Learning Efficient Task-Specific Meta-Embeddings with Word Prisms

Jingyi He[*1], **KC Tsiolis**[*1], **Kian Kenyon-Dean**[†*2], **Jackie C. K. Cheung**[1]
Mila – Québec AI Institute / McGill University, Montréal, QC, Canada [1]
BMO AI Capabilities Team – Bank of Montréal, Toronto, ON, Canada [2]
{jingyi.he,kc.tsiolis}@mail.mcgill.ca, jcheung@cs.mcgill.ca
kian.kenyon-dean@bmo.com

Abstract

Word embeddings are trained to predict word cooccurrence statistics, which leads them to possess different lexical properties (syntactic, semantic, etc.) depending on the notion of context defined at training time. These properties manifest when querying the embedding space for the most similar vectors, and when used at the input layer of deep neural networks trained to solve downstream NLP problems. Meta-embeddings combine multiple sets of differently trained word embeddings, and have been shown to successfully improve intrinsic and extrinsic performance over equivalent models which use just one set of source embeddings. We introduce *word prisms*: a simple and efficient meta-embedding method that learns to combine source embeddings according to the task at hand. Word prisms learn orthogonal transformations to linearly combine the input source embeddings, which allows them to be very efficient at inference time. We evaluate word prisms in comparison to other meta-embedding methods on six extrinsic evaluations and observe that word prisms offer improvements in performance on all tasks.[1]

1 Introduction

A popular approach to representing word meaning in NLP is to characterize a word by "the company that it keeps" (Firth, 1957). This intuition is the basis of famous word embedding techniques such as Word2vec (Mikolov et al., 2013a) and Glove (Pennington et al., 2014). However, the question of *what company a word keeps* — i.e., what should define a word's context — is open. A word's context could be defined via a symmetric window of 1, 2, 5, 10, 20 words, the words that precede it, the words that follow it, the words with which it shares a dependency edge, etc. Determining the utility of such different notions of context for training word embeddings is a problem that has attracted considerable attention (Yatbaz et al., 2012; Levy and Goldberg, 2014a; Bansal et al., 2014; Lin et al., 2015; Melamud et al., 2016; Lison and Kutuzov, 2017) but there is no conclusive evidence that any single notion of context could be the best for solving NLP problems in general. Thus, many deep learning solutions for NLP have yet another hyperparameter to tune: what set of word embeddings should be selected for the input layer of the model. As NLP tasks become more and more complex, the practice of providing a deep model with only one notion of a word's meaning becomes limiting.

Word meta-embeddings address aspects of this problem by proposing techniques for combining multiple sets of word embeddings before providing them into the input layer of a downstream model. Yin and Schütze (2016) motivated word meta-embeddings by arguing that they are advantageous for the following reasons: *diversity* — combining embeddings trained with different algorithms on different corpora will allow for more distinct meanings of the words to persist; and, *coverage* — combining embeddings trained on different corpora help to better solve the out-of-vocabulary problem. However, they did not acknowledge that different sets of word embeddings can be diverse even when trained on the same corpus with the same algorithm, so long as their context windows are different. Additionally, due to the various practical and theoretical similarities between different algorithms (Levy and Goldberg, 2014b;

[*] Equal contribution. † This work was pursued prior to Kian's employment at BMO.

[1] https://github.com/kylie-box/word_prisms

Proceedings of the 28th International Conference on Computational Linguistics, pages 1229–1241
Barcelona, Spain (Online), December 8-13, 2020

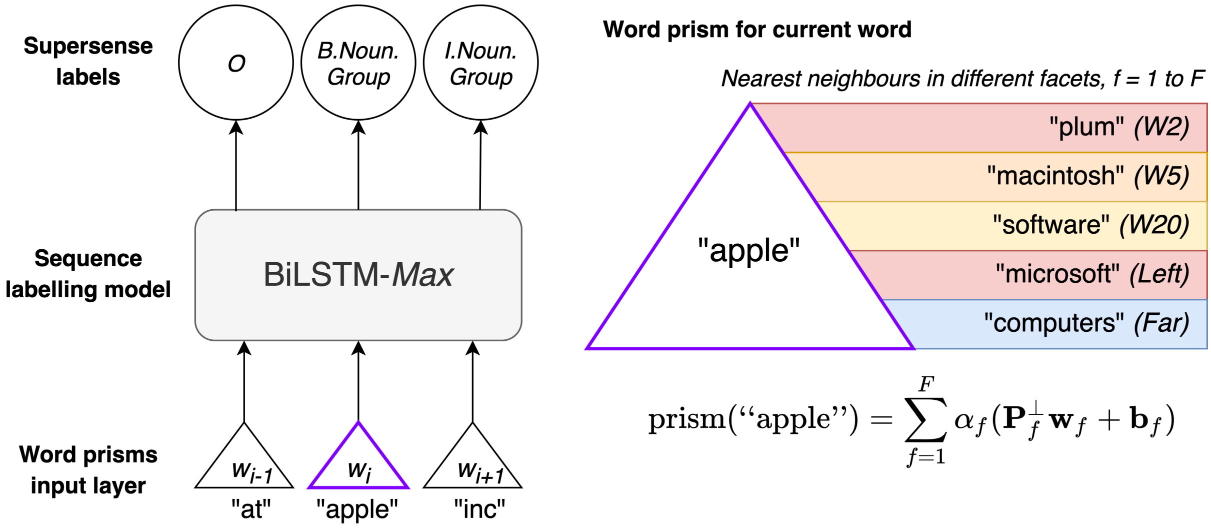

Figure 1: Word prisms (§3) during supersense tagging (§4). We display the nearest neighbor to the embedding for the word "apple" in five of the window-based facets (Figure 2, §3.2) used in the downstream model (§4). We include the ⊥ symbol on the learned transformation, $\mathbf{P}_f$, to indicate its orthogonality.

