Learning Meta Word Embeddings by Unsupervised Weighted Concatenation of Source Embeddings

Danushka Bollegala
University of Liverpool, Amazon
danushka@liverpool.ac.uk

Abstract

Given multiple source word embeddings learnt using diverse algorithms and lexical resources, meta word embedding learning methods attempt to learn more accurate and wide-coverage word embeddings. Prior work on meta-embedding has repeatedly discovered that simple vector concatenation of the source embeddings to be a competitive baseline. However, it remains unclear as to why and when simple vector concatenation can produce accurate meta-embeddings. We show that weighted concatenation can be seen as a spectrum matching operation between each source embedding and the meta-embedding, minimising the pairwise inner-product loss. Following this theoretical analysis, we propose two *unsupervised* methods to learn the optimal concatenation weights for creating meta-embeddings from a given set of source embeddings. Experimental results on multiple benchmark datasets show that the proposed weighted concatenated meta-embedding methods outperform previously proposed meta-embedding learning methods.

1 Introduction

Distributed word representations have shown impressive performances in multiple, diverse, downstream NLP applications when used as features [Mikolov *et al.*, 2013; Pennington *et al.*, 2014]. The learning objectives, optimisation methods as well as the lexical resources used in these word embedding learning methods vary significantly, resulting in a diverse set of word embeddings that capture different aspects of lexical semantics such as polysemy [Neelakantan *et al.*, 2014; Reisinger and Mooney, 2010; Huang *et al.*, 2012] and morphology [Cotterell *et al.*, 2016]. Meta-embedding has emerged as a promising solution to combine diverse pre-trained *source* word embeddings for the purpose of producing *meta* word embeddings that preserve the complementary strengths of individual source word embeddings. The input and output word embeddings to the meta-embedding algorithm are referred respectively as the source and meta-embeddings.

The problem setting of meta-embedding learning differs from that of source word embedding learning in several important aspects. In a typical word embedding learning scenario, we would randomly initialise the word embeddings and subsequently update them such that some goodness criterion is optimised such as predicting the log co-occurrences in Global Vectors (**GloVe**) [Pennington *et al.*, 2014] or likelihood in skip-gram with negative sampling (**SGNS**) [Mikolov *et al.*, 2013]. Therefore, the source word embedding methods can significantly differ in their training objectives and optimisation methods being used. On the other hand, for a meta-embedding learning method to be generalisable to different source embedding learning methods, it must be agnostic to the internal mechanisms of the source embedding learning methods. Moreover, the vector spaces as well as their optimal dimensionalities will be different for different source embeddings, which makes it difficult to directly compare source embeddings.

Despite the above-mentioned challenges encountered in meta-embedding learning, it has several interesting properties. From a practitioners point-of-view, meta-embedding is attractive because it obviates the need to pick a single word embedding learning algorithm, which can be difficult because different word embedding learning algorithms perform best in different downstream NLP tasks under different settings [Levy *et al.*, 2015]. Moreover, meta-embedding learning does not require the original linguistic resources (which might not be publicly available due to copyright issues) used to learn the source word embeddings, and operates directly on the (often publicly available) pre-trained word embeddings. Even in cases where the original linguistic resources are available, retraining source word embeddings from scratch can be time consuming and require specialised hardware.

Given a set of source embeddings, a simple yet competitive baseline for creating their meta-embedding is to concatenate the source embeddings [Bollegala *et al.*, 2018; Yin and Schütze, 2016; Goikoetxea *et al.*, 2016]. Concatenation has been justified in prior work as a method that preserves the information contained in individual sources in their corresponding vector spaces. However, this explanation has no theoretical justification and it is unclear how to concatenate source embeddings with different accuracies and dimensionalities, or what losses are being minimised.

Contributions. For word embedding methods that can be expressed as matrix factorisations, we show that their concatenated meta-embedding minimises the *Pairwise Inner Product* (**PIP**) [Yin and Shen, 2018] loss between the sources and an ideal meta-embedding. Specifically, we show that PIP loss

can be decomposed into a *bias* term that evaluates the amount of information lost due to meta-embedding and a series of *variance* terms that account for how source embedding spaces should be aligned with the ideal meta-embedding space to minimise the PIP loss due to meta-embedding. Our theoretical analysis extends the bias-variance decomposition of PIP loss for word embedding [Yin and Shen, 2018] to meta-embedding.

Motivated by the theory, we propose two *unsupervised* methods to learn the optimal concatenation weights by aligning the spectrums of the source embeddings against that of an (estimated) ideal meta-embedding matrix. In particular, no labelled data for downstream tasks are required for learning the optimal concatenation weights. We propose both source-weighted and dimension-weighted concatenation methods. In particular, the dimension-weighted meta-embedding learning method consistently outperforms prior proposals in a range of downstream NLP tasks such as word similarity prediction, analogy detection, part-of-speech tagging, sentiment classification, sentence entailment and semantic textual similarity prediction for various combinations of source embeddings. The source code for reproducing the results reported in this paper is publicly available.[1]

2 Related Work

Yin and Schütze [2016] proposed 1TON, by projecting source embeddings to a common space via source-specific linear transformations. This method minimises squared ℓ_2 distance between the meta and each source embedding assuming a common vocabulary. 1TON+ overcomes this limitation by learning pairwise linear transformations between two given sources for predicting the embeddings for out of vocabulary (OOV) words. Both of these methods can be seen as *globally-linear* transformations because *all* the words in a particular source are projected to the meta-embedding space using the *same* transformation matrix. In contrast, *locally-linear* meta-embedding (LLE) [Bollegala *et al.*, 2018] computes the nearest neighbours for a particular word in each source and then represent a word as the linearly-weighted combination of its neighbours. Next, meta-embeddings are learnt by preserving the neighbourhood constraints. This method does not require all words to be represented by all sources, thereby obviating the need to predict missing source embeddings for OOVs.

