

Dynamic Meta-Embeddings for Improved Sentence Representations

Douwe Kiela[†], Changan Wang[‡] and Kyunghyun Cho^{†‡*}

[†] Facebook AI Research; [‡] New York University; * CIFAR Global Scholar

{dkiela, changan, kyunghyuncho}@fb.com

Abstract

While one of the first steps in many NLP systems is selecting what pre-trained word embeddings to use, we argue that such a step is better left for neural networks to figure out by themselves. To that end, we introduce dynamic meta-embeddings, a simple yet effective method for the supervised learning of embedding ensembles, which leads to state-of-the-art performance within the same model class on a variety of tasks. We subsequently show how the technique can be used to shed new light on the usage of word embeddings in NLP systems.

1 Introduction

It is no exaggeration to say that word embeddings have revolutionized NLP. From early distributional semantic models (Turney and Pantel, 2010; Erk, 2012; Clark, 2015) to deep learning-based word embeddings (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2016), word-level meaning representations have found applications in a wide variety of core NLP tasks, to the extent that they are now ubiquitous in the field (Goldberg, 2016).

A sprawling literature has emerged about what types of embeddings are most useful for which tasks. For instance, there has been extensive work on understanding what word embeddings learn (Levy and Goldberg, 2014b), evaluating their performance (Milajevs et al., 2014; Schnabel et al., 2015; Bakarov, 2017), specializing them for certain tasks (Maas et al., 2011; Faruqui et al., 2014; Kiela et al., 2015; Mrkšić et al., 2016; Vulić and Mrkšić, 2017), learning sub-word level representations (Wieting et al., 2016; Bojanowski et al., 2016; Lee et al., 2016), et cetera.

One of the first steps in designing many NLP systems is selecting what kinds of word embed-

dings to use, with people often resorting to freely available pre-trained embeddings. While this is often a sensible thing to do, the usefulness of word embeddings for downstream tasks tends to be hard to predict, as downstream tasks can be poorly correlated with word-level benchmarks. An alternative is to try to combine the strengths of different word embeddings. Recent work in so-called “meta-embeddings”, which ensembles embedding sets, has been gaining traction (Yin and Schütze, 2015; Bollegala et al., 2017; Muromägi et al., 2017; Coates and Bollegala, 2018). Meta-embeddings are usually created in a separate pre-processing step, rather than in a process that is dynamically adapted to the task. In this work, we explore the supervised learning of task-specific, dynamic meta-embeddings, and apply the technique to sentence representations.

The proposed approach turns out to be highly effective, leading to state-of-the-art performance within the same model class on a variety of tasks, opening up new areas for exploration and yielding insights into the usage of word embeddings.

Why Is This a Good Idea? Our technique brings several important benefits to NLP applications. First, it is embedding-agnostic, meaning that one of the main (and perhaps most important) hyperparameters in NLP pipelines is made obsolete. Second, as we will show, it leads to improved performance on a variety of tasks. Third, and perhaps most importantly, it allows us to overcome common pitfalls with current systems:

- **Coverage** One of the main problems with NLP systems is dealing with out-of-vocabulary words: our method increases lexical coverage by allowing systems to take the union over different embeddings.
- **Multi-domain** Standard word embeddings

are often trained on a single domain, such as Wikipedia or newswire. With our method, embeddings from different domains can be combined, optionally while taking into account contextual information.

- **Multi-modality** Multi-modal information has proven useful in many tasks (Baroni, 2016; Baltrušaitis et al., 2018), yet the question of multi-modal fusion remains an open problem. Our method offers a straightforward solution for combining information from different modalities.
- **Evaluation** While it is often unclear how to evaluate word embedding performance, our method allows for inspecting the weights that networks assign to different embeddings, providing a direct, task-specific, evaluation method for word embeddings.
- **Interpretability and Linguistic Analysis** Different word embeddings work well on different tasks. This is well-known in the field, but knowing *why* this happens is less well-understood. Our method sheds light on which embeddings are preferred in which linguistic contexts, for different tasks, and allows us to speculate as to why that is the case.