Levy et al., 2015; Newell et al., 2019), the gains to be found in diversifying at the level of algorithmic variation are likely to be minimal. With regard to vocabulary coverage, the out-of-vocabulary problem is at least partially addressed by character n-gram based embedding algorithms such as FastText (Joulin et al., 2017) and subword-based decomposition techniques that can be applied post-training (Zhao et al., 2018; Sasaki et al., 2019). Nonetheless, meta-embeddings have been shown to consistently outperform models that use only a single set of embeddings in their input layer. Our goal is to determine how to best combine many sets of input embeddings in order to obtain high quality results in downstream tasks.

This work proposes *word prisms* as a simple and general way to produce and understand meta-embeddings, visualized in Figure 1. Word prisms excel at combining many sets of source embeddings, which we call *facets*. They do so by learning task-specific orthogonal transformations to map embeddings from their facets to the common meta-embedding space. This produces a vector space that is more disentangled than the original space of facet embeddings. It allows the combination of multiple source embeddings while preserving most information within each embedding set.

To our knowledge, this work is the first to incorporate both explicit orthogonal transformations of source embeddings and importance weights for source embedding sets that are dynamically learned with the downstream tasks in the same meta-embedding method. Furthermore, it is the first to explore combining so many sets of source embeddings (thirteen). We compare the word prisms method to other standard meta-embedding algorithms (averaging (Coates and Bollegala, 2018), concatenation (Yin and Schütze, 2016), and dynamic meta-embeddings, DMEs, (Kiela et al., 2018)). Word prisms overcome the shortcomings of each of these algorithms: (1) in averaging, performance deteriorates considerably when there are many facets — the orthogonal transformations in word prisms resolve this problem; (2) concatenation and DMEs are too expensive during inference when there are many facets — word prisms only need the final meta-embeddings at inference time, making them as efficient as averaging. Our results demonstrate that neural downstream models using word prisms generally obtain better results than the other algorithms across six downstream tasks, including supersense tagging, POS tagging, named entity recognition, natural language inference, and sentiment analysis. In our ablation studies, we find that our method improves performance on downstream tasks even when the vocabulary is the same across nine source facets trained on the same corpus that differ only by the definition of context window (see Figure 2); performance further improves by incorporating four more sets of off-the-shelf facets.

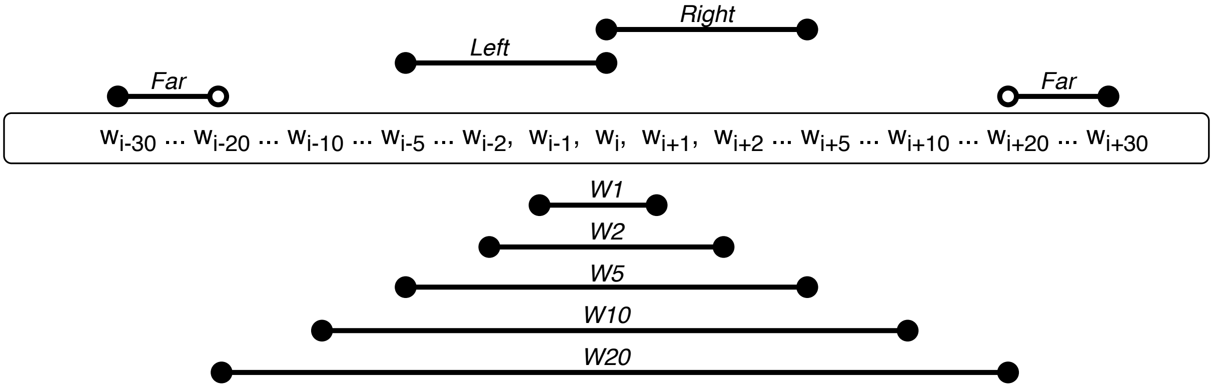
Right
Left
Far
Far
w_{i-30} ··· w_{i-20} ··· w_{i-10} ··· w_{i-5} ··· w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2} ··· w_{i+5} ··· w_{i+10} ··· w_{i+20} ··· w_{i+30}
W1
W2
W5
W10
W20

Figure 2: The eight different context windows with which we trained word embedding facets (note that the dependency-based facet is not included here). Word w_i represents the center of the context windows.

2 Related Work

Pre-trained word embedding algorithms use word cooccurrence statistics from a training corpus to map words to a low-dimensional vector space such that words with similar meanings are mapped to similar points in vector space. However, changing the embedding algorithm, training corpus, or definition of cooccurrence can have a strong impact on the resulting embeddings. Consequently, word *meta-embedding* algorithms have been developed to combine multiple embedding sets.

2.1 Word embedding training and notions of context

Word embeddings are trained to reflect the cooccurrence statistics in the input corpus, which depend on the specific definition of context being employed. The standard definition in Word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) is a symmetric context window of fixed size around each word in the corpus. Levy and Goldberg (2014a) explore dependency-based contexts, where the context of each word is defined as its governor and dependents, along with the corresponding dependency relation labels. They observe that embeddings trained on smaller context windows and dependency-based contexts relate words that can be substituted for one another. By contrast, embeddings trained on larger context windows relate words which address the same topic. Bansal et al. (2014) observe the same phenomenon. Lin et al. (2015) find that small window sizes work best for POS tagging. Lison and Kutuzov (2017) evaluate word similarity and word analogy task performance for SGNS embeddings trained on different window sizes, as well as left-sided and right-sided context windows.

