arXiv:1809.05886v1 [cs.CL] 16 Sep 2018

Semi-Supervised Multi-Task Word Embeddings

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

Word embeddings have been shown to benefit from ensembling several word embedding sources, often carried out using straightforward mathematical operations over the set of vectors to produce a meta-embedding representation. More recently, unsupervised learning has been used to find a lower-dimensional representation, similar in size to that of the word embeddings within the ensemble. However, these methods do not use the available manual labeled datasets that are often used solely for the purpose of evaluation.

We propose to improve word embeddings by simultaneously learning to reconstruct an ensemble of pretrained word embeddings with supervision from various labeled word similarity datasets. This involves reconstructing word meta-embeddings while simultaneously using a Siamese Network to also learn word similarity where both processes share a hidden layer. Experiments are carried out on 6 word similarity datasets and 3 analogy datasets. We find that performance is improved for all word similarity datasets when compared to unsupervised learning methods with a mean increase of 11.33 in the Spearman Correlation coefficient. Moreover, 4 of 6 of word similarity datasets from our approach show best performance when using of a cosine loss for reconstruction and Brier's loss for word similarity.

Introduction

Distributed word representations have shown good performance for tasks in natural language. Given that the performance is dependent on the model used for embedding (e.g `skipgram`, `cbow`), it is clear that each model exploits different aspects of the semantic space. Hence, meta-embedding combines multiple word embeddings to increase the coverage and accuracy of word embeddings. We are further motivated in that adding the additional step of combining several word representations into one representation is a relatively fast computation and only requires a shallow neural network for reconstruction. Moreover, stacked ensembles have shown state of the art performance for various problems in NLP (Jozefowicz et al. 2016; Wang et al. 2017; Bao and Bollegala 2018). Current approaches to meta-word representations rely on unsupervised learning such as autoencoding (Bao and Bollegala 2018) to find a lower-dimensional hidden representation of the ensemble set. This can be advantageous in cases where (1) pre-training is expensive, (2) pre-training embeddings are available but not the algorithm or data and (3) the available embeddings vary in dimensionality size. However, for problems that rely on representations that are better aligned with human judgment, word embeddings and word meta-embeddings alike can find it difficult to preserve from word associations alone. Therefore, we aim to incorporate supervision from these datasets into word meta-embeddings.

Hence, this paper describes our proposed semi-supervised learning approach that combines the benefits of unsupervised learning for finding a lower-dimensional representations of concatenated word meta-embeddings, and also learns to predict word similarity using a siamese network that incorporates a shared representation from an autoencoder. We consider both of these to be individual tasks, hence the reference to multi-task learning. We evaluate our approach on held-out word similarity datasets and also include an evaluation on the transferability of the resultant word meta-embeddings on three analogy tasks. We find that performance is improved for all word similarity datasets with a mean increase of 11.33 in the Spearman Correlation coefficient when compared to unsupervised learning methods. Before discussing related work, we summarize the main points of our work.

Angular-Based Cost Function Current unsupervised learning methods minimize the Euclidean distance (ℓ_2) between source word embeddings their meta-embedding.

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. We also 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 which corresponds to co-occurrence frequencies. We demonstrated this across multiple benchmark datasets.

Supervision From Manual Annotations

The second point is that both word embeddings and word meta-embedding methods currently do not leverage the available manual annotations in the learning process, such as Spearman correlation scores for word similarity. In particular, word vectors often struggle to preserve true similarity, which in many cases is difficult to identify from statistical associations alone. Hence, we find in previous work (Hill, Reichart, and Korhonen 2016) that has found word embeddings to struggle for word similarity in comparison to word association, particularly for abstract concepts.

Our semi-supervised method addresses this point by learning to reconstruct meta-embeddings while using sharing the hidden layer to also predict word pair similarity. We argue that this explicit use of true similarity scores can greatly improve embeddings for tasks that rely on true similarity. This is reflected in the results as we find for Simlex and RareWord there is a 29.63 and 27.05 increase in Spearman correlation respectively. Improvements on RareWord also brings us to our last point that word meta-embeddings allow for more coverage.