Bao and Bollegala [2018] modelled meta-embedding learning as an *autoencoding* problem where information embedded in different sources are integrated at different levels to propose three types of meta-embeddings: Decoupled Autoencoded Meta-Embedding (DAEME) (independently encode each source and concatenate), Concatenated Autoencoded Meta-Embedding (CAEME) (independently decode meta-embeddings to reconstruct each source), and Averaged Autoencoded Meta-Embedding (AAEME) (similar to DAEME but instead of concatenation use average). In comparison to methods that learn globally or locally linear transformations [Bollegala *et al.*, 2018; Yin and Schütze, 2016], autoencoders learn nonlinear transformations. Neill and Bollegala [2018] further extend this approach using squared cosine proximity loss as the reconstruction loss.

[1] https://github.com/LivNLP/meta-concat

Vector concatenation has been used as a baseline for producing meta-embeddings [Yin and Schütze, 2016]. Goikoetxea *et al.* [2016] concatenated independently learnt word embeddings from a corpus and the WordNet. Moreover, applying Principal Component Analysis on the concatenation further improved their performance on similarity tasks. Interestingly, Coates and Bollegala [2018] showed that accurate meta-embeddings can be produced by averaging source embeddings that exist in *different* vector spaces. Recent work [He *et al.*, 2020; Jawanpuria *et al.*, 2020] show that learning orthogonal transformations prior to averaging can further improve accuracy.

Meta-embedding methods have been used for creating sentence-level meta-embeddings from contextualised embeddings [Takahashi and Bollegala, 2022] Both contextualised embeddings and sentence-level meta-embeddings are beyond the scope of this paper, which focuses on context-independent (static) word-level meta-embedding and we defer the interested reader to [Bollegala and O'Neill, 2022] for a comprehensive survey on meta-embedding learning. To the best of our knowledge, we are the first to provide a theoretical analysis of concatenated meta-embedding learning.

3 Meta-Embedding by Concatenation

Let us consider a set of N source word embeddings $s_1, s_2, \ldots, s_N$, all covering a common vocabulary[2] $\mathcal{V}$ of n words (i.e. $|\mathcal{V}| = n$). The embedding of a word w in s_j is denoted by $\boldsymbol{s}_j(w) \in \mathbb{R}^{k_j}$, where k_j is the dimensionality of s_j. We represent s_j by an embedding matrix $\mathbf{E}_j \in \mathbb{R}^{n \times k_j}$. For example, $\mathbf{E}_1$ could be the embedding matrix obtained by running SGNS on a corpus, whereas $\mathbf{E}_2$ could be that obtained from GloVe etc. Then, the problem of meta-embedding via weighted concatenation can be stated as – *what are the optimal weights to concatenate* $\mathbf{E}_1, \ldots, \mathbf{E}_n$ *row-wise such that some loss that represents the amount of information we lose due to meta-embedding is minimised?*

3.1 Two Observations

We build our analysis on two main observations. **First**, we note that word embeddings to be unitary-invariant. Unitary invariance is empirically verified in prior work [Artetxe *et al.*, 2016; Hamilton *et al.*, 2016; Smith *et al.*, 2017] and states that two source embeddings are identical if one can be transformed into the other by a unitary matrix. Unfortunately the dimensions in different source embeddings cannot be directly compared [Bollegala *et al.*, 2017]. To overcome this problem, Yin and Shen [2018] proposed the Pairwise Inner-Product (PIP) matrix, $\mathrm{PIP}(\mathbf{E})$, of $\mathbf{E}$ given by (1) to compare source embeddings via their inner-products over the word embeddings for the same set of words.

$$\mathrm{PIP}(\mathbf{E}) = \mathbf{E}\mathbf{E}^\top \tag{1}$$

If the word embeddings are normalised to unit ℓ_2 length, $\mathbf{E}_j\mathbf{E}_j{}^\top$ becomes the pairwise word similarity matrix. PIP

[2] Missing embeddings can be predicted using, for example, linear projections as in 1TON+.

loss between two embedding matrices $\mathbf{E}_1, \mathbf{E}_2$ is defined as the Frobenius norm of the difference between their PIP matrices:

$$||\mathrm{PIP}(\mathbf{E}_1) - \mathrm{PIP}(\mathbf{E}_2)||_F = \left|\left|\mathbf{E}_1\mathbf{E}_1^\top - \mathbf{E}_2\mathbf{E}_2^\top\right|\right|_F \quad (2)$$

PIP loss enables us to compare word emebddings with different dimensionalities, trained using different algorithms and resources, which is an attractive property when analysing meta embeddings.