2.2 Word meta-embeddings

Previous work has shown that combining embedding sets can lead to improvements in downstream performance. Melamud et al. (2016) combine embeddings trained with different notions of context via concatenation, as well as via SVD and CCA, leading to improved performance in multiple downstream tasks. However, they only combine two embedding sets at a time. Yin and Schütze (2016) introduce the term "meta-embeddings" and demonstrate that concatenation and singular value decomposition (SVD) are solid baselines on word similarity, analogy, and POS tagging tasks. They propose 1TON, which simultaneously learns meta-embeddings and projections from the meta-embedding space to each individual source embedding space. Ghannay et al. (2016) apply PCA and autoencoders after concatenating source embeddings. Zhang et al. (2016) apply a convolutional layer to each source embedding before concatenating the resulting feature maps. Bollegala et al. (2018) represent the meta-embedding for a word as a linear combination of the meta-embeddings for its nearest neighbours in each source embedding set. Bao and Bollegala (2018) produce meta-embeddings by either averaging or concatenating the outputs of encoders which take GloVe and CBOW (Mikolov et al., 2013a) embeddings as input.

Coates and Bollegala (2018) demonstrate that, in certain settings, meta-embeddings produced by averaging can be as performant as concatenated ones. Kiela et al. (2018) propose dynamic meta-embeddings

(DMEs), which perform attention over the linearly transformed source embeddings. The linear transformations applied to the source embeddings are not constrained to be orthogonal. Their model learns which source embedding sets are most useful for a particular downstream task and for particular classes of words. They also present a contextualized version, where attention weights also depend on a word's surrounding context, but this provides little to no improvement on their downstream evaluations. By contrast, we propose a simpler attention mechanism by learning a single importance weight for each source embedding set, and we apply orthogonal transformations to source embeddings prior to linear combination. We also experiment with a larger selection of source embeddings, including embeddings trained with different notions of context.

Orthogonal transformations have previously been employed in the context of mapping monolingual embeddings for different languages into a common space (Artetxe et al., 2016; Smith et al., 2017; Artetxe et al., 2018; Conneau et al., 2018; Doval et al., 2018). With these alignment transformations on monolingual space, one can obtain a better cross-lingual integration of the vector spaces. Recent work has also found that applying orthogonal transformations to source embeddings facilitates averaging (García et al., 2020; Jawanpuria et al., 2020). We expand on this work by incorporating orthogonal transformations in word prisms, which learn word meta-embeddings for specific downstream tasks. Additionally, we provide an analysis of source embeddings before and after orthogonal transformation, which leads to the insight that these mappings cause source embedding sets to be more easily clusterable within the meta-embedding space.

3 Word Prisms and Meta-Embeddings

In this section we introduce meta-embeddings, word prisms, and the source embeddings they are composed with in this work. A meta-embedding combines pre-trained embeddings from multiple sources (e.g., from Glove and FastText), which we call *facets*. We define $\mathbf{w}_f$ as the embedding of word w in facet f, for each facet $f \in \{1, ..., F\}$. The dimensionality of each embedding in a facet is denoted d_f, and the final dimensionality of the meta-embedding is d'. The following equation represents the general form of all meta-embedding variants and baselines considered in this work:

$$\mathrm{meta}(w) = \sum_{f=1}^{F} \alpha_f(\mathbf{P}_f \mathbf{w}_f + \mathbf{b}_f) \qquad \text{s.t. } \alpha_f \in \mathbb{R},\, \mathbf{w}_f \in \mathbb{R}^{d_f},\, \mathbf{P}_f \in \mathbb{R}^{d' \times d_f},\, \mathbf{b}_f \in \mathbb{R}^{d'}. \tag{1}$$

That is, the meta-embedding for a word w, $\mathrm{meta}(w) \in \mathbb{R}^{d'}$ is constructed as follows: first, it is projected by a linear transformation (learned or fixed) characterized by a matrix $\mathbf{P}_f$ and bias $\mathbf{b}_f$ for some set of embedding facets. Next, the meta-embedding is a linear combination of the transformed embeddings scaled by some (learned or fixed) weights, α_f. The vocabulary of a word prism is the union of the vocabularies of its facets. If a word is out-of-vocabulary for a facet f, we assign its representation $\mathbf{w}_f$ to be the centroid of the embeddings in the facet.

3.1 Word prisms

Word prisms learn orthogonal transformation matrices $\mathbf{P}_f$ and bias vectors $\mathbf{b}_f$ via back-propagation to make the space of input facets more well-separated so that the downstream model can learn which lexical qualities in the facets are most appropriate for the given task at hand. It is desirable to impose an orthogonality constraint on the transformation matrix because orthogonal matrices preserve the dot products within the original vector space, which has been shown to be important in studies of multilingual embeddings (Artetxe et al., 2016). This requires $\mathbf{P}_f$ to be square ($\mathbf{P}_f \in \mathbb{R}^{d' \times d'}$) which further requires the dimensionality of the facets to all be the same (i.e., each $d_f = d'$, without loss of generality[2]). After each gradient descent update, we apply the following update rule used by Cisse et al. (2017) and Conneau et al. (2018), which approximates a procedure that keeps each $\mathbf{P_f}$ on the manifold of orthogonal matrices:

$$\mathbf{P_f} \leftarrow (1+\beta)\mathbf{P_f} - \beta \left(\mathbf{P_f}\mathbf{P_f}^T\right)\mathbf{P_f}. \tag{2}$$

[2] When dimensions are different, simple strategies can be pursued to equalize them. For example, zero-padding short embeddings (Coates and Bollegala, 2018) or using SVD to compress long embeddings are reliable strategies.