Outline In what follows, we explore dynamic meta-embeddings and show that this method outperforms the naive concatenation of various word embeddings, while being more efficient. We apply the technique in a BiLSTM-max sentence encoder (Conneau et al., 2017) and evaluate it on well-known tasks in the field: natural language inference (SNLI and MultiNLI; §4), sentiment analysis (SST; §5), and image-caption retrieval (Flickr30k; §6). In each case we show state-of-the-art performance within the class of single sentence encoder models. Furthermore, we include an extensive analysis (§7) to highlight the general usefulness of our technique and to illustrate how it can lead to new insights.

2 Related Work

Thanks to their widespread popularity in NLP, a sprawling literature has emerged about learning and applying word embeddings—much too large to fully cover here, so we focus on previous work that combines multiple embeddings for downstream tasks.

Maas et al. (2011) combine unsupervised embeddings with supervised ones for sentiment classification. Yang et al. (2016) and Miyamoto and Cho (2016) learn to combine word-level and character-level embeddings. Contextual representations have been used in neural machine translation as well, e.g. for learning contextual word vectors and applying them in other tasks (McCann et al., 2017) or for learning context-dependent representations to solve disambiguation problems in machine translation Choi et al. (2016).

Neural tensor skip-gram models learn to combine word, topic and context embeddings (Liu et al., 2015); context2vec (Melamud et al., 2016) learns a more sophisticated context representation separately from target embeddings; and Li et al. (2016) learn word representations with distributed word representation with multi-contextual mixed embedding. Recent work in “meta-embeddings”, which ensembles embedding sets, has been gaining traction (Yin and Schütze, 2015; Bollegala et al., 2017; Muromägi et al., 2017; Coates and Bollegala, 2018)—here, we show that the idea can be applied in context, and to sentence representations. Furthermore, these works obtain meta-embeddings as a preprocessing step, rather than learning them dynamically in a supervised setting, as we do here. Similarly to Peters et al. (2018), who proposed deep contextualized word representations derived from language models and which led to impressive performance on a variety of tasks, our method allows for contextualization, in this case of embedding set weights.

There has also been work on learning multiple embeddings per word (Chen et al., 2014; Nee-lakantan et al., 2015; Vu and Parker, 2016), including a lot of work in sense embeddings where the senses of a word have their own individual embeddings (Iacobacci et al., 2015; Qiu et al., 2016), as well as on how to apply such sense embeddings in downstream NLP tasks (Pilehvar et al., 2017).

The question of combining multiple word embeddings is related to multi-modal and multi-view learning. For instance, combining visual features from convolutional neural networks with word embeddings has been examined (Kielbaso and Bottou, 2014; Lazaridou et al., 2015), see Baltrušaitis et al. (2018) for an overview. In multi-modal semantics, for instance, word-level embeddings from different modalities are often mixed via concatenation $\mathbf{r} = [\alpha \mathbf{u}, (1 - \alpha) \mathbf{v}]$ (Bruni et al., 2014). Here, we

dynamically learn the weights to combine representations. Recently, related dynamic multi-modal fusion methods have also been explored (Wang et al., 2018; Kiros et al., 2018). There has also been work on unifying multi-view embeddings from different data sources (Luo et al., 2014).

The usefulness of different embeddings as initialization has been explored (Kocmi and Bojar, 2017), and different architectures and hyperparameters have been extensively examined (Levy et al., 2015). Problems with evaluating word embeddings intrinsically are well known (Faruqui et al., 2016), and various alternatives for evaluating word embeddings in downstream tasks have been proposed (e.g., Tsvetkov et al., 2015; Schnabel et al., 2015; Ettinger et al., 2016). For more related work with regard to word embeddings and their evaluation, see Bakarov (2017).

Our work can be seen as an instance of the well-known attention mechanism (Bahdanau et al., 2014), and its recent sentence-level incarnations of self-attention (Lin et al., 2017) and inner-attention (Cheng et al., 2016; Liu et al., 2016), where the attention mechanism is applied within the same sentence instead of for aligning multiple sentences. Here, we learn (optionally contextualized) attention weights for different embedding sets and apply the technique in sentence representations (Kiros et al., 2015; Wieting et al., 2015; Hill et al., 2016; Conneau et al., 2017).