Dealing with Out-of-Vocabulary Words Word vectors suffer in performance for out-of-vocabulary words that are not seen during training. This is often addressed by taking the stem or lemma of the unseen word and using this word representation (if present in the word vector dictionary) as a replacement. We find this is particularly an issue on evaluation datasets that look to gauge performance on words that are morphologically complex words or rare words (Luong, Socher, and Manning 2013), and words that convey more abstract concepts with low concreteness (Hill, Reichart, and Korhonen 2016) (often found to be the case for verbs). In fact (Luong, Socher, and Manning 2013) have used sub-word vectors for such issues. Alternatively, (Cao and Rei 2016) have used a character-level composition from morphemes to word embeddings where morphemes that yield better predictive power for predicting context words are given larger weights, showing improvement over word-based embeddings for syntactic analogy answering. Their model incorporated morphological information into character-level embeddings which in turn, produced better representations for unseen words.

In contrast, word meta-embeddings allow for much larger coverage by combining the ensemble set of pretrained word embeddings, trained from different corpora. Hence, the likelihood of rare words occurring is decreased. This approach also allows for sub-word level combinations between word vectors.

Related Research

Word Meta-Embeddings

The most straightforward approaches to meta-embeddings are: concatenation (CONC) and averaging (AV) (Coates and Bollegala 2018). The former is limited since the dimensionality grows with the number of source embeddings as more vectors are concatenated and the latter does not preserve most of the information encoded in each embedding. Although, it would seem surprising that concatenating vectors from different embedding spaces to be valid, it has been shown by (Coates and Bollegala 2018) if the word vectors are approximately orthogonal, AV approximates CONC even though the embedding spaces may be different. Hence, we include AV in our comparisons of unsupservised methods as one of the baselines.

Although, to address the loss of information when using AV, Singular Value Decomposition (SVD) has been used 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 simply is trained using an ℓ_2-based loss. Similarly, (Bollegala, Hayashi, and ichi Kawarabayashi 2018) have 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 Embedding (DAEME) that keep activations separated for each respective embedding input during encoding and uses a reconstruction loss for both predicted embeddings while minimizing the mean of both predictions, (2) Coupled Autoencoded Meta Embedding (CAEME) instead learn to predict from a shared encoding and drop the expectation minimization loss used in DAEME, and (3) Averaged Autoencoded Meta-Embedding (AAME) is simply an averaging of the embedding set is performed instead of using a concatenated input. This is the most relevant work to ours, hence, we include these 3 autoencoding schemes along with aforementioned methods for experiments, described in Section . 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, where after training 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 .

MTL Representations

Using Multi-Task Learning (MTL) to improve generalization of text representations in natural language tasks has been well-established within the past two decades. Here, we briefly describe some of the related MTL research.

MTL for Named Entity Recognition and Part-Of-Speech Tagging (Ando and Zhang 2005) introduce work for learning predictive representations from multiple tasks in a partially supervised and unsupervised way, which draws similarities to the work presented in this paper. The challenge can be characterized as (1) labels are predicted for an auxiliary task from another task that is trained with full supervision, and (2) both tasks are in some way related. Both characteristics also hold for the work presented in our paper with the subtle difference that we are using many related word similarity datasets with full supervision to predict the auxiliary task (ie a left out word similarity dataset for testing). In their work, context part-of speech (PoS) tags are used to predict the current words PoS tag in an unsupervised fashion, similar to predict-based methods for the current state of the art word embeddings (e.g `skipgram`). This is done by masking some features that are to be predicted. This is carried out while learning on the supervised task of text categorization, allowing the model to choose masks that remove features that are unrelated to the main task. They yielded SoTA performance on PoS tagging and NER using a language model that predicted a target word given context words.

MTL for Neural Machine Translation (Dong et al. 2015) have used MTL to improve the quality of machine translation to multiple target languages. Therefore, source language representations are shared in the encoder-decoder sequence model given the availability of the required parallel data. The model showed higher BLEU scores over independent sequence-to-sequence language models when there is full availability of parallel data and partially available parallel data, highlighting the importance of integrating related source language representations.