The orthogonal transformation keeps the L2 norm of the original embeddings the same, since it does not rescale the vectors. Our preliminary experiments found $\beta = 0.001$ is a good option.

Word prisms also learn a linear combination of the projected facets to further adapt to the task at hand. Indeed, it is necessary to learn the α_f separately from $\mathbf{P}_f$ since the transformations are orthogonal and cannot perform rescaling independently. Word prisms learn the facet-level weight coefficients, α_f, directly, where each α_f is a floating-point number also learned via back-propagation from the downstream task signal, initialized to be $1/F$ for each facet. That is, the parameters in word prisms are learned simultaneously with the downstream model for a given task. This approach is advantageous because it allows the model to assign importance weights to each facet, but it is not bound to do so via a dynamic attention vector. So, for word prisms, all of the meta-embeddings can be pre-computed for a vocabulary after training. This means that a word prism model in an inference-only production environment benefits from low memory complexity, as it does not need to hold all of the original facets in memory, only the meta-embeddings. Thus, given a vocabulary size V, and number of facets F, the memory complexity during inference is only $O(V)$ for word prisms (and the average baseline), but is $O(VF)$ for DMEs (and the concatenation baseline).

3.2 Facets

We include 13 various facets into word prisms with the aim of capturing a wide variety of semantic and syntactic information. To our knowledge, this is the first work on meta-embeddings to explore combining so many sets of source embeddings. Our collection of facets is diverse in the following two ways: (1) it incorporates many notions of context using the same algorithm and the same corpus; (2) it incorporates off-the-shelf embeddings trained on much larger corpora and tuned to knowledge graphs.

We make use of nine different notions of context to train standard PMI-based word embeddings (Levy and Goldberg, 2014b; Newell et al., 2019), each with a dimension of 300 and vocabulary size of 500,000. Training is done with the open-source sampling-based implementation of Hilbert-MLE[3] (Newell et al., 2019), which facilitates the use of arbitrarily structured context windows for training. For eight of the nine notions of context, embeddings are trained on the Gigaword 3 corpus (Graff et al., 2007) combined with a Wikipedia 2018 dump, which amounts to approximately 6 billion tokens.

We visualize the different window settings for these eight embedding sets in Figure 2. Letting *W* be the window size, we trained the following sets of embeddings: *W1*, *W2*, *W5*, *W10*, and *W20*. Furthermore, we trained embeddings using only a *Left* context of 5 words, and another set of embeddings with only a *Right* context of 5 words. Lastly, we trained a set of embeddings with only a *Far* context window, which only includes words between 20 and 30 words away, in order to create strong topic-based representations. We also trained a variant of dependency-based embeddings (*Deps*) (Levy and Goldberg, 2014a), where we defined a word's context to be its governor. We ran the CoreNLP (Manning et al., 2014) dependency parser on Gigaword 3 to obtain a parsed corpus.

We also experiment with the following off-the-shelf embeddings: **GloVe** (Pennington et al., 2014); trained on 840B tokens from the Common Crawl Corpus with 2.2M words in the vocabulary. **FastText** (Joulin et al., 2017); trained on 600B tokens from the Common Crawl Corpus with 2M words in the vocabulary. **ConceptNet Numberbatch** (Speer et al., 2017); retrofitted (Faruqui et al., 2015) on both Word2vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014) with 516K words in the vocabulary; this facet allows us to incorporate information from knowledge graphs. **LexSub** (Arora et al., 2020); GloVe embeddings trained on 6B tokens from Wikipedia 2014 and the Gigaword 5 corpus (Parker et al., 2011), modified so that they can easily be projected into "lexical subspaces", in which a word's nearest neighbours reflect a particular lexical relation (e.g. synonymy, antonymy, hypernymy, meronymy).

4 Experiments

Our experiments seek to determine if: (1) word prisms offer improvements over the other common meta-embedding methods; and, (2) if it is desirable to produce meta-embeddings with many different notions of context from the same corpus. For (1), we pursue a variety of experiments comparing word

[3] `https://github.com/enewe101/hilbert`

prisms to the following meta-embedding methods: the averaging baseline, the concatenation baseline, and dynamic meta-embeddings (DMEs) (Kiela et al., 2018). For (2), we experiment with several sets of meta-embedding facet combinations. The first set is FastText and Glove (**FG**), as is done by Kiela et al. (2018). The second set is a combination of 13 different facets (**All**), as detailed in §3.2. In §5 we present our main results, and in §6 we present an ablation study to determine the impact of other meta-embedding combinations and the transformation matrices in word prisms.

4.1 Baselines

We will compare word prisms with three baseline algorithms. The first two, averaging and concatenation, are standard meta-embedding methods often explored in studies on meta-embeddings (Yin and Schütze, 2016; Coates and Bollegala, 2018). The third is dynamic meta-embeddings (DMEs) (Kiela et al., 2018).

Average baseline. Averaging word embeddings is the simplest method to create meta-embeddings. Assuming each facet dimension d_f is equal to d' (as with word prisms), this baseline corresponds to Equation 1 with the following fixed parameter settings: $\alpha_f = \frac{1}{F}$, $\mathbf{P}_f = \mathbf{I}_{d'}$, and $\mathbf{b}_f = \vec{0}$.

Averaging is a sensible strategy to combine multiple source word embeddings, since, first of all, it aggregates the information from all the input facets without introducing additional parameters. Second, it captures semantic information by preserving the relative word distances within the embedding spaces (Coates and Bollegala, 2018). However, as we demonstrate later (§5), the quality of this baseline deteriorates when there are many different facets as the signals start to become too mixed.