3 Dynamic Meta-Embeddings

Commonly, NLP systems use a single type of word embedding, e.g., word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) or Fast-Text (Bojanowski et al., 2016). We propose giving networks access to multiple types of embeddings, allowing a network to learn which embeddings it prefers by predicting a weight for each embedding type, optionally depending on the context.

For a sentence of s tokens $\{t_j\}_{j=1}^s$, we have n word embedding types, leading to sequences $\{\mathbf{w}_{i,j}\}_{j=1}^s \in \mathbb{R}^{d_i}$ ($i = 1, 2, \dots, n$). We center each type of word embedding to zero mean.

Naive baseline We compare to naive concatenation as a baseline. Concatenation is a sensible strategy for combining different embedding sets, because it provides the sentence encoder with all of the information in the individual embeddings:

$$\mathbf{w}_j^{CAT} = [\mathbf{w}_{1,j}, \mathbf{w}_{2,j}, \dots, \mathbf{w}_{n,j}].$$

The downside of concatenating embeddings and giving that as input to an RNN encoder, however, is that the network then quickly becomes inefficient as we combine more and more embeddings.

DME For dynamic meta-embeddings, we project the embeddings into a common d' -dimensional space by learned linear functions $\mathbf{w}'_{i,j} = \mathbf{P}_i \mathbf{w}_{i,j} + \mathbf{b}_i$ ($i = 1, 2, \dots, n$) where $\mathbf{P}_i \in \mathbb{R}^{d' \times d_i}$ and $\mathbf{b}_i \in \mathbb{R}^{d'}$. We then combine the projected embeddings by taking the weighted sum

$$\mathbf{w}_j^{DME} = \sum_{i=1}^n \alpha_{i,j} \mathbf{w}'_{i,j}$$

where $\alpha_{i,j} = g(\{\mathbf{w}'_{i,j}\}_{j=1}^s)$ are scalar weights from a self-attention mechanism:

$$\alpha_{i,j} = g(\mathbf{w}'_{i,j}) = \phi(\mathbf{a} \cdot \mathbf{w}'_{i,j} + b) \quad (1)$$

where $\mathbf{a} \in \mathbb{R}^{d'}$ and $b \in \mathbb{R}$ are learned parameters and ϕ is a softmax (or could be a sigmoid or tanh, for gating). We also experiment with an **Unweighted** variant of this approach, that just sums up the projections.

CDME Alternatively, we can make the self-attention mechanism context-dependent, leading to contextualized DME (CDME):

$$\alpha_{i,j} = g(\{\mathbf{w}'_{i,j}\}_{j=1}^s) = \phi(\mathbf{a} \cdot \mathbf{h}_j + b) \quad (2)$$

where $\mathbf{h}_j \in \mathbb{R}^{2m}$ is the j^{th} hidden state of a BiLSTM taking $\{\mathbf{w}'_{i,j}\}_{j=1}^s$ as input, $\mathbf{a} \in \mathbb{R}^{2m}$ and $b \in \mathbb{R}$. We set $m = 2$, which makes the contextualization very efficient.

Sentence encoder We use a standard bidirectional LSTM encoder with max-pooling (BiLSTM-Max), which computes two sets of s hidden states, one for each direction:

$$\begin{aligned} \vec{\mathbf{h}}_j &= \overrightarrow{\text{LSTM}}_j(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_j) \\ \overleftarrow{\mathbf{h}}_j &= \overleftarrow{\text{LSTM}}_j(\mathbf{w}_j, \mathbf{w}_{j+1}, \dots, \mathbf{w}_s) \end{aligned}$$

The hidden states are subsequently concatenated for each timestep to obtain the final hidden states, after which a max-pooling operation is applied over their components to get the final sentence representation:

$$\mathbf{h} = \max(\{\{\vec{\mathbf{h}}_j, \overleftarrow{\mathbf{h}}_j\}\}_{j=1,2,\dots,s})$$