Second, we observe that many word embedding learning algorithms such as Latent Semantic Analysis (LSA) [Deerwester *et al.*, 1990], GloVe, SGNS, etc. can be written as either an explicit or an implicit low-rank decomposition of a suitably transformed co-occurrence matrix, computed from a corpus [Li *et al.*, 2015; Dhillon *et al.*, 2015]. For example, LSA applies Singular Value Decomposition (SVD) on a Positive Pointwise Mutual Information (PPMI) matrix, GloVe decomposes a log co-occurrence matrix, and SGNS implicitly decomposes a Shifted PMI (SPMI) matrix.

More generally, if $\mathbf{M}$ is a signal matrix that encodes information about word associations (e.g. a PPMI matrix) and $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be its SVD, then a k-dimensional embedding matrix $\mathbf{E}$ of $\mathbf{M}$ is given by $\mathbf{E} = \mathbf{U}_{1:k}\mathbf{D}^\alpha_{1:k,1:k}$ for some $\alpha \in [0, 1]$ by selecting the largest k singular values (in the diagonal matrix $\mathbf{D}$) and corresponding left singular vectors (in the unitary matrix $\mathbf{U}$). Setting $\alpha = 0.5$ induces symmetric target and context vectors, similar to those obtained via SGNS or GloVe and has been found to be empirically more accurate [Levy and Goldberg, 2014]. The hyperparameter α was found to be controlling the robustness of the embeddings against over-fitting [Yin and Shen, 2018] (See Supplementary for a discussion of α). Given, α and k, a source word embedding learning algorithm can then be expressed as a function $\mathbf{E} = f_{\alpha,k}(\mathbf{M}) = \mathbf{U}_{1:k}\mathbf{D}^\alpha_{1:k,1:k}$ that returns an embedding matrix $\mathbf{E}$ for an input signal matrix $\mathbf{M}$.

Having stated the two observations we use to build our analysis, next we propose two different approaches for constructing meta-embeddings as the weighted concatenation of source embeddings.

3.2 Source-weighted Concatenation

Yin and Schütze [2016] observed that it is important to emphasise accurate source embeddings during concatenation by multiplying all embeddings of a particular source by a source-specific weight, which they tune using a semantic similarity benchmark. Specifically, they compute the *source-weighted* meta-embedding, $\hat{\boldsymbol{e}}_{sw}(w) \in \mathbb{R}^{k_1+\cdots+k_N}$, of a word $w \in \mathcal{V}$ using (3), where $\oplus$ denotes the vector concatenation.

$$\begin{aligned}\hat{\boldsymbol{e}}_{sw}(w) &= c_1\boldsymbol{s}_1(w) \oplus \ldots \oplus c_N\boldsymbol{s}_N(w) \\ &= \oplus_{j=1}^N c_j\boldsymbol{s}_j(w) \end{aligned} \quad (3)$$

Concatenation weights c_j satisfy $\sum_{j=1}^N c_j = 1$. Then, the source-weighted concatenated meta-embedding matrix, $\hat{\mathbf{E}}_{sw}$ is given by (4).

$$\hat{\mathbf{E}}_{sw} = \bigoplus_{j=1}^N c_j\mathbf{E}_j, \quad (4)$$

$\bigoplus$ to denotes the row-wise matrix concatenation.

3.3 Dimension-wighted Concatenation

The number of source embeddings is usually significantly smaller compared to their sum of dimensionalities (i.e. $N \ll \sum_{j=1}^N k_j$). Moreover, the source-weighted concatenation can only adjust the length of each source embedding, and cannot perform any rotations. Therefore, the flexibility of the source-weighting to produce accurate meta-embeddings is limited. To overcome these limitations, we propose a *dimension-weighted* concatenation method given by (5).

$$\hat{\boldsymbol{e}}_{dw}(w) = \oplus_{j=1}^N \mathbf{C}_j\boldsymbol{s}_j(w) \quad (5)$$

Here, $\mathbf{C}_j$ is a diagonal matrix with $c_{j,1}, \ldots, c_{j,k_j}$ in the main diagonal. We require that for all j, $\sum_{i=1}^{k_j} c_{j,i} = 1$. The dimension-weighted concatenated meta-embedding matrix $\hat{\mathbf{E}}_{dw}$ can be written as follows:

$$\hat{\mathbf{E}}_{dw} = \bigoplus_{j=1}^N \mathbf{E}_j\mathbf{C}_j \quad (6)$$

Here, we have $\sum_{j=1}^N k_j (\gg N)$ number of parameters at our disposal to scale each dimension of the sources, which is more flexible than the source-weighted concatenation. Indeed, source-weighting can be seen as a special case of dimension-weighting when $c_j = c_{j,1} = \ldots c_{j,k_j}$ for all j.