Concatenation baseline. Concatenating multiple source embeddings is another trivial way to construct meta-embeddings. The parameter settings here correspond to Equation 1 when $d' = \sum_{f=1}^{F} d_f$, $\alpha_f = 1$, $\mathbf{b}_f = \vec{0}$, and $\mathbf{P}_f$ is a fixed selector matrix that places embeddings into their corresponding concatenated positions; more simply, $\mathrm{meta}(w) = [\mathbf{w}_1, \ldots, \mathbf{w}_F]$.

Concatenation can be desirable because it maintains all of the structure of the original embeddings. However, it is problematic because the dimensionality increases linearly with respect to the number of facets, requiring more model parameters to be learned at the input layer for downstream model.

Dynamic meta-embeddings. Kiela et al. (2018) introduced DMEs, demonstrating that sentence representations can be improved by combining multiple source embeddings with dynamically learned linear transformations and attention weights. Note that, like with concatenation, it is necessary to maintain all of the individual facets in memory during inference when using DMEs.

DMEs are encapsulated in Equation 1 via the following parameter settings: d' is a hyperparameter for the desired meta-embedding size (set to 256 by Kiela et al. (2018)); $\mathbf{P}_f$ and $\mathbf{b}_f$ are learned via backpropagation from supervised learning during the current task; the α_f are obtained via a self-attention mechanism on an additional learned parameter vector $\mathbf{a} \in \mathbb{R}^{d'}$: $\alpha_f = \phi(\mathbf{a} \cdot (\mathbf{P}_f \mathbf{w}_f + \mathbf{b}_f) + b)$, where ϕ is the softmax function and $b \in \mathbb{R}$ is an additional learned bias parameter.

4.2 Datasets and downstream models

We evaluate meta-embedding methods on a variety of downstream text classification and sequence labelling tasks. For text classification, we choose the Stanford Sentiment Treebank binary sentiment analysis dataset (**SST2**) (Socher et al., 2013) and the Stanford NLI (Bowman et al., 2015) (**SNLI**) benchmark. For sequence labelling, we select the CoNLL 2003 named entity recognition task (**NER**) (Tjong Kim Sang and De Meulder, 2003), POS tagging on the Brown corpus (**Brown**)[4], POS tagging on the WSJ corpus (**WSJ**) (Marcus et al., 1993), and Supersense tagging (Ciaramita and Johnson, 2003) on the Semcor 3.0 corpus (**Semcor**) (Miller et al., 1993). Supersense tagging is a problem situated between NER and word sense disambiguation. The task consists of 41 lexicographer class labels for nouns and verbs with IOB tags, producing 83 fine-grained classes in total. We report the micro F1 score for the supersense tagging and the NER tagging tasks, discarding the O-tags in the predictions, as is standard (Alonso and Plank, 2017; Changpinyo et al., 2018). For the rest of the tasks, we report the accuracy on

[4] Retrieved from the NLTK toolkit: `http://www.nltk.org/nltk_data/`.

Model	Facets	Semcor	WSJ	Brown	NER	SNLI	SST2
Average	FG	69.42 ± .1	96.76 ± .02	98.44 ± .02	90.16 ± .2	85.33 ± .3	87.76 ± .5
Concat	FG	72.23 ± .2	96.85 ± .04	98.53 ± .02	90.49 ± .1	85.45 ± .3	**88.57** ± .3
DME	FG	72.15 ± .2	96.81 ± .03	98.53 ± .02	89.49 ± .2	85.57 ± .3	88.10 ± .6
Prism	FG	73.51 ± .1	96.91 ± .01	98.58 ± .02	90.70 ± .4	**85.82** ± .1	87.80 ± .6
Average	All	65.34 ± .3	96.63 ± .01	98.21 ± .03	88.92 ± .3	83.91 ± .1	86.03 ± .9
Concat	All	**73.95** ± .1	97.02 ± .01	98.63 ± .01	90.55 ± .1	84.03 ± .2	88.15 ± .2
DME	All	72.09 ± .1	96.89 ± .01	98.58 ± .02	89.36 ± .3	85.47 ± .1	87.63 ± .6
Prism	All	73.82 ± .2	**97.04** ± .01	**98.65** ± .01	**90.74** ± .2	85.71 ± .3	88.45 ± .5

Table 1: Test set results for word prisms and baseline meta-embedding algorithms (concatenation, averaging, and DMEs) on different combinations of input facets (§3.2) — **FG** is FastText and Glove only, **All** is all 13 facets. We report the mean and standard deviation from runs with five different random seeds. Best result is **bold**, second best is underlined.

the test set. We use the standard train-validation-test splits whenever they are provided with the dataset. Otherwise, we split 10% of the training set to be the validation set for hyperparameter tuning.

We use a simple neural model to compare meta-embedding methods. To replicate the models used by Kiela et al. (2018), we use a single layer BiLSTM with 512 hidden units in SST2 and 1024 hidden units in SNLI for sentence encoding. The sentence representations are learned by max pooling over the forward and backward hidden states. We use the following representation for a pair of hypothesis and premise: $[\mathbf{u}, \mathbf{v}, \mathbf{u} * \mathbf{v}, |\mathbf{u} \text{ - } \mathbf{v}|]$. The $*$ operator denotes the element-wise multiplication. The sentence encoder is followed by a 512 dimensional MLP with ReLU activation. For all the sequence labelling tasks, we use a 2-layer BiLSTM with 256 hidden units for sequence encoding.