MTL Neural networks for Query Classification and Web Search Liu et al. (2015) too use MTL for query classification using multiple binary classifiers, and web search ranking based on maximum likelihood with deep neural networks. Their MTL architecture consisted of 3 shared hidden layers that use character and word n-gram inputs, where the last layer are independent task-specific layers for query classification and web search respectively. The MTL approach showed large improvements over baseline Support Vector Machines and neural networks that learned each task independently.

MTL Benchmarks More recently, we have seen an uptake of MTL for various natural language text similarity tasks. In fact, a benchmark called Understanding Evaluation benchmark (GLUE) has been introduced for this purpose (Wang et al. 2018). This benchmark allows both transfer learning (TL) and MTL models to be developed for sentence classification and learning relationships between sentences (e.g paraphrasing, natural language inference). For the purpose of learning better distributed word representations, in a similar vein we encourage further use of the word similarity datasets for MTL of word meta-embeddings, such as those used for evaluation in this work.

Remarks In contrast, tasks that do benefit from each other and share representations have the advantage of producing better regularization by (1) introducing relevant interdependent features, (2) multiple tasks mutually regularize the model, (3) using future tasks to interpolate to present tasks (less relevant in this work as the dynamics of word meaning do not change greatly of the timeline of dataset creation), (4) improve the model's ability to learn general features from noisy signals and (5) potentially leverages loose structure among the parent tasks that aid more specific downstream child sub-tasks (e.g tasks are designated based on the word relation type such as hyponymy, antonymy or synonymy). These are just a few reasons as to why MTL can improve performance over single-task learners, for a detailed exploration see (Caruana 1998).

Lastly, all of the aforementioned work focus on using MTL on high-level natural language tasks. This work is distinctly different in that it is focusing on improving the input distributed word representations themselves using a semi-supervised MTL approach.

Methodology

Before introducing the semi-supervised MTL approach to learning word meta-representations we first outline the unsupervised learning baselines used for comparison. Firstly, we include both the aforementioned 1TON/1TON^{+} (Yin and Schütze 2015) and standard AEs (Bao and Bollegala 2018) presented in previous work. We also include a slight variant of the AE, which we refer to as a Target Autoencoder (TAE) which learns an ensemble of nonlinear transformations between sets of bases X_s in sets of vector spaces $\mathcal{X}_S = \{\mathcal{X}_1, .., \mathcal{X}_s, .., \mathcal{X}_N\} \quad s.t \quad \mathcal{X}_s \in \mathbb{R}^{|v_s| \times d_s}$ to a target space $\mathcal{X}_t \in \mathcal{R}^{|v_t| \times d_t}$, where $\mathtt{f}_w^{(i)} : \mathcal{X}_S^{(i)} \to \mathcal{X}_t \quad \forall i$ is the nonlinear transformation function used to make the mapping. Once a set of parameteric models $\mathtt{f}_\theta = \{\mathtt{f}_\theta^{(1)}, \mathtt{f}_\theta^{(i)}, .., \mathtt{f}_\theta^{(M)}\}$ are trained with various objective functions $\mathcal{L}_\theta$ to learn the mappings between the word vectors, we obtain a set of lower-dimensional target latent representations that represent different combinations of mappings from one vector space to another.

Figure 1 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 predictions that are subsequently concatenated. 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. The TAE approach is motivated by Caruana (1998) who points out that treating inputs as auxiliary output tasks can, in some cases, be more useful.

AutoEncoder Meta-Embedding

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 $X_i \in \mathbb{R}^{|v_i| \times 200}$ $\forall i$ 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).

$$\mathcal{L}_\theta(\hat{x}, x) = \frac{1}{N} \sum_{i=1}^{N} \left(x^{(i)} - \hat{x}^{(i)}\right)^2 \tag{1}$$

Similarly, the MAE (ℓ_1 norm of difference) loss $1/N \sum_{i=1}^{N} |x - \hat{x}|$ 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 $\mathrm{KL}(p|q) = \sum_{i-1}^{N} p(x_i) \log\left(q(x_i)/p(x_i)\right)$.

$$\mathcal{L}_\theta(\hat{x}, x) = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} \cdot \left(\log\left(x^{(i)}\right) - \log\left(\hat{x}^{(i)}\right)\right) \tag{2}$$

Since $\tanh$ activations are used and input vectors are ℓ_2 normalized we propose a Squared Cosine Proximity (SCP) loss, shown in Equation 3. 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 also 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 metric (i.e symmetric), unlike KL-divergence.