3.4 Bias-Variance in Meta-Embedding

Armed with the two key observations in § 3.1, we are now in a position to show how meta-embeddings under source- and dimension weighted concatenation induce a bias-variance tradeoff in the PIP loss. Given that source-weighting is a special case of dimension-weighting, we limit our discussion to the latter. Moreover, for simplicity of the description we consider two sources here but it can be extended any number of sources.[3]

Let us consider the dimension-weighted concatenated meta-embedding $\hat{\mathbf{E}} = [\mathbf{E}_1\mathbf{C}_1; \mathbf{E}_2\mathbf{C}_2]$ of two source embedding matrices $\mathbf{E}_1 \in \mathbb{R}^{n\times k_1}$ and $\mathbf{E}_2 \in \mathbb{R}^{n\times k_2}$ with concatenation coefficient matrices $\mathbf{C}_1 = \mathrm{diag}(c_{1,1}, \ldots, c_{1,k_1})$ and $\mathbf{C}_2 = \mathrm{diag}(c_{2,1}, \ldots, c_{2,k_2})$. $\mathbf{E}_1$ and $\mathbf{E}_2$ are obtained by applying SVD on respective signal matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ and are given by $\mathbf{E}_1 = \mathbf{U}^{(1)}_{\cdot,1:k_1}\mathbf{D}^{(1)\alpha}_{1:k_1,1:k_1}$ and $\mathbf{E}_2 = \mathbf{U}^{(2)}_{\cdot,1:k_2}\mathbf{D}^{(2)\alpha}_{1:k_2,1:k_2}$. The diagonal matrices $\mathbf{D}^{(1)}_{1:k_1,1:k_1} = \mathrm{diag}(\mu_1, \ldots, \mu_{k_1})$ and $\mathbf{D}^{(2)}_{1:k_2,1:k_2} = \mathrm{diag}(\nu_1, \ldots, \nu_{k_2})$ contain the top k_1 and k_2 singular values of respectively of $\mathbf{M}_1$ and $\mathbf{M}_2$.

Let us assume the exsistence of an oracle that provides us with an *ideal* meta-embedding matrix $\mathbf{E} \in \mathbb{R}^{n\times d}$, where $d \geq (k_1 + k_2)$ is the dimensionality of this ideal meta-embedding space. $\mathbf{E}$ is ideal in the sense that it has the minimal PIP loss $\left|\left|\mathrm{PIP}(\mathbf{E}) - \mathrm{PIP}(\hat{\mathbf{E}})\right|\right|_F$ between $\hat{\mathbf{E}}$ created from source embedding matrices $\mathbf{E}_1$ and $\mathbf{E}_2$. Following the low-rank matrix decomposition approach, $\mathbf{E} = \mathbf{U}_{\cdot,1:d}\mathbf{D}^\alpha_{1:d,1:d}$ can be computed using an ideal signal matrix $\mathbf{M}$, where $\mathbf{D}_{1:d,1:d} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$. However, in practice, we are

[3] Indeed we use three sources later in experiments.

not blessed with such oracles and will have to estimate $\mathbf{M}$ via sampling as described later in §3.5. Interestingly, in the case of $\hat{\mathbf{E}}$ constructed by dimension-weighted concatenation, we can derive an upper bound on the PIP loss using the spectral decompositions of $\mathbf{M}, \mathbf{M}_1$ and $\mathbf{M}_2$ as stated in Theorem 1.

Theorem 1. *For two source embedding matrices* $\mathbf{E}_1$ *and* $\mathbf{E}_2$*, the PIP loss between their dimension-weighted meta-embedding* $\hat{\mathbf{E}} = [\mathbf{E}_1\mathbf{C}_1; \mathbf{E}_2\mathbf{C}_2]$ *and an ideal meta-embedding* $\mathbf{E}$ *is given by* (7).

$$\left|\left|\mathrm{PIP}(\mathbf{E}) - \mathrm{PIP}(\hat{\mathbf{E}})\right|\right|_F \leq \underbrace{\sqrt{\sum_{i=k_1+k_2}^{d} \lambda_i^{4\alpha}}}_{\text{bias}} + \underbrace{\sqrt{2}\sum_{i=1}^{k_1}\left(\lambda_i^{2\alpha} - \lambda_{i+1}^{2\alpha}\right)\left|\left|\mathbf{U}_{\cdot,1:i}^{(1)\top}\mathbf{U}_{\cdot,i:n}\right|\right|}_{\text{directional variance in } s_1} + \underbrace{\sqrt{\sum_{i=1}^{k_1}\left(\lambda_i^{2\alpha} - c_{1,i}^2\mu_i^{2\alpha}\right)^2}}_{\text{magnitude variance in } s_1} + \underbrace{\sqrt{\sum_{i=k_1+1}^{k_1+k_2}\left(\lambda_i^{2\alpha} - c_{2,i-k_1}^2\nu_{i-k_1}^{2\alpha}\right)^2}}_{\text{magnitude variance in } s_2} + \underbrace{\sqrt{2}\sum_{i=k_1+1}^{k_1+k_2}\left(\lambda_i^{2\alpha} - \lambda_{i+1}^{2\alpha}\right)\left|\left|\mathbf{U}_{\cdot,1:i}^{(2)\top}\mathbf{U}_{\cdot,i:n}\right|\right|}_{\text{directional variance in } s_2} \quad (7)$$

Proof. See the Appendix in [Bollegala, 2022] for the proof. □

For symmetric signal matrices such as the ones computed via word co-occurrences, embedding matrices are given by the eigendecomposition of the signal matrices and with real eigenvalues. $\mathbf{M}, \mathbf{M}_1, \mathbf{M}_2$ (hence their spectrums) as well as respective optimal dimensionalities d, k_1, k_2 are unknown but can be estimated as described in §3.5. Although we used the same α for the oracle and all sources for simplicity, Theorem 1 still holds when all α are different.