We use standard cross entropy loss for all supervised downstream tasks. The parameters of the downstream model and word prisms are learned via back-propagation. We choose the initial learning rate to be 0.001 for sequence labeling tasks and 0.0004 for text classification tasks with a reduction factor of 0.1 if there is no improvement on the validation set after 2 consecutive epochs. In all of our experiments, we keep the source embeddings in their original forms without performing normalization. Counterintuitively, normalizing the source embeddings to have unit norm makes little difference in the text classification tasks but substantially hurts the performance of the sequence labelling tasks. Previous work (Schakel and Wilson, 2015) shows the length of the embedding vector encodes the unigram frequency of the word, which is useful in the sequence labelling tasks.

5 Results

Table 1 presents the main results for this work on four sequence labelling tasks and two text classification tasks. We compare word prisms to standard meta-embedding baselines (averaging, concatenation) and dynamic meta-embeddings (DME). In the first four rows, we experiment with FastText and Glove (**FG**) as a two-facet combination, while the next four rows use a 13-facet combination (**All**) detailed in §3.2.

Our first finding is that word prisms almost always offer substantial improvements over DMEs, regardless of whether we are using two facets or thirteen. The only exception is in the case of text classification on SST2 with **FG**, although the difference is within the margin of error (0.6). Note that Kiela et al. (2018) report slightly different results than our reimplementation of their system, for **FG**: 86.2 ± .2 for SNLI (compared to 85.57 ± .3), and 88.7 ± .6 for SST2 (compared to 88.10 ± .6). These discrepancies are attributable to random initializations and the different representation of out-of-vocabulary words[5].

Our second finding is that concatenation is still a very strong baseline for meta-embeddings. This is not surprising because it preserves all of the information in the facets, and also introduces more model parameters, while the other meta-embedding methods seek to compress the information from all the meta-embeddings. Yet, for 4 out of the 6 tasks, word prisms outperform concatenation, and in the

[5] While Kiela et al. (2018) uses zero-vectors to represent OOV words, we opted to use the facet-level centroid as it resulted in better validation performance for most tasks. We found we were unable to exactly replicate their results even with zero-vectors.

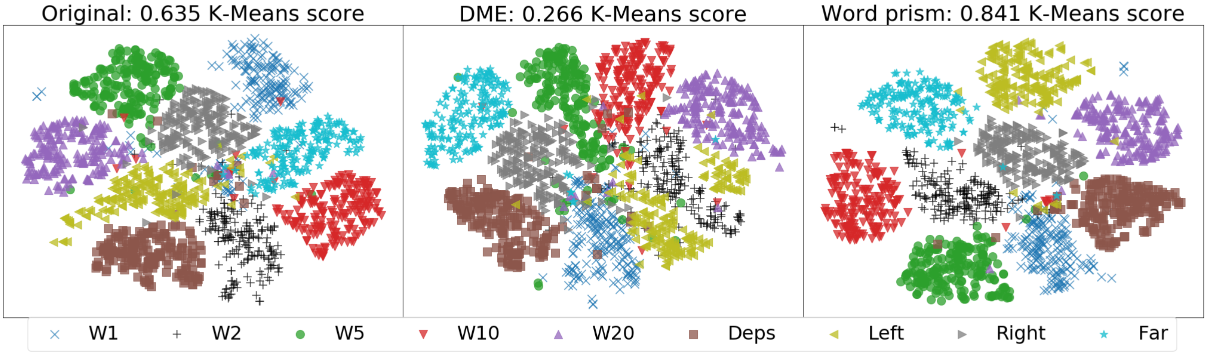

Figure 3: Impact of the linear transformations on the embedding space in dynamic meta-embeddings (DME) versus word prisms. The K-means score is the adjusted mutual information score for clustering embeddings into their respective facets, consistent across 5 clustering runs with different random seeds. Visualized is the TSNE-projected original facet-space versus the facets after projection (i.e., the stacked $\mathbf{w}_f$ versus the stacked $\mathbf{P}_f\mathbf{w}_f + \mathbf{b}_f$) in DMEs and word prisms.

other two tasks (Semcor and SST2) word prisms obtain second best results within the margin of error. Furthermore, note that concatenation is very expensive in the case of including 13 facets, requiring 3900-dimensional meta-embeddings at inference time, versus only 300-dimensions for word prisms.

Our third major finding is that sequence labelling tasks highly benefit from including all 13 facets, while the text classification tasks seem generally satisfied with only FG. Work on multi-task learning for supersense tagging on Semcor (using only a single set of embeddings with similar neural sequence labelling models) report results of 62.36 (Alonso and Plank, 2017) and 68.25 (Changpinyo et al., 2018). In contrast, our word prism model obtains a score of 73.82, indicating that word prisms can offer substantial improvements to supersense tagging models. Moreover, because supersense tagging is a coarse-grained version of word sense disambiguation, it is likely the case that word prisms can improve results in that domain as well. For further comparison, Huang et al. (2015) obtain accuracies of 96.04 and 83.52 on the WSJ and NER tasks respectively (when using a BiLSTM with only a single set of embeddings as features) while our word prisms offer improvements to 97.04 (+1.00) and 90.74 (+7.22).

