Angular-Based Word Meta-Embedding Learning

James O' Neill and Danushka Bollegala
Department of Computer Science, University of Liverpool
Liverpool, L69 3BX
England
{james.o-neill,danushka.bollegala}@liverpool.ac.uk

Abstract

Ensembling word embeddings to improve distributed word representations has shown good success for natural language processing tasks in recent years. These approaches either carry out straightforward mathematical operations over a set of vectors or use unsupervised learning to find a lower-dimensional representation. This work compares meta-embeddings trained for different losses, namely loss functions that account for angular distance between the reconstructed embedding and the target and those that account normalized distances based on the vector length. We argue that meta-embeddings are better to treat the ensemble set equally in unsupervised learning as the respective quality of each embedding is unknown for upstream tasks prior to meta-embedding. We show that normalization methods that account for this such as cosine and KL-divergence objectives outperform meta-embedding trained on standard ℓ_1 and ℓ_2 loss on *defacto* word similarity and relatedness datasets and find it outperforms existing meta-learning strategies.

1 Introduction

Meta-embeddings are a quick and useful prior step for improving word representations in natural language learning tasks. This involves combining several learned embeddings in a way that improve the overall input representation. This approach is a less computationally expensive compared to if a practitioner were to train a set of word embeddings from scratch, particularly when considering non-sliding window methods. The most straightforward approaches to meta-embeddings are: concatenation (CONC) and averaging (AV). The former is limited since the dimensionality grows large for multiple embeddings as more vectors are concatenated and the latter, while fast, does not preserve most of the information encoded in each embedding when taking the arithmetic mean. Although, it would seem surprising concatenating vectors from different embedding spaces is valid, it has been shown that (Coates and Bollegala, 2018) AV approximates CONC even though the embedding spaces are different. Although, to address the loss of information when using AV, Singular Value Decomposition has been used as a dimensionality reduction technique to factorize the embeddings into a lower-rank approximation of the concatenated meta-embedding set.

Linear methods include the use of a projection layer for meta-embedding (known as 1TON) Yin and Schütze (2015), which is simply trained using an ℓ_2-based loss. Similarly, Bollegala et al. (Bollegala et al., 2017) has focused on finding a linear transformation between count-based and prediction-based embeddings, showing that linearly transformed count-based embeddings can be used for predictions in the localized neighborhoods in the target space.

Most recent work (Bao and Bollegala, 2018) has focused on the use of an autoencoder (AE) to encode a set of N pretrained embeddings using 3 different variants: (1) Decoupled Autoencoded Meta Embeddings (DAEME) that keep activations separated for each respective embedding input during encoding and uses a reconstruction loss for both predicted embeddings while minimizing the loss for each respective decoded output, (2) Coupled Autoencoded Meta Embeddings (CAEME) which instead learn to predict from a shared encoding and (3) Averaged Autoencoded Meta-Embedding (AAME) is simply an averaging of the embedding set as input instead of using a concatenation. This is the most relevant work to our paper, hence, we include these 3 autoencoding schemes along with aforementioned methods for experiments, described in Section 3. We also include two subtle variations of the aforementioned

AEs. The first predicts a target embedding from an embedding set using the remaining embedding set, post-learning the single hidden layer is used as the word meta-embedding. The second method is similar except an AE is used for each input embedding to predict the designated target embedding, followed by an averaging over the resulting hidden layers. This alternative is described in more detail in Section 2.

The aforementioned previously proposed unsupervised learning have a common limitation, that is minimising the Euclidean (ℓ_2) distance between source word embeddings and their meta-embedding. Arora et al. (2016) have shown that the ℓ_2 norm of a word embedding vector is proportional to the frequency of the corresponding word in the corpus used to learn the word embeddings. Considering that in meta-embedding learning we use source embeddings trained on different resources, we argue that it is more important to preserve the semantic orientation of words, which is captured by the angle between two word embeddings, not their length. Indeed, cosine similarity, a popularly used measure for computing the semantic relatedness among words, ignores the length related information. Additionally, we note the relationship between KL-divergence and cosine similarity in the sense both methods perform a normalization that is proportional to the semantic information. Hence, we compare several popular measures such as MSE and MAE that use ℓ_2 and ℓ_1 respectively, against KL-divergence and cosine similarity for the purpose of learning meta-embeddings and show that the loss which accounts for this orientation consistently outperforms the former objectives that only consider length. We demonstrated this across multiple benchmark datasets.