The perturbation bounds on the expected PIP losses are known to be tight [Vu, 2011]. The first term in the R.H.S. in (7) can be identified as a *bias* due to meta-embedding because as we use more dimensions in the sources (i.e. $k_1 + k_2$) for constructing a meta-embedding this term would become smaller, analogous to increasing the complexity of a prediction model. On the other hand, third and fourth terms in the R.H.S. are square roots of the sum of squared differences between the spectra of the ideal meta-embedding and each source embedding, weighted by the concatenation coefficients. These terms capture the variance in the magnitudes of the spectra. Likewise, the second and fifth terms compare the left singular vectors (directions) between each source and the ideal meta-embedding, representing the variance in the directions.

The bias-variance decomposition (7) provides a principled approach for constructing concatenated meta-embeddings. To minimise the PIP loss, we must *match* the spectra of the sources against the ideal meta-embedding by finding suitable concatenation coefficients such that the magnitude variance terms are minimised. Recall that in meta-embedding, the sources are pretrained and given, thus their spectra are fixed. Therefore, PIP loss (7) is a function only of concatenation coefficients for all $c_{1,i}, c_{2,j}$ and their optimal values can be obtained by differentiating w.r.t. those variables. For example, for the first source, under dimension- and source-weighted concatenations the optimal weights are given respectively by (8) and (9).

$$c_{1,i} = \frac{\lambda_i^{\alpha}}{\mu_i^{\alpha}} \quad (8) \qquad c_1 = \sqrt{\frac{\sum_i^{k_1}\lambda_i^{2\alpha}\mu_i^{2\alpha}}{\sum_{i=1}^{k_1}\mu_i^{4\alpha}}} \quad (9)$$

Interestingly, directional variances can be further minimised by, for example, rotating the sources to have orthonormal directions with the ideal meta-embedding. Indeed, prior work [He *et al.*, 2020; Jawanpuria *et al.*, 2020] have experimentally shown that by applying orthogonal transformations to the source embeddings we can further improve the performance in averaged meta-embeddings. Alternatively, if we have a choice on what sources to use, we can use directional variance terms to select a subset of source embeddings for creating meta-embeddings.

3.5 Ideal Embedding Estimation

In meta-embedding learning, we assume that we are given a set of trained source embeddings with specific dimensionalities. Therefore, $k_1, \ldots, k_N, \alpha$ that determine the source embeddings, $\mathbf{E}_1, \ldots, \mathbf{E}_N$, obtained by applying SVD on the respective signal matrices, $\mathbf{M}_1, \ldots, \mathbf{M}_N$, are *not* parameters in the meta-embedding learning task, but are related to the individual source embedding learning tasks. On the other hand, the singular values, $\lambda_1, \ldots, \lambda_d$, of the ideal meta-embedding signal matrix, $\mathbf{M}$, its dimensionality, d, are unknown parameters related to the meta-embedding learning task and must be estimated. In the case of concatenated meta-embeddings discussed in this paper, we have $d = \sum_j k_j$. However, we must still estimate $\lambda_1, \ldots, \lambda_d$ as they are required in the computations of source- and dimension-weighted concatenation coefficients, given respectively by (8) and (9).

For this purpose, we model a signal matrix, $\tilde{\mathbf{M}}$, that we compute from a corpus using some co-occurrence measure as the addition of zero mean Gaussian noise matrix $\mathbf{Z}$ to an ideal signal matrix $\mathbf{M}$, (i.e. $\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{Z}$). Let the spectrum of $\tilde{\mathbf{M}}$ be $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_d$. Next, we randomly split the training corpus into two equally large subsets, and compute two signal matrices: $\tilde{\mathbf{M}}_1 = \mathbf{M} + \mathbf{Z}_1$ and $\tilde{\mathbf{M}}_2 = \mathbf{M} + \mathbf{Z}_2$, where $\mathbf{Z}_1, \mathbf{Z}_2$ are two independent copies of noise with variance $2\sigma^2$. Observing that $\tilde{\mathbf{M}}_1 - \tilde{\mathbf{M}}_2 = \mathbf{Z}_1 - \mathbf{Z}_2$ is a random matrix with zero mean and $4\sigma^2$ variance, we can estimate the noise for symmetric signal matrices by $\sigma = \frac{1}{2n}\left|\left|\tilde{\mathbf{M}}_1 - \tilde{\mathbf{M}}_2\right|\right|_F$.

Then, we can estimate the spectrum of the ideal signal matrix $\lambda_1, \ldots, \lambda_d$ from the spectrum of the estimated signal matrix using universal singular value thresholding [Chatterjee, 2012] as $\lambda_i = \max(\tilde{\lambda}_i - 2\sigma\sqrt{n}, 0)$. The rank of $\mathbf{M}$ is determined by the smallest $(i+1)$ for which $\tilde{\lambda}_i \leq 2\sigma\sqrt{n}$. Although we only require the spectra of the ideal signal matrices for computing the meta-embedding coefficients, if we wish we could reconstruct $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^{\top}$, where $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, $\mathbf{U}$ and $\mathbf{V}$ are random orthonormal matrices, obtained via SVD on a random matrix of suitable shape.

	Method	SimLex	SimVerb	PoS
source	GloVe	22.66	9.95	77.06
	SGNS	29.93	12.47	87.28
	LSA	32.64	21.01	85.22
meta	UW	37.37	21.37	87.92
	AVG	36.11	19.47	87.88
	SVD	36.22	19.55	87.71
	SW	35.97	21.59	88.11
	DW	**39.12**	**23.99**	**88.63**

Table 1: Comparing different weighting methods.