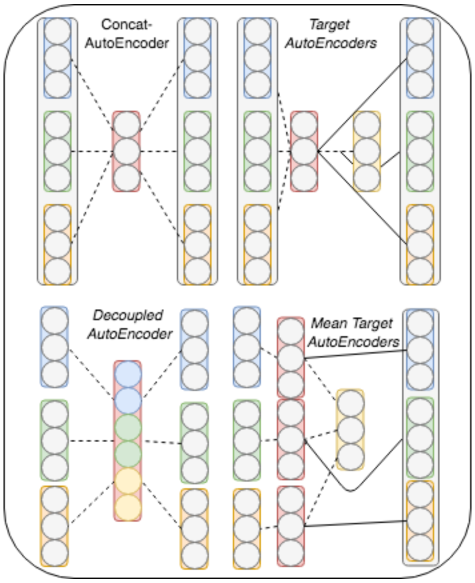

Figure 1: AE Meta-Embedding Methods

$$\mathcal{L}_\theta(\hat{x}, x) = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{\sum_{j=1}^{m} \hat{x}_{ij} \times x_{ij}}{\sqrt{\sum_{j=1}^{m} \hat{x}_{ij}^2}\sqrt{\sum_{j=1}^{m} x_{ij}^2}}\right)^2 \tag{3}$$

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. In experimentation, we found performance obtained via deep networks was not significantly better than a shallow network. Additionally, all comparison are that of models which are trained from co-occurrence statistics that are not leveraging any external knowledge (cf. AutoExtend (Rothe and Schütze 2015)).

Semi-Supervised Multi-Task Meta-Embeddings

MTL for Word Similarity Semi-supervised MTL is also used to learn both the word similarity from human provided annotations and reconstructions of an ensemble of word vectors. This is deemed semi-supervised because both tasks are distinct and both share the hidden layer representation. MTL is particularly apt for improving the overall generalization of ensembled word vectors since the shared representation that is used in standard neural network MTL is also explicitly being used as input word vectors for all other upstream tasks.

The labels y for each dataset d are [0,1] normalized as some of $d \in D$ are in the range [0,10]. The scores are then considered as soft labels where the targets are considered probabilities. We had also considered converting the continuous outputs into binary classes where the threshold was based on the mean of the annotation scores. However, the quartiles of the distribution over the annotation scores are not symmetric around the median, with the exception of MEN and Simlex.

For testing, we train word similarity on all datasets except the one we test on, while an autoencoder is also used to produce reconstructions from word pairs (x_1, x_2) is also used on the test dataset as it is the unsupervised (self-supervised) learning part of the network. This is illustrated in Figure 3, where red coloring indicates the hidden layer representations. For training, the hidden layer dimension sizes of $h^{(1)}$-$h^{(2)}$-$h^{(3)}$ is $200-50-10$. Note that $h^{(1)}$ has dimensionality $d = 200$, the same size as the aforementioned unsupervised learning approach that does not use MTL (ie word similarity is not learned). The example shows inputs words "cup" of "tea", a typical example of where *true* similarity is different from the commonly associated teacup and teabag. Once MTL has converged over a set of epochs, we compare the Spearman correlation ρ_s of the shared hidden layer h^1 outputs, as opposed to using $\hat{y}$ produced in the siamese network that predicts word similarity directly.

