algorithms. In particular, we use the conjugate gradient algorithm of Pymanopt. For this, we just need to supply the objective function of (1). This can be done efficiently as the numerical cost of computing the objective function is $O(nd^2)$. The overall computational cost of our implementation scales linearly with the number of words in the vocabulary sets. Our code is available at `https://github.com/SatyadevNtv/geo-meta-emb`.

3 Experiments

In this section, we evaluate the performance of the proposed meta-embedding models.

3.1 Evaluation Tasks and Datasets

We consider the following standard evaluation tasks (Yin and Shutze, 2016; Coates and Bollegala, 2018):

- **Word similarity**: in this task, we compare the human-annotated similarity scores between pairs of words with the corresponding cosine similarity computed via the constructed meta-embeddings. We report results on the following benchmark datasets: **RG** (Rubenstein and Goodenough, 1965), **MC** (Miller and Charles, 1991), **WS** (Finkelstein et al., 2001), **MTurk** (Halawi et al., 2012), **RW** (Luong et al., 2013), and **SL** (Hill et al., 2015). Following previous works (Yin and Shutze, 2016; Coates and Bollegala, 2018; O'Neill and Bollegala, 2020), we report the Spearman correlation score (higher is better) between the cosine similarity (computed via meta-embeddings) and the human scores.
- **Word analogy**: in this task, the aim is to answer questions which have the form "*A is to B as C is to ?*" (Mikolov et al., 2013). After generating the meta-embeddings a, b, and c (corresponding to terms *A*, *B*, and *C*, respectively), the answer is chosen to be the term whose meta-embedding has the maximum cosine similarity with $(b-a+c)$ (Mikolov et al., 2013). The benchmark datasets include **MSR** (Gao et al., 2014), **GL** (Mikolov et al., 2013), and **SemEval** (Jurgens et al., 2012). Following previous works (Yin and Shutze, 2016; Coates and Bollegala, 2018; O'Neill and Bollegala, 2020), we report the percentage of correct answers for MSR and GL datasets, and the Spearman correlation score for SemEval. In both cases, a higher score implies better performance.

We learn the meta-embeddings from the following publicly available 300-dimensional pre-trained word embeddings for English.

- **CBOW** (Mikolov et al., 2013): has 929 023 word embeddings trained on Google News.
- **GloVe** (Pennington et al., 2014): has

Model	RG	MC	WS	MTurk	RW	SL	Avg.(WS)	MSR	GL	SemEvaL	Avg.(WA)
CBOW	76.1	80.0	77.2	68.4	53.4	44.2	66.5	71.7	55.4	**20.4**	49.2
GloVe	82.9	84.0	79.6	70.0	48.7	45.3	68.4	69.3	**75.2**	18.6	54.4
CONC	81.1	84.6	**81.4**	71.9	54.6	46.0	69.9	76.6	69.9	20.1	55.5
AVG	81.5	83.7	79.4	**72.1**	52.9	46.2	69.3	73.7	66.9	19.7	53.4
Geo-CONC	**86.0**	**85.0**	81.2	70.5	55.6	**48.2**	**71.1**	**78.1**	73.3	19.9	**57.1**
Geo-AVG	85.8	83.5	81.2	69.1	**55.7**	**48.2**	70.6	77.3	72.3	19.5	56.3

Table 1: Generalization performance of the meta-embedding algorithms on the word similarity and the word analogy tasks with GloVe and CBOW source embeddings. The columns 'Avg.(WS)' and 'Avg.(WA)' correspond to the average performance on the word similarity and the word analogy tasks, respectively.

Model	RG	MC	WS	MTurk	RW	SL	Avg.(WS)	MSR	GL	SemEvaL	Avg.(WA)
GloVe	82.9	**84.0**	79.6	70.0	48.7	45.3	68.4	69.3	75.2	18.6	54.4
fastText	**83.8**	82.5	**83.5**	73.3	**58.0**	46.4	71.2	78.7	71.0	22.5	57.4
CONC	**83.8**	82.5	83.4	73.3	57.9	46.4	71.2	79.8	71.7	22.5	58.0
AVG	83.4	82.1	**83.5**	73.3	**58.0**	46.5	71.1	79.7	71.7	22.4	57.9
Geo-CONC	83.7	**84.0**	82.6	**74.6**	55.1	**48.4**	**71.4**	**80.4**	**79.3**	21.5	**60.4**
Geo-AVG	83.6	82.0	82.7	74.3	57.0	**48.4**	71.3	79.1	71.1	**23.1**	57.8

Table 2: Generalization performance of the meta-embedding algorithms on the word similarity and the word analogy tasks with GloVe and fastText source embeddings. The columns 'Avg.(WS)' and 'Avg.(WA)' correspond to the average performance on the word similarity and the word analogy tasks, respectively.

1 917 494 word embeddings trained on 42B tokens of web data from the common crawl.

- **fastText** (Bojanowski et al., 2017): has 2 000 000 word embeddings trained on common crawl.

The meta-embeddings are learned on the common set of words from different pairs of the source embeddings. The number of common words between various source embeddings pairs are as follows: 154 077 (GloVe ∩ CBOW), 552 168 (GloVe ∩ fastText), and 641 885 (CBOW ∩ fastText).