4 Experiments

4.1 Effect of Weighted Concatenation

To evaluate the different weighting methods, we create the meta-embeddings of the following three source embeddings: **GloVe** embeddings trained on the Wikipedia Text8 corpus (17M tokens)[4], **SGNS** embeddings trained on the WMT21 English News Crawl (206M tokens),[5] and **LSA** embeddings trained on an Amazon product review corpus (492M tokens) [Ni *et al.*, 2019].

For GloVe, the elements of the signal matrix M_{ij} are computed as $\log(X_{ij})$ where X_{ij} is the frequency of co-occurrence between word w_i and w_j. The elements of the signal matrix for SGNS are set to $M_{ij} = \mathrm{PMI}(w_i, w_j) - \log \beta$, where $\mathrm{PMI}(w_i, w_j)$ is the PMI between w_i and w_j. For LSA, the elements of its signal matrix are set to $M_{ij} = \max(\mathrm{PMI}(w_i, w_j), 0)$. Source embeddings are created by applying SVD on the signal matrices.

The optimal dimensionality of a source embedding is determined by selecting the dimensionality that minimises the PIP loss between the computed and ideal source embedding matrices for varying dimensionalities upto their estimated rank. The estimated optimal dimensionality for GloVe ($k_1 = 1707$) is significantly larger than that for SGNS ($k_2 = 303$) and LSA ($k_3 = 220$). This is because the estimated noise for GloVe from the Text8 corpus is significantly smaller ($\sigma = 0.0891$) compared to that for SGNS ($\sigma = 0.3292$) and LSA ($\sigma = 0.3604$), enabling us to fit more dimensions for GloVe for the same cost in bias. We used the MTurk-771 as a development dataset to estimate the co-occurrence window (set to 5 tokens) and β (set to 3) in our experiments. Moreover, α is set to 0.5 for all source embeddings, which reported the best performance on MTurk-771.

We compare the proposed source-weighted (**SW**) and dimension-weighted (**DW**) concatenation methods against several baselines: unweighted concatenation (**UW**) where we concatenate the source embeddings without applying any weighting, averaging (**AVG**) the source embeddings as proposed by Coates and Bollegala [2018] after padding zeros to sources with lesser dimensionalities, and applying **SVD** on **UW** to reduce the dimensionality to 200, which reported the best performance on the development data.

[4]http://mattmahoney.net/dc/textdata.html

[5]http://statmt.org/wmt21/translation-task.html

Table 1 shows the Spearman correlation coefficient on two true word similarity datasets – SimLex and SimVerb, which are known to be reliable true similarity datasets [Faruqui *et al.*, 2016]. To evaluate meta-embeddings for Part-of-Speech (PoS) tagging as a downstream task, we initialise an LSTM with pre-trained source/meta embeddings and train a PoS tagger using the CoNLL-2003 train dataset. PoS tagging accuracy on the CoNLL-2003 test dataset is reported for different embeddings in Table 1. Among the three sources, we see significant variations in performance over the tasks, where LSA performs best on SimLex and SimVerb, while SGNS on PoS. This is due to the size and quality of corpora used to train the source embeddings, and simulate a challenging meta-embedding learning problem. UW remains a strong baseline, outperforming AVG and SVD, and even SW in SimLex. However, SW outperforms UW in SimVerb and PoS tasks. Overall, DW reports best performance on all tasks demonstrating its superior ability to learn accurate meta-embeddings even in the presence of weak source embeddings.

4.2 Comparisons against Prior Work

We compare against previously proposed meta-embedding learning methods such as **LLE**, **1TON**, **DAEME**, **CAEME**, **AAEME** described in § 2. Unfortunately, prior work have used different sources and corpora which makes direct comparison of published results impossible. To conduct a fair evaluation, we train GloVe ($k = 736, \sigma = 0.1472$), SGNS ($k = 121, \sigma = 0.3566$) and LSA ($k = 119, \sigma = 0.3521$) on the Text8 corpus as the source embeddings. For prior methods, we use the publicly available implementations by the original authors. The average run times of SW and DW are ca. 30 min (wall clock time) measured on a EC2 p3.2xlarge instance. Hyperparameters for those methods were tuned on MTurk-771 as reported in the Supplementary.

In Table 2, we use evaluation tasks and benchmark datasets used in prior work: (1) *word similarity prediction* (Word Similarity-353 (WS), Rubenstein-Goodenough (RG), rare words (RW), Multimodal Distributional Semantics (MEN), (2) *analogy detection* (Google analogies (GL), Microsoft Research syntactic analogy dataset (MSR), (3) *sentiment classification* (movie reviews (MR), customer reviews (CR), opinion polarity (MPQA)), (4) *semantic textual similarity* benchmark (STS), (5) *textual entailment* (Ent) and (6) *relatedness* (SICK). For computational convenience, we limit the vocabulary to the most frequent 20k words. Note that this is significantly smaller compared to vocabulary sizes and training corpora used in publicly available word embeddings (e.g. GloVe embeddings are trained on 42B tokens from Common Crawl corpus covering 2.8M word vocabulary). Therefore, the absolute performance scores we report here cannot be compared against SoTA on these benchmark. The goal in our evaluation is to compare the relative performances among the different meta-embedding methods.