We test various distance measures for word similarity, including Manhattan, Euclidean and Cosine dissimilarity. Since the data is 0-1 normalized the pairwise distance $z^l = d_\theta(h_1^l, h_2^l)$, as shown in Equation 4 is kept in this range using the negative exponent $\hat{y} = \exp(-z^l)$. This corresponds to estimating the probability density of the output labels,

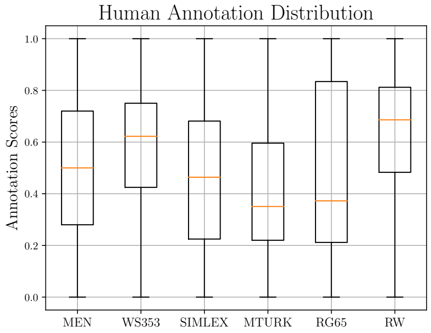

Figure 2: Word Association Annotation Distribution

where y are soft labels. Since each dataset proposes different guidelines for annotation with different annotators, the output distributions $\forall Y \in \mathcal{Y}$ are quite different as shown in Figure 2.

$$\hat{y} = e^{-d_{\omega}(h_1^l, h_2^l)} \tag{4}$$

The total loss $\mathcal{L}$ is simply $\mathcal{L}_r + \mathcal{L}_s$ where the former $\mathcal{L}_r$ in Equation 5 is the reconstruction loss and $\mathcal{L}_s$ shown in Equation 6 the cross-entropy error between both word encodings for the similarity between word pair meta-embeddings guided by the provided annotation scores y.

$$\mathcal{L}_r = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{2}\Big(x_{ij} - \hat{x}_{ij}\Big)^2 \tag{5}$$

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\Big) \tag{6}$$

We argue that when annotators decide on word similarity given a word pair that they choose on how x_1 relates to x_2 only, and not vice-versa (Tversky 1977). In other words, the relationship between two words is not strictly symmetric and the viewing order matters when humans are tasked with estimating word similarity. Therefore, when coming up with a similarity measure using this argument, we test an asymmetric similarity measure between (h_2^l, h_2^l) encodings. This simply involves replacing the denominator of the cosine similarity $(||x_1||_2) \cdot ||x_2||_2)$ that is used as the distance function $d_{\omega}(\cdot)$ to $||x_1||_2 \cdot ||x_1||_2$ before being passed to the negative exponent for the final probability output. Hence, the model is trained to learn how x_1 is related to x_2.

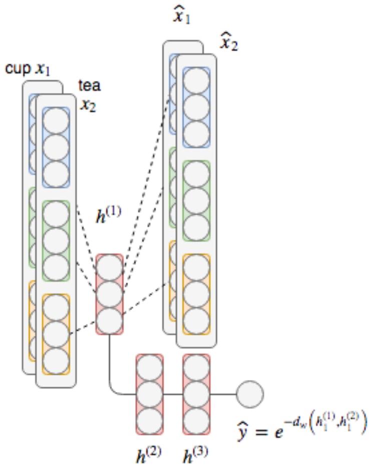

Figure 3: Multi-Task Meta-Embedding

MTL for Analogy We also wish to test our method for analogy, given that it also plays a fundamental role in human cognition (Gentner et al. 2001; Gentner and Forbus 2011). Since not all words in the embedding set are known to preserve analogies as a side effect of training, we would expect that the single embeddings that this quality does hold for would outperform the ensemble. We test if the word meta-embedding encodes analogical structure. Furthermore, we test if the supervision has improved performance in this regard and whether it is transferable to analogy.

Experiments

The word vectors considered in the embeddings set are `skipgram` and `cbow` (Mikolov et al. 2013a), FastText (Bojanowski et al. 2016), LexVec (Salle, Idiart, and Villavicencio 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.

Word Similarity Results

The following word association and word similarity datasets are used throughout experimentation: Simlex (Hill, Reichart, and Korhonen 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).

Table 1 shows the results, where (1) shows the original single embeddings, (2) results for standard meta-embedding approaches that either apply a single mathematical operation or use a linear projection as an encoding, (3) presents the results using autoencoder schemes by Bollegala and Bao (2018) that we have used to test the various losses, and (4) shows the results of concatenating Y with the lower-dimensional (200-dimensions) vector that encodes all embeddings apart from the target vector. Therefore, (4) concatenates the vector leading to a vector between 300-500 dimensions depending on the target vector size. 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, Reichart, and Korhonen 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 against other meta-embedding approaches, showing good results on WS353. However, overall, standard AE performs better. Lastly, we found that when weak embeddings such as HPCA directly attribute to a degradation in the performance of models which use dimensionality reduction techniques or reconstruction methods such as autoencoding.