Our results additionally contribute to the findings of Coates and Bollegala (2018) on the difference between averaging and concatenation for meta-embeddings. When there are 13 facets, we observe a marked drop in quality for averaging. We posit that this is due to the increased noise pollution in the averaged vector space because, while it is likely that two sets of random vectors will be generally orthogonal, when 13 sets of random vectors come into play the “birthday paradox” becomes much more likely to unveil itself, and the odds of being well-separated become lower. In Figure 3, we present the vector space for a subset of the nine window-based embeddings, projected to 2D with TSNE (Maaten and Hinton, 2008). Observe that the embeddings are not very well-separable on their own, as a K-Means clustering algorithm can only obtain an adjusted mutual information score of 0.635 to cluster them into their corresponding facets. We also present the embedding spaces being linearly transformed (i.e., $\mathbf{P}_f\mathbf{w}_f + \mathbf{b}_f$, see Eq. 1) when trained for supersense tagging. The learned, unconstrained transformations in DMEs cause the embedding space to become much less well-separated, as the clustering score deteriorates to 0.266. In contrast, the orthogonal transformations for word prisms improve the separatedness of the embedding space, bumping clustering score to 0.841. This (combined with the improved downstream results from word prisms) provides evidence of the validity of the “natural clustering” hypothesis for representation learning (Bengio et al., 2013; Kenyon-Dean et al., 2019); namely, that it is preferable for neural representations to be well separated (or, disentangled) within their latent spaces.

6 Ablation Studies

We perform two ablation studies to further inspect the impact of the orthogonal transformations and meta-embedding combination choice in word prisms. We compare the orthogonal transformation in word

Model	Proj.	Semcor	WSJ	Brown	NER	SNLI	SST2
Avg.	None	64.86 ± .2	96.53 ± .03	98.02 ± .02	86.33 ± .2	82.76 ± .2	85.55 ± .6
Prism	None	70.05 ± .2	96.73 ± .03	98.21 ± .02	87.56 ± .2	82.73 ± .4	85.61 ± .6
Prism	Uncon.	71.54 ± .1	96.90 ± .01	98.36 ± .00	87.68 ± .3	83.87 ± .2	86.15 ± .7
Prism	Orthog.	**72.41** ± .2	**96.98** ± .04	**98.44** ± .01	**88.91** ± .2	**83.89** ± .4	**86.68** ± .6

Table 2: **Ablation study 1**: *The transformation in word prisms.* Test set results for word prisms with different projection constraints (no transformation, an unconstrained one, and the orthogonal transformation), taking the nine window-based facets as the meta-embedding combination.

Facets	Semcor	WSJ	Brown	NER	SNLI	SST2
Best-Window	67.55 ± .1	96.60 ± .03	98.17 ± .01	87.54 ± .3	83.08 ± .2	85.82 ± .7
Best-All	70.50 ± .2	96.75 ± .03	98.42 ± .03	90.39 ± .2	85.60 ± .2	87.48 ± 1.0
W1-10	71.67 ± .2	96.86 ± .02	98.36 ± .02	88.39 ± .2	83.88 ± .3	86.52 ± .4
W1-Far	71.78 ± .1	96.89 ± .02	98.36 ± .01	88.54 ± .2	83.69 ± .3	87.39 ± .3
All windows	72.41 ± .2	96.98 ± .02	98.44 ± .01	88.91 ± .2	83.89 ± .4	86.68 ± .6
FG	73.51 ± .1	96.91 ± .01	98.58 ± .02	89.70 ± .4	**85.82** ± .1	87.80 ± .6
FGCL	73.51 ± .1	96.98 ± .03	98.63 ± .01	89.83 ± .3	85.68 ± .2	**88.90** ± .4
All	**73.82** ± .2	**97.04** ± .01	**98.65** ± .02	**90.74** ± .2	85.71 ± .3	88.45 ± .5

Table 3: **Ablation study 2**: *Different combinations of facets in word prisms.* Test set results when using different combinations of input facets in word prisms. First two rows contain only the single best performing window-based facet (Best-Window) or the best facet overall (Best-All), for the specific task.

prisms to two alternatives: *none* (i.e., $\mathbf{P}_f = \mathbf{I}_{d'}$, in which case the word prism is only learning the facet-combination weights α_f), and an *unconstrained* transformation which does not apply the orthogonality-imposing update rule detailed in Equation 2. For each experiment, we report the average and standard deviation across runs performed with five different random seeds.

Ablation study 1. In this experiment we determine the impact of the transformation matrix in word prisms when isolating the input facets to be the 9 window-based facets trained on the same dataset, with the same vocabulary, differing only in the definition of the context window. Table 2 presents the results for this experiment, which furthermore demonstrates the effectiveness of the orthogonal transformation.

Ablation study 2. In this experiment we investigate the impact of different choices of facet combinations for word prisms. We experiment with several different sets of facets: **W1-10** denotes window sizes between 1 and 10 (inclusive) [4 facets]; **W1-Far** includes the prior facets plus *W20* and *Far* [6 facets]; **All windows** includes the *Left*, *Right*, and *Deps* with the prior facets [9 facets]. **FG** denotes FastText and GloVe, while **FGCL** denotes FastText, GloVe, ConceptNet, and LexSub. **All** denotes all 13 of these facets. The results for these combinations are summarized in Table 3. In the first two rows, we also include the results for the best-performing (on the held out validation set) single-facet embedding model for the best window-based facet (**Best-Window**, out of the 9 window-based facets), and the best overall single-facet model (**Best-All** out of all 13 facets). *W2* is the best window-based facet for Semcor, WSJ (tie), Brown, NER, and SST2, while *W5* is the best window-based facet for WSJ (tie) and SNLI. GloVe is best overall for WSJ, Brown, NER, and SST2, while FastText is best overall for Semcor and SNLI.