2 Methodology

Before describing the loss functions used, we explain the aforementioned variation on the autoencoding method and how it slightly differs from 1TON/1TON$^+$ (Yin and Schütze, 2015) and standard AEs (Bao and Bollegala, 2018) presented in previous work. Target Autoencoders (TAE) are defined as learning an ensemble of nonlinear transformations between sets of bases X_s in sets of vector spaces $\mathcal{X}_S = \{\mathcal{X}_1, .., \mathcal{X}_s, .., \mathcal{X}_N\}$ $s.t$ $\mathcal{X}_s \in \mathbb{R}^{|v_s| \times d_s}$ to a target space $\mathcal{X}_t \in \mathcal{R}^{|v_t| \times d_t}$, where $\mathtt{f}_w^{(i)} : \mathcal{X}_S^{(i)} \rightarrow \mathcal{X}_t$ $\forall i$ is the nonlinear transformation function used to make the mapping. Once a set of M number of parameteric models $\mathtt{f}_w = \{\mathtt{f}_w^{(1)}, \mathtt{f}_w^{(i)}, .., \mathtt{f}_w^{(M)}\}$ are trained with various objective functions to learn the mappings between the word vectors we obtain a set of lower-dimensional target latent representation that represents different combinations of mappings from one vector space to another. After training, all H set of latent variables $Z_s = \{z_1, .., z_H\}$ are concatenated with an autoencoded target vector. This means thats all vector spaces have been mapped to a target space and there hidden meta-word representations have been averaged, as illustrated in Figure 1.

Figure 2 shows a comparison of the previous autoencoder approaches (Bao and Bollegala, 2018) (left) and the alternative AE (right), where dashed lines indicate connections during training and bold lines indicate prediction. The Concat-AutoEncoder (CAEME) simply concatenates the embedding set into a single vector and trains the autoencoder so to produce a lower-dimensional representation (shown in red), while the decoupled autoencoder (DAEME) keeps the embedding vectors separate in the encoding. In contrast the target encoder (TAE) is similar to that of CAEME only the label is a single embedding from the embedding set and the input are remaining embeddings from the set. After training, TAE then concatenates the hidden layer encoding with the original target vector. The Mean Target AutoEncoder (MTE) instead performs an averaging over separate autoencoded representation.

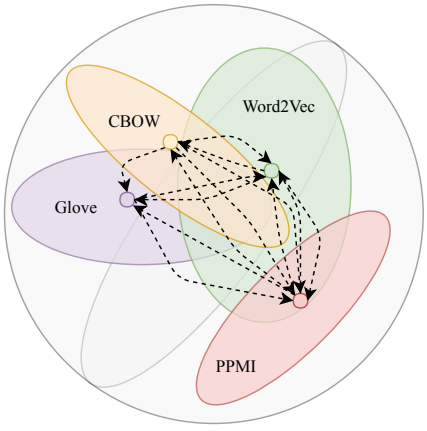

Figure 1: TAE Embedding

AutoEncoder Settings The standard Autoencoder (AE) is a 1-hidden layer AE of hidden layer

dimension $h_d = 200$. Weights are initialized with a normal distribution, mean $\mu = 0$ and standard deviation $\sigma = 1$. Dropout is used with a dropout rate $p = 0.2$ for all datasets. The model takes all unique vocabulary terms pertaining to all tested word association and word similarity datasets ($n = 4819$) and performs Stochastic Gradient Descent (SGD) with batch size $\tilde{x} = 32$ trained between 50 epochs for each dataset $\forall d \in \mathcal{D}$. This results in a set of vectors $X_j \in \mathbb{R}^{|v_j| \times 200}$ $\forall j$ that are then used for finding the similarity between word pairs. The above parameters were chosen (h_d, $\tilde{x}$ and number of epochs) over a small grid search. As stated, we compare against previous methods (Yin and Schütze, 2015; Bao and Bollegala, 2018) that use ℓ_2 distance, as shown in Equation 1).

$$\mathcal{L}_\theta(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2 \quad (1)$$

Similarly, the Mean Absolute Error (ℓ_1 norm of difference) loss $\frac{1}{N}\sum_{i=1}^{N}|y - \hat{y}|$ is tested. We also compare against a KL divergence objective, as shown in Equation 2, $\hat{y}$ is the last activation output from the log-softmax that represents $q(x)$ and the KL-divergence is given as $KL(p|q) = \sum_{i=1}^{N} p(x_i)\log\big(q(x_i)/p(x_i)\big)$.