The autoencoder that uses a squared cosine loss and a KL-divergence loss improves performance in the majority of the cases, reinforcing the argument that accounting for angles explicitly through normalization (log-softmax for KL) is an important step in encoding to account for large documents of varying length and semantics. Lastly, we have shown its use in the context of improving word meta-embeddings, although this suggests cosine loss is also suitable for minimizing angular differences for word embeddings.

Table 1 shows the results of unsupervised learning methods, where grey represents the best model for each section (1-4) and red represents the best model for all sections (also the case for proceeding tables). We find it is clearly difficult to obtain relatively good performance on Simlex and RareWord. The former was introduced to make a clear distinction between association and true similarity, hence the annotation scores reflect this difference, making it difficult for DSMs which only rely solely on word associations.

In contrast, we see in Table 2 there is a large improvement in ρ_s over these datasets using the semi-supervised multi-task learning with asynchronous loss updates (ALU). ALU refers to propagating $\mathcal{L}_r$ and $\mathcal{L}_s$ separately one after the other. In training, the order of the updates is randomly chosen to avoid biasing towards either reconstruction of the word meta-embedding or learning word similarity.

The first measure (e.g Cosine-) represents the reconstruction loss $\mathcal{L}_r$ and second represents the word similarity loss $\mathcal{L}_s$ (e.g NLL). We find that a cosine $\mathcal{L}_r$ and the Brier's score $\mathcal{L}_s$ performs the best on average. Brier's score (Brier 1950) is a scoring function of probabilistic predictions (since we are considering soft labels), where MSE is used in practice. We distinguish from ℓ_2 that is the MSE used for reconstruction of word meta-embeddings. Log-likelihood is often used for fitting models, however when the evaluation measures are different from that of the maximum likelihood (i.e Spearman Correlation) it is not completely clear if this is the best option. Furthermore, our labels in this case are soft labels. Brier's score is often used in classification in such cases which is equivalent to the Mean Squared Error (MSE) class probabilities, in this case binary classification. We find that using Brier's score for the annotations improves the Spearman correlations for 4 out of 6 datasets, as shown in Table 2.

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	70.50	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 +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 Unsupervised Results

Meta-embeddings that are learned only using unsupervised methods (Table 1) give $\rho_s = 45.10$, on Simlex, while the semi-supervised MTL approach produces the most noticeable performance gain with a dramatic increase of $\rho_s = 74.73$. Firstly, this indicates that although Simlex has made a clear distinction between word associations and true similarity in its annotation scheme, there is still value in predicting such scores from different annotation distributions that only score word association and not the true similarity that is the focus of Simlex (Hill, Reichart, and Korhonen 2016).

In the semi-supervised MTL setting shown in Table 2, we see that results are also consistent with Table 1 as the cosine loss in reconstruction results in best performance for 4 out of the 6 datasets.

Analogy Results

We evaluate how the learned models from Table 2 transfer to analogy tasks, namely MSR Word Representation dataset (Gao, Bian, and Liu 2014) (8000 questions with 8 relations), Google Analogy dataset (Mikolov et al. 2013b) (19,544 questions with 15 relations) amd SemEval 2012 Task 2 Competition Dataset (Jurgens et al. 2012) (3218

	Simlex	WS353	RG	MTurk	RW	MEN
Cosine-OLS	53.63	73.13	83.07	69.41	60.49	80.25
Cosine-NLL	59.22	76.09	80.45	70.43	61.31	82.49
Cosine-Brier's	63.72	80.21	89.54	83.45	70.76	84.14
ℓ_1-OLS	55.16	68.80	82.82	70.35	61.07	78.56
ℓ_1-NLL	53.54	77.82	82.09	73.12	64.46	79.12
ℓ_1-Brier's	68.78	77.60	87.44	80.67	78.05	79.73
ℓ_2-OLS	68.31	73.85	84.48	70.91	53.20	81.60
ℓ_2-NLL	53.80	71.15	85.10	71.51	50.61	79.38
ℓ_2-Brier's	74.73	69.68	85.29	76.30	80.07	70.64
KL-OLS	62.47	68.93	85.75	72.35	50.38	80.95
KL-NLL	48.91	67.93	86.67	72.33	48.91	78.98
KL-Brier's	71.39	66.91	87.58	73.43	67.11	81.78