3.2 Results and Discussion

The performance of our geometry-aware averaging and concatenation models, henceforth termed as Geo-AVG and Geo-CONC, respectively, are reported in Tables 1-3. Each table corresponds to a pair of source embeddings (from CBOW, GloVe, and fastText) and the meta-embeddings generated from the source embeddings. We report the performance of the following:

- the proposed models Geo-AVG and Geo-CONC
- the meta-embeddings models AVG (Coates and Bollegala, 2018) and CONC (Yin and Shutze, 2016), which perform plain averaging and concatenation, respectively
- the source embeddings, which serve as a benchmark the meta-embeddings algorithms should ideally surpass in order to justify their usage

We observe that the proposed geometry-aware models (Geo-AVG and Geo-CONC) outperform the individual source embeddings in most datasets. Among the source embeddings, fastText performs better than CBOW and GloVe. Interestingly, we observe that the performance of the meta-embeddings generated by the proposed Geo-CONC with CBOW and GloVe (results in Table 1) is at par with the fastText embeddings (results in Table 2).

The proposed models also easily surpass the AVG and CONC models in both the word similarity and the word analogy tasks. In all the three tables, the proposed models obtain the best overall performance in both the tasks. This shows that the alignment of word embedding spaces with orthogonal rotations and the Mahalanobis metric improves the overall quality of the meta-embeddings.

Model	RG	MC	WS	MTurk	RW	SL	Avg.(WS)	MSR	GL	SemEvaL	Avg.(WA)
CBOW	76.1	80.0	77.2	68.4	53.4	44.2	66.5	71.7	55.4	20.4	49.2
fastText	83.8	82.5	**83.5**	73.3	58.0	46.4	71.2	78.7	71.0	**22.5**	57.4
CONC	83.8	82.5	**83.5**	73.6	**59.9**	46.4	71.6	79.9	75.8	**22.5**	59.4
AVG	83.7	82.5	83.4	**73.7**	59.8	46.4	71.6	79.9	75.8	**22.5**	59.4
Geo-CONC	85.3	84.3	82.9	73.6	59.7	**47.4**	72.2	**80.1**	**76.9**	22.1	**59.7**
Geo-AVG	**85.5**	**84.6**	82.9	73.6	59.7	**47.4**	**72.3**	79.9	**76.9**	22.0	59.6

Table 3: Generalization performance of the meta-embedding algorithms on the word similarity and the word analogy tasks with CBOW and fastText source embeddings. The columns 'Avg.(WS)' and 'Avg.(WA)' correspond to the average performance on the word similarity and the word analogy tasks, respectively.

4 Conclusion

We propose a geometric framework for learning meta-embeddings of words from various sources of word embeddings. Our framework aligns the embeddings in a common latent space. The importance of learning the latent space is shown in several benchmark datasets, where the proposed algorithms (Geo-AVG and Geo-CONC) outperforms the plain averaging and the plain concatenation models.

Extending the proposed geometric framework to non-linear word meta-embedding approaches and for generating sentence meta-embeddings are promising directions of future research.

References

P.-A. Absil, R. Mahony, and R. Sepulchre. 2008. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ.

C. Bao and D. Bollegala. 2018. Learning word meta-embeddings by autoencoding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1650–1661.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146. https://fasttext.cc/docs/en/english-vectors.html.

D. Bollegala, K. Hayashi, and K. Kawarabayashi. 2018. Think globally, embed locally—locally linear meta-embedding of words. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

S. Bonnabel and R. Sepulchre. 2009. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070.

J. N. Coates and D. Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proceedings of NAACL-HLT 2018*, pages 194–198.

P. S. Dhillon, D. P. Foster, and L. H. Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research*, 16:3035–3078.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web. ACM*, pages 406–414.

B. Gao, J. Bian, and T.-Y. Liu. 2014. Wordrep: A benchmark for research on learning word representation. Technical report, arXiv preprint arXiv:1407.1640.

G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren. 2012. Large-scale learning of word relatedness with constraint. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414.

F. Hill, R. Reichart, and A. Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, pages 665–695.

P. Jawanpuria, A. Balgovind, A. Kunchukuttan, and B. Mishra. 2019. Learning multilingual word embeddings in latent metric space: A geometric approach. *Transactions of the Association for Computational Linguistics*, 7:107–120.

D. A. Jurgens, P. D. Turney, S. M. Mohammad, and K. J. Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364.

J. M. Lee. 2003. *Introduction to smooth manifolds*, second edition, volume 218 of *Graduate Texts in Mathematics*. Springer-Verlag, New York.

T. Luong, R. Socher, and C. D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL)*, pages 104–113.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NeurIPS)*, pages 3111–3119.

G. A. Miller and W. G. Charles. 1991. Contextual correlates of semantic similarity. *Language and congnitive processes*, pages 1–28.

B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre. 2014. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3):591–621.

A. Muromägi, K. Sirts, and S. Laur. 2017. Linear ensembles of word embedding models. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 96–104.

J. O'Neill and D. Bollegala. 2020. Meta-embedding as auxiliary task regularization. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 14:1532–1543.

H. Rubenstein and J. B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of ACM*, pages 627–633.

J. Townsend, N. Koep, and S. Weichwald. 2016. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5.

J. Turian, L. Ratinov, and B. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 384–394.

W. Yin and H. Shutze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1351–1360.