$$\mathcal{L}_\theta(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N} y^{(i)} \cdot \Big(\log\big(y^{(i)}\big) - \hat{y}^{(i)}\Big) \quad (2)$$

Since $\tanh$ functions are used and input vectors are ℓ_2 normalized we propose a Squared Cosine

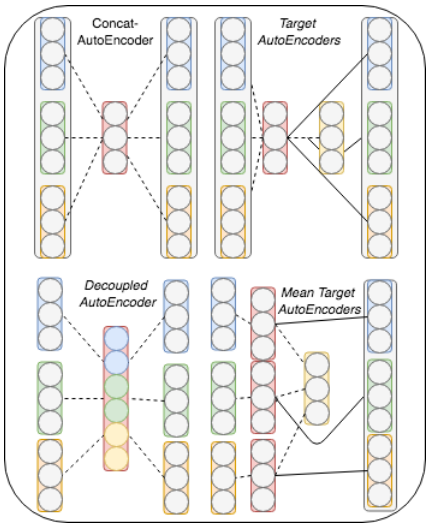

Figure 2: AE Meta-Embedding Methods

Proximity (SCP) loss, shown in Equation 3 where m is the output dimensionality. This forces the optimization to tune weights such that the rotational difference between the embedding spaces is minimized, thus preserving semantic information in the reconstruction. In the context of its utility for the TAE, we too want to minimize the angular difference between corresponding vectors in different vector spaces. It is also a suitable fit since it is a proper distance measure (i.e symmetric), unlike KL-divergence.

$$\mathcal{L}_\theta(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\sum_{j=1}^{m}\hat{y}_{ij} * y_{ij}}{\sqrt{\sum_{ij}^{m}\hat{y}_{ij}^2}\sqrt{\sum_{ij}^{m}y_{ij}^2}}\right)^2 \quad (3)$$

The model is kept relatively simple so that the comparisons against previous methods are directly comparable and that the performance comparison between the proposed SCP loss and KL divergence loss against MSE and MAE is controlled. Additionally, all comparison are that of models which are only trained from co-occurrence statistics that are not leveraging any external knowledge (e.g AutoExtend (Rothe and Schütze, 2015)).

3 Experiments

The following word association and word similarity datasets are used throughout experimentation: Simlex (Hill et al., 2015), WordSim-353 (Finkelstein et al., 2001), RG (Rubenstein and Goodenough, 1965), MTurk (MechanicalTurk-771) (Halawi et al., 2012), RareWord (RW) (Luong et al., 2014) and MEN (Bruni et al., 2012). The word vectors considered in the embeddings set are `skipgram` and `cbow` (Mikolov et al., 2013), FastText (Bojanowski et al., 2016), LexVec (Salle et al., 2016), Hellinger PCA (HPCA) (Lebret and Collobert, 2013) and Hierarchical Document Context (HDC) (Sun et al., 2015). We now report results on the performance of meta-embedding autoencodings with various loss functions, while also presenting target autoencoders for combinations of word embeddings and compare against existing current SoTA meta-embeddings.

Table 1 shows the scaled Spearman correlation test scores, where (1) shows the original single embeddings, (2) results for standard meta-embedding approaches that either apply a single mathematical operation or employ a linear projection as an encoding, (3) presents the re-

sults using autoencoder schemes by (Bao and Bollegala, 2018) that we have used to test the various losses, (4) introduces TAE without concatenating the target Y embedding post-training with MSE loss and (5) shows the results of concatenating Y with the lower-dimensional (200-dimensions) vector that encodes all embeddings apart from the target vector. Therefore reported results from (4) are of a 200d vector, while (5) concatenates the vector leading to a vector between 300-500 dimensions dependent on the target vector. All trained encodings from sections 3-5 are 200-dimensional vectors. Results in red shading indicate the best performing meta-embedding for all presented approaches, while black shading indicates the best performing meta-embedding for the respective section.

Best performing word meta-embeddings are held between concatenated autoencoders that use the proposed Cosine-Embedding loss, while a KL-divergence also performs well on Simlex and RareWord. Interestingly, both of these dataset are distinct in that Simlex is the only dataset providing scores on *true similarity* instead of free association, which has shown to be more difficult for word embeddings to account for (Hill et al., 2016), while Rareword provides morphologically complex words to find similarity between. Concretely, it would seem KL-divergence is well suited for encoding when the word relations exhibits of a more complex or rare nature. Similarly, we find SCP loss to achieve best results on RG and MEN, both the smallest and largest datasets of the set. Furthermore, the TAE variant has lead to competitive performance overall against other meta-embedding approaches and even produces best results on WS353. Lastly, we find that HPCA embeddings are relatively weak for word similarity.