Table 2: Semi-Supervised Multi-Task Word Embedding Learning Results on Word Similarity

question with 79 relations). The former two consist of categories of different analogy questions and the latter includes ranked candidate word pairs based on word pair relational similarity for a set of chosen word pairs. CosAdd (Mikolov, Yih, and Zweig 2013) is used for calculating the analogy answers for Google and MSR which ranks candidates given as CosAdd$(a : b, c : d) = \cos(b - a + c, d)$ and chooses the answer as the highest ranking candidate. For SemEval 2012 task 2, the word pair relations are manually assigned categories that are also assigned a class membership score which represents how well the pair represent a class. Therefore, the scores provided are word relation similarity scores, hence the spearman correlation is used for this dataset when evaluating the models.

Table 3 shows the results of transferring the learned semi-supervised multi-task learning (SS-MTL) embeddings to analogy tasks. Here, we analyze (1) whether the word meta-embeddings carry over to analogy even if not all embedding algorithms preserve analogy relations and (2) check if the similarity encoded with SS-MTL has any effect on performance on analogy. We find that in general, semi-supervised MTL that incorporates similarity scores has some transferability to analogy, at least based on the scores provided by the aforementioned word similarity datasets. Using word similarity scores for supervision is a general measure of similarity, whereas analogy relations are more specific, hence it is not surprising that the difference in performance is slight.

However, for Google Analogy, the larger of the three datasets with the smallest range of relation types, we find that the SS-MTL model that previously trained with Cosine-Brier's loss functions shows the best performance overall. This is consistent with findings from Table 2 where the same model performs best over 4 of 6 word similarity datasets. This suggests that performing additional nonlinear meta-word encoding somewhat preserves the linear structures preserved in models such as skipgram and fasttext. Additionally, it remains clear that Brier's score (i.e ℓ_2 for classification) is best suited, at least when evaluating with Spearman correlation.

	MSR	Google	SemEval
Skipgram	73.13	72.89	22.64
FastText	64.19	73.82	24.77
GloVe	71.45	71.73	19.98
LexVec	74.03	67.28	21.49
Cosine-OLS	73.24	71.57	22.13
Cosine-NLL	71.23	68.39	20.16
Cosine-Brier's	74.78	74.18	23.44
ℓ_1-OLS	69.32	68.21	20.45
ℓ_1-NLL	68.69	67.27	19.02
ℓ_1-Brier's	70.37	72.55	20.36
ℓ_2-OLS	73.20	72.16	22.71
ℓ_2-NLL	72.37	69.35	21.08
ℓ_2-Brier's	75.72	74.11	24.84
KL-OLS	68.08	65.28	18.24
KL-NLL	65.51	65.90	19.66
KL-Brier's	64.30	67.22	20.75

Table 3: SS-MTL Embedding Transferability To Analogy

Conclusion

This paper introduced a semi-supervised learning method for improving word meta-embeddings by reconstructing an ensemble of word vectors while also learning to predict word similarity whereby the hidden layer representation is shared between both tasks. We find performance increased significantly when using manually annotated scores from word similarity datasets in comparison to single word embeddings and unsupervised word meta-embedding approaches. We also find that angular-based loss functions are well suited for word meta-learning for both unsupervised learning and the proposed multi-task semi-supervised learning method, showing best results on 4 out of the 6 word similarity datasets in both cases. In particular, we find most significant improvements on relatively difficult word similarity and association datasets such as Simlex and RareWord, while still improving by a large margin on the remaining datasets. Finally, we see slight improvements made when transferring the semi-supervised models for analogy tasks. However, this is expected given that similarity scores are more general than specific word pair relation types and not all word embedding algorithms preserve analogical relations to the same degree.

References

[Ando and Zhang 2005] Ando, R. K., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6(Nov):1817–1853.