We note three important takeaways from these results. First, word prisms always perform better than their single-facet counterparts, even though the single-facet models were selected to be the one that maximized validation performance for the specific task. Second, we observe that progressively incorporating more facets trained solely on different notions of context (i.e., from **W1-10** to **W1-Far** to **All windows**) improves results quality substantially for the sequence labelling tasks, while the text classification tasks (SNLI and SST2) do not benefit as much, although SST2 does seem to prefer the topic-based representations included by **W1-Far**. This suggests that NLP practitioners would benefit from training multiple sets of embeddings with different context windows in their specific problem domains (e.g., on a Twitter corpus), where they can expect improvements in results, especially if they are faced with a sequence labelling problem. Our third takeaway is that FastText and Glove are much better than the window-based embeddings, although including them all together still improves results in 3 out of the 6 tasks. This is not surprising since the FastText and Glove embeddings are trained on a corpus with over 600 billion words, while our window-based embeddings are only trained on a 6 billion word corpus; i.e., 1% of the data of the former. Thus, our results in this ablation study and the former suggest that training embeddings with different notions of context on such corpora will lead to even further gains.

7 Conclusion

In this paper, we study a simple and efficient method for constructing meta-embeddings from wide-ranging facets while preserving individual invariance with orthogonal transformations. The effectiveness of the proposed *word prisms* is validated by six supervised extrinsic evaluation tasks. Our word prism models obtain consistent improvements over dynamic meta-embeddings (Kiela et al., 2018) and the averaging and concatenation baselines (Coates and Bollegala, 2018) in all six tasks. Analysis of the transformed embeddings suggests the “natural clustering” hypothesis for representation learning (Bengio et al., 2013) is important to consider for combining various source embeddings to create performant task-specific meta-embeddings.

Several future directions present themselves from this work. First, we believe that contextualized embedding models can benefit from prismatic representations of their input embeddings (Devlin et al., 2019), and that word prisms can benefit from including contextualized embeddings as facets. Second, we expect that word prisms can improve performance in other tasks such as automatic summarization, which often use a single set of word embeddings in their input layers (Dong et al., 2019). Third, we believe that meta-embeddings and the method behind word prisms can be generalized past word-based representations to sentence representations (Pagliardini et al., 2018) and may improve their quality, as was recently demonstrated by Poerner et al. (2019). Lastly, recent work has found simple word embeddings to be useful for solving diverse problems from the medical domain (Zhang et al., 2019), to materials science (Tshitoyan et al., 2019), to law (Chalkidis and Kampas, 2019); we expect that word prisms and their motivations can further improve results in these applications.

Acknowledgments

This work is supported by the Fonds de recherche du Québec – Nature et technologies, by the Natural Sciences and Engineering Research Council of Canada, and by Compute Canada. The last author is supported in part by the Canada CIFAR AI Chair program.

References

Hector Martinez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL 2017-15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–10.

Kushal Arora, Aishik Chakraborty, and Jackie C. K. Cheung. 2020. Learning lexical subspaces in a distributional vector space. *Transactions of the Association for Computational Linguistics*, 8:311–329.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, November. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815. Association for Computational Linguistics.

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

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Danushka Bollegala, Kohei Hayashi, and Ken-Ichi Kawarabayashi. 2018. Think globally, embed locally: locally linear meta-embedding of words. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3970–3976. AAAI Press.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ilias Chalkidis and Dimitrios Kampas. 2019. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, 27(2):171–198.

Soravit Changpinyo, Hexiang Hu, and Fei Sha. 2018. Multi-task learning for sequence tagging: An empirical study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2965–2977.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 168–175. Association for Computational Linguistics.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 854–863. JMLR. org.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 194–198.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 294–304, Brussels, Belgium, October-November. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Iker García, Rodrigo Agerri, and German Rigau. 2020. A common semantic space for monolingual and cross-lingual meta-embeddings. *arXiv preprint arXiv:2001.06381*.

Sahar Ghannay, Benoit Favre, Yannick Esteve, and Nathalie Camelin. 2016. Word embedding evaluation and combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 300–305.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2007. English gigaword third edition ldc2007t07. *Web Download. Philadelphia: Linguistic Data Consortium*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Pratik Jawanpuria, Satya Dev N T V, Anoop Kunchukuttan, and Bamdev Mishra. 2020. Learning geometric word meta-embeddings. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 39–44, Online, July. Association for Computational Linguistics.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Kian Kenyon-Dean, Andre Cianflone, Lucas Page-Caccia, Guillaume Rabusseau, Jackie Chi Kit Cheung, and Doina Precup. 2019. Clustering-oriented representation learning with attractive-repulsive loss. *AAAI 2019 Workshop: Network Interpretability for Deep Learning*.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316.

Pierre Lison and Andrey Kutuzov. 2017. Redefining context windows for word embedding models: An experimental study. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 284–288.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June. Association for Computational Linguistics.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1030–1040.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics.

Edward Newell, Kian Kenyon-Dean, and Jackie Chi Kit Cheung. 2019. Deconstructing and reconstructing word embedding algorithms. *arXiv preprint arXiv:1911.13280*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition ldc2011t07. *Web Download. Philadelphia: Linguistic Data Consortium.*

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Sentence meta-embeddings for unsupervised semantic textual similarity. *arXiv preprint arXiv:1911.03700.*

Shota Sasaki, Jun Suzuki, and Kentaro Inui. 2019. Subword-based Compact Reconstruction of Word Embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3498–3508. Association for Computational Linguistics.

Adriaan M. J. Schakel and Benjamin J. Wilson. 2015. Measuring word significance using distributed representations of words.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017).*

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -.*

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763).

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951. Association for Computational Linguistics.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360.

Ye Zhang, Stephen Roller, and Byron C Wallace. 2016. Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of NAACL-HLT*, pages 1522–1527.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1).

Jinman Zhao, Sidharth Mudgal, and Yingyu Liang. 2018. Generalizing word embeddings using bag of subwords. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 601–606.