1. Embeddings	Simlex	WS353	RG	MTurk	RW	MEN
Skipgram	44.19	77.17	76.08	68.15	49.70	75.85
FastText	38.03	75.33	79.98	67.93	47.90	76.36
GloVe	37.05	66.24	76.95	63.32	36.69	73.75
LexVec	41.93	64.79	76.45	71.15	48.94	80.92
HPCA	16.60	57.11	41.72	37.45	13.36	34.90
HDC	40.68	76.81	80.58	65.76	46.34	76.03
2. Standard Meta						
CONC	42.57	72.13	81.36	71.88	49.91	80.33
SVD	41.10	72.06	81.18	71.50	49.13	79.85
AV	40.63	70.5	80.05	70.51	49.28	78.31
1TON	41.30	70.19	80.20	71.52	50.80	80.39
1TON*	41.49	70.60	78.40	71.44	50.86	80.18
3. ℓ_2-AE						
Decoupled	42.56	70.62	82.81	71.16	50.79	80.33
Concatenated	43.10	71.69	84.52	71.88	50.78	81.18
ℓ_1-AE						
Decoupled	43.52	70.30	82.91	71.43	51.48	81.16
Concatenated	44.41	70.96	81.16	69.63	51.89	80.92
Cosine-AE						
Decoupled	43.13	71.96	84.23	70.88	50.20	81.02
Concatenated	44.85	72.44	85.41	70.63	50.74	81.94
KL-AE						
Decoupled	44.13	71.96	84.23	70.88	50.20	81.02
Concatenated	45.10	74.02	85.34	67.75	53.02	81.14
4. TAE						
→Skipram	37.80	67.33	76.50	63.41	37.52	74.86
→ FastText	38.17	66.62	77.184	64.73	37.84	74.77
→Glove	39.95	77.14	81.58	68.82	47.94	76.67
→ LexVec	37.48	67.19	75.98	63.96	37.70	74.75
→HPCA	40.78	65.79	38.64	59.49	38.65	74.50
→HDC	38.15	66.96	76.62	63.08	37.53	76.62
5. TAE +Y						
→ Skipram	42.43	75.33	80.11	66.51	44.77	78.98
→ FastText	41.69	72.65	80.51	67.64	47.41	77.48
→ Glove	41.75	76.65	82.40	68.92	48.83	78.27
→ LexVec	42.85	73.33	80.97	69.17	46.71	79.63
→ HPCA	40.03	69.65	70.43	61.31	36.38	73.10
→ HDC	42.43	74.08	80.11	66.51	44.76	77.93

Table 1: Meta-Embedding Results

4 Conclusion

We find the meta-embeddings trained using Autoencoders with a Squared Cosine loss and a KL-divergence loss improves performance in the majority of cases, reinforcing the argument that accounting for angles explicitly through normalization (log-softmax for KL) is an important criterion for encoding. It is particularly useful for distributed word representations, since embeddings are learned from large documents of varying length and semantics. Lastly, we have shown its use in the context of improving meta-embeddings, although this suggests cosine loss is also suitable for minimizing angular differences for word embeddings, not only for meta-embeddings. Concretely, this paper has carried out a comprehensive study of methods to embed a lower-dimensional representation from embedding sets, while proposing losses that explicitly keep angular information intact for meta-embeddings.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Cong Bao and Danushka Bollegala. 2018. Learning word meta-embeddings by autoencoding. *COLING*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vec-

tors with subword information. *arXiv preprint arXiv:1607.04606*.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2017. Learning linear transformations between counting-based and prediction-based word embeddings. *PloS one*, 12(9):e0184544.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding–computing meta-embeddings by averaging source word embeddings. *arXiv preprint arXiv:1804.05262*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

Rémi Lebret and Ronan Collobert. 2013. Word emdeddings through hellinger pca. *arXiv preprint arXiv:1312.5542*.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Alexandre Salle, Marco Idiart, and Aline Villavicencio. 2016. Matrix factorization using window sampling and negative sampling for improved word representations. *arXiv preprint arXiv:1606.00819*.

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2015. Learning word representations by jointly modeling syntagmatic and paradigmatic relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 136–145.

Wenpeng Yin and Hinrich Schütze. 2015. Learning meta-embeddings by using ensembles of embedding sets. *arXiv preprint arXiv:1508.04257*.