[Bao and Bollegala 2018] Bao, C., and Bollegala, D. 2018. Learning word meta-embeddings by autoencoding. *COLING*.

[Bojanowski et al. 2016] Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

[Bollegala, Hayashi, and ichi Kawarabayashi 2018]

Bollegala, D.; Hayashi, K.; and ichi Kawarabayashi, K. 2018. Think globally, embed locally — locally linear meta-embedding of words. In *Proc. of IJCAI-EACI*.

[Brier 1950] Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthey Weather Review* 78(1):1–3.

[Bruni et al. 2012] Bruni, E.; Boleda, G.; Baroni, M.; and Tran, N.-K. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 136–145. Association for Computational Linguistics.

[Cao and Rei 2016] Cao, K., and Rei, M. 2016. A joint model for word embedding and word morphology. *arXiv preprint arXiv:1606.02601*.

[Caruana 1998] Caruana, R. 1998. Multitask learning. In *Learning to learn*. Springer. 95–133.

[Coates and Bollegala 2018] Coates, J., and Bollegala, D. 2018. Frustratingly easy meta-embedding–computing meta-embeddings by averaging source word embeddings. *arXiv preprint arXiv:1804.05262*.

[Dong et al. 2015] Dong, D.; Wu, H.; He, W.; Yu, D.; and Wang, H. 2015. Multi-task learning for multiple language translation. 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, 1723–1732.

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

[Gao, Bian, and Liu 2014] Gao, B.; Bian, J.; and Liu, T.-Y. 2014. Wordrep: A benchmark for research on learning word representations. *arXiv preprint arXiv:1407.1640*.

[Gentner and Forbus 2011] Gentner, D., and Forbus, K. D. 2011. Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science* 2(3):266–276.

[Gentner et al. 2001] Gentner, D.; Holyoak, K. J.; Holyoak, K. J.; and Kokinov, B. N. 2001. *The analogical mind: Perspectives from cognitive science*. MIT press.

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

[Hill, Reichart, and Korhonen 2015] Hill, F.; Reichart, R.; and Korhonen, A. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* 41(4):665–695.

[Hill, Reichart, and Korhonen 2016] Hill, F.; Reichart, R.; and Korhonen, A. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

[Jozefowicz et al. 2016] Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; and Wu, Y. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

[Jurgens et al. 2012] Jurgens, D. A.; Turney, P. D.; Mohammad, S. M.; and Holyoak, K. J. 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*, 356–364. Association for Computational Linguistics.

[Lebret and Collobert 2013] Lebret, R., and Collobert, R. 2013. Word emdeddings through hellinger pca. *arXiv preprint arXiv:1312.5542*.

[Liu et al. 2015] Liu, X.; Gao, J.; He, X.; Deng, L.; Duh, K.; and Wang, Y.-Y. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL*.

[Luong et al. 2014] Luong, M.-T.; Sutskever, I.; Le, Q. V.; Vinyals, O.; and Zaremba, W. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

[Luong, Socher, and Manning 2013] Luong, T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, 104–113.

[Mikolov et al. 2013a] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al. 2013b] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

[Mikolov, Yih, and Zweig 2013] Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, 746–751.

[Rothe and Schütze 2015] Rothe, S., and Schütze, H. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*.

[Rubenstein and Goodenough 1965] Rubenstein, H., and Goodenough, J. B. 1965. Contextual correlates of synonymy. *Communications of the ACM* 8(10):627–633.

[Salle, Idiart, and Villavicencio 2016] Salle, A.; Idiart, M.; and Villavicencio, A. 2016. Matrix factorization using window sampling and negative sampling for improved word representations. *arXiv preprint arXiv:1606.00819*.

[Sun et al. 2015] Sun, F.; Guo, J.; Lan, Y.; Xu, J.; and Cheng, X. 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, 136–145.

[Tversky 1977] Tversky, A. 1977. Features of similarity. *Psychological review* 84(4):327.

[Wang et al. 2017] Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings* *of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 189–198.

[Wang et al. 2018] Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

[Yin and Schütze 2015] Yin, W., and Schütze, H. 2015. Learning meta-embeddings by using ensembles of embedding sets. *arXiv preprint arXiv:1508.04257*.