From Table 2 we see that the proposed SW and DW methods report the best performance in all benchmarks. In particular, the performance of DW in RW, CR and SW in GL, MSR are statistically significant over the second best methods for those datasets. Among the three sources, we see that SGNS and LSA perform similarly in most tasks. Given that both SGNS

Method	WS	RG	RW	MEN	GL	MSR	MR	CR	MPQA	Ent	SICK	STS
GloVe	69.46	47.55	47.61	71.34	37.06	37.01	67.25	74.78	80.51	75.48	65.54	50.60
SGNS	71.26	79.66	48.71	71.43	39.48	37.07	64.44	72.61	79.27	75.54	64.48	50.30
LSA	71.81	80.39	50.36	72.79	40.05	37.66	64.78	71.97	79.12	74.85	62.41	46.84
SVD	70.11	75.00	49.13	70.44	39.35	36.70	63.10	71.44	78.12	73.68	61.70	47.33
AVG	67.92	72.06	46.32	70.84	39.96	36.95	67.38	75.95	80.71	76.33	67.58	51.87
LLE	64.23	69.85	41.90	69.96	23.74	22.41	62.46	65.53	76.00	62.03	56.33	45.99
1TON	70.08	72.06	48.01	69.52	38.89	35.80	60.18	64.29	77.56	61.92	55.31	44.43
CAEME	69.94	79.33	50.47	72.07	43.55	39.21	59.63	66.12	78.59	72.69	61.80	50.85
DAEME	62.02	52.70	43.56	66.55	44.72	38.71	58.22	65.93	79.16	72.55	62.25	50.78
AAEME	69.94	79.33	50.47	72.07	43.20	39.52	59.63	66.12	78.59	72.69	61.80	50.85
UW	72.88	81.37	49.43	72.27	44.80	40.93	67.34	75.71	81.02	76.50	68.89	54.39
SW	73.03	**81.62**	49.59	72.27	**54.10***	**50.78***	69.03	74.89	81.61	76.84	68.43	55.44
DW	**74.37**	81.37	**54.24***	**74.35**	50.25	46.58	**69.37**	**76.77***	**82.22**	**77.78**	**69.01**	**55.98**

Table 2: Performance of source embeddings (top) baselines/prior work (middle) and concatenated meta-embeddings (bottom) for tasks described in §4.2. Best performance for each dataset is shown in bold, whereas * denotes when this is statistically significantly better than the second best method for the same dataset.

and LSA use variants of PMI[6] this behaviour is to be expected. However, we note that the absolute performance of source embeddings usually improves when trained on larger corpora and thus what is important in meta-embedding evaluation is not the *absolute* performance of the sources, but how much *relative* improvements one can obtain by applying a meta-embedding algorithm on top of the pre-trained source embeddings. In this regard, we see that in particular **DW** significantly improves over the best input source embedding in all tasks, except in **RG** where the improvement over **LSA** is statistically insignificant.

We see that UW is a strong baseline, outperforming even more complex prior proposals. By learning unsupervised concatenation weights, we can further improve UW as seen by the performance of SW and DW. In particular, DW outperforms SW in all datasets, except in RG, GL and MSR. RG is a smaller (65 word pairs) dataset and the difference in performance between SW and DW there is not significant. Considering that GL and MSR are datasets for word analogy prediction, we see that SW is particularly better for analogy tasks.

SVD and AVG perform even worse than some of the sources due misaligned dimensionalities. Therefore, when the sources are of different dimensionalities the projection matrices have different numbers of parameters, requiring careful balancing. We recommend that future work evaluate the robustness in performance considering the optimal dimensionalities of the sources. To ablate sources, we consider all pairwise meta-embeddings in Table 3, which shows the performance on word similarity (MEN) and multiple STS benchmarks (STS 13, 14, 15 and 16). We see that DW consistently outperforms all other methods, showing the effectiveness of dimension-weighing when meta-embedding sources of different dimensionalities and performances.

5 Conclusion

We showed that concatenated meta-embeddings minimise PIP loss and proposed two weighted-concatenation methods. Experiments showed that the dimension-weighted meta-embeddings outperform prior work on multiple tasks. We plan to extend this result to contextualised embeddings.

[6]when $\beta = 1$ SPMI becomes PMI

	Method	MEN	STS13	STS14	STS15	STS16
GloVe+SGNS	UW	72.45	45.13	49.91	54.18	41.93
	SW	72.64	43.86	48.83	53.17	41.04
	DW	**74.27**	**46.37**	**51.93**	**55.84**	**44.36**
	CAEME	72.63	43.11	48.46	53.48	41.56
	DAEME	64.51	42.15	48.69	53.33	41.93
	AAEME	71.70	44.24	48.39	53.32	41.67
GloVe+LSA	UW	73.58	44.95	49.95	54.52	42.52
	SW	73.30	43.85	48.91	53.33	41.23
	DW	**74.21**	**46.39**	**52.3**	**56.38**	**44.63**
	CAEME	73.35	43.22	48.85	54.05	42.03
	DAEME	64.02	42.63	49.21	53.35	42.09
	AAEME	72.40	44.63	49.01	53.92	42.62
SGNS+LSA	UW	72.21	45.23	49.94	53.13	42.47
	SW	71.81	45.28	49.96	52.81	42.14
	DW	**73.10**	**46.48**	**51.47**	**53.85**	**43.73**
	CAEME	72.21	43.77	47.90	53.03	41.42
	DAEME	71.13	43.50	47.74	52.57	41.98
	AAEME	72.13	43.86	48.12	53.16	41.65

Table 3: Pairwise meta-embedding using two sources.

References

[Artetxe *et al.*, 2016] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP*, pages 2289–2294, 2016.

[Bao and Bollegala, 2018] Cong Bao and Danushka Bollegala. Learning word meta-embeddings by autoencoding. In *COLING*, pages 1650–1661, 2018.

[Bollegala and O'Neill, 2022] Danushka Bollegala and James O'Neill. A survey on word meta-embedding learning. In *Proc. of IJCAI*, 2022.

[Bollegala *et al.*, 2017] Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. Learning linear transformations between counting-based and prediction-based word embeddings. *PLoS ONE*, 12(9):1–21, September 2017.

[Bollegala *et al.*, 2018] Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. Think globally, embed locally — locally linear meta-embedding of words. In *Proc. of IJCAI-EACI*, pages 3970–3976, 2018.

[Bollegala, 2022] Danushka Bollegala. Learning meta word embeddings by unsupervised weighted concatenation of source embeddings. *10.48550/ARXIV.2204.12386*, 2022.

[Chatterjee, 2012] Sourav Chatterjee. Matrix estimation by Universal Singular Value Thresholding. *Annals of Statistics*, 2012.

[Coates and Bollegala, 2018] Joshua Coates and Danushka Bollegala. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proc. of NAACL-HLT*, pages 194–198, 2018.

[Cotterell *et al.*, 2016] Ryan Cotterell, Hinrich Schütze, and Jason Eisner. Morphological smoothing and extrapolation of word embeddings. In *Proc. of ACL*, 2016.

[Deerwester *et al.*, 1990] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[Dhillon *et al.*, 2015] Paramveer S. Dhillon, Dean P. Foster, and Lyle H. Ungar. Eigenwords: Spectral word embeddings. *JMLR*, 16:3035–3078, 2015.

[Faruqui *et al.*, 2016] Mannal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *Eval Vec*, pages 30–35, 2016.

[Goikoetxea *et al.*, 2016] Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. Single or multiple? combining word representations independently learned from text and wordnet. In *Proc. of AAAI*, pages 2608–2614, 2016.

[Hamilton *et al.*, 2016] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *ACL*, pages 1489–1501, 2016.

[He *et al.*, 2020] Jingyi He, KC Tsiolis, Kian Kenyon-Dean, and Jackie Chi Kit Cheung. Learning Efficient Task-Specific Meta-Embeddings with Word Prisms. In *Proc. of COLING*, 2020.

[Huang *et al.*, 2012] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proc. of ACL*, pages 873–882, 2012.

[Jawanpuria *et al.*, 2020] Pratik Jawanpuria, Satya Dev N T V, Anoop Kunchukuttan, and Bamdev Mishra. Learning geometric word meta-embeddings. In *Reps4NLP*, pages 39–44, Online, 2020.

[Levy and Goldberg, 2014] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NuerIPS*, 2014.

[Levy *et al.*, 2015] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225, 2015.

[Li *et al.*, 2015] Shaohua Li, Jun Zhu, and Chunyan Miao. A generative word embedding model and its low rank positive semidefinite solution. In *EMNLP*, pages 1599–1609, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, and Jeffrey Dean. Efficient estimation of word representation in vector space. In *Proc. of ICLR*, 2013.

[Neelakantan *et al.*, 2014] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proc. of EMNLP*, pages 1059–1069, 2014.

[Neill and Bollegala, 2018] James O' Neill and Danushka Bollegala. Angular-Based Word Meta-Embedding Learning. *10.48550/arXiv.1808.04334*, 2018.

[Ni *et al.*, 2019] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*, pages 188–197, 2019.

[Pennington *et al.*, 2014] Jeffery Pennington, Richard Socher, and Christopher D. Manning. Glove: global vectors for word representation. In *Proc. of EMNLP*, pages 1532–1543, 2014.

[Reisinger and Mooney, 2010] Joseph Reisinger and Raymond J. Mooney. Multi-prototype vector-space models of word meaning. In *Proc. of HLT-NAACL*, pages 109–117, 2010.

[Smith *et al.*, 2017] Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformation and the inverted softmax. In *Proc. of ICLR*, 2017.

[Takahashi and Bollegala, 2022] Keigo Takahashi and Danushka Bollegala. Unsupervised attention-based sentence-level meta-embeddings from contextualised language models. In *Proc. of LREC*, 2022.

[Vu, 2011] Van Vu. Singular vectors under random perturbation. *Random Structures & Algorithms*, 39(4):526–538, May 2011.

[Yin and Schütze, 2016] Wenpeng Yin and Hinrich Schütze. Learning meta-embeddings by using ensembles of embedding sets. In *Proc. of ACL*, pages 1351–1360, 2016.

[Yin and Shen, 2018] Zi Yin and Yuanyuan Shen. On the dimensionality of word embedding. In *Proc. of NeurIPS*, pages 887–898, 2018.