Model	SNLI	MNLI
InferSent (Conneau et al., 2017)	84.5	-
NSE (Munkhdalai and Yu, 2017)	84.6	-
G-TreeLSTM (Choi et al., 2017)	86.0	-
SSE (Nie and Bansal, 2017)	86.1	73.6
ReSan (Shen et al., 2018)	86.3	-
GloVe BiLSTM-Max (8.6M)	85.2±.3	70.0±.5
FastText BiLSTM-Max (8.6M)	85.2±.2	70.3±.3
Naive baseline (9.8M)	85.6±.3	71.1±.2
Naive baseline (61.3M)	86.0±.5	73.0±.2
Unweighted DME (8.6M)	86.3±.4	74.4±.2
DME (8.6M)	86.2±.2	74.4±.2
CDME (8.6M)	86.4±.3	74.1±.2
DME* (9.0M)	86.7±.2	74.3±.4
CDME* (9.0M)	86.5±.2	74.9±.5

Table 1: Accuracy scores on the Stanford Natural Language Inference (SNLI) and MultiNLI Mismatched (MNLI) tasks. DME=Dynamic Meta-Embeddings; CDME=Contextualized Dynamic Meta-Embeddings; *=multiple different embedding sets (see Section 4). Number of parameters included in parenthesis. Results averaged over five runs with different random seeds, using a BiLSTM-Max sentence encoder.

4 Natural Language Inference

Natural language inference, also known as recognizing textual entailment (RTE), is the task of classifying pairs of sentences according to whether they are neutral, entailing or contradictory. Inference about entailment and contradiction is **fundamental** to understanding natural language, and there are two established datasets to evaluate semantic representations in that setting: SNLI (Bowman et al., 2015) and the more recent MultiNLI (Williams et al., 2017).

The SNLI dataset consists of 570k human-generated English sentence pairs, manually labeled for entailment, contradiction and neutral. The MultiNLI dataset can be seen as an extension of SNLI: it contains 433k sentence pairs, taken from ten different genres (e.g. fiction, government text or spoken telephone conversations), with the same entailment labeling scheme.

We train sentence encoders with dynamic meta-embeddings using two well-known and often-used embedding types: FastText (Mikolov et al., 2018; Bojanowski et al., 2016) and GloVe (Pennington et al., 2014). Specifically, we make use of the 300-dimensional embeddings trained on a similar WebCrawl corpus, and compare three scenarios: when used individually, when naively concatenated or in the dynamic meta-embedding setting (unweighted, **context-independent** DME and

contextualized CDME). We also compare our approach against other models in the same class—in this case, models that encode sentences individually and **do not allow attention across the two sentences**.¹ We include InferSent (Conneau et al., 2017), which also makes use of a BiLSTM-Max sentence encoder.

In addition, we include a setting where we combine not two, but six different embedding types, adding FastText wiki-news embeddings², English-German and English-French embeddings from Hill et al. (2014), as well as the BOW2 embeddings from Levy and Goldberg (2014a) trained on Wikipedia.

4.1 Implementation Details

The **two sentences** are represented **individually** using the **sentence encoder**. As is standard in the literature, the **sentence representations** are subsequently **combined** using $\mathbf{m} = [\mathbf{u}, \mathbf{v}, \mathbf{u} * \mathbf{v}, |\mathbf{u} - \mathbf{v}|]$. We train a two-layer classifier with rectifiers on top of the combined representation. Notice that there is no interaction (e.g., attention) between the representations of \mathbf{u} and \mathbf{v} for this class of model.

We use 256-dimensional embedding projections, 512-dimensional BiLSTM encoders and an MLP with 1024-dimensional hidden layer in the classifier. The initial learning rate is set to 0.0004 and dropped by a factor of 0.2 when dev accuracy stops improving, dropout to 0.2, and we use Adam for optimization (Kingma and Ba, 2014). The loss is standard cross-entropy.

For MultiNLI, which has no designated validation set, we use the in-domain *matched* set for validation and report results on the out-of-domain *mis-matched* set.

4.2 Results

Table 1 shows the results. We report accuracy scores averaged over five runs with different random seeds, together with their standard deviation, for the SNLI and MultiNLI datasets. We include two versions of the naive baseline: one with a 512-dimensional BiLSTM encoder; and a bigger one with 2048 dimensions. Both naive baseline models outperform the single encoders that have only GloVe or FastText embeddings. This shows how

¹This is a common distinction, see e.g. the SNLI leaderboard at <https://nlp.stanford.edu/projects/snli/>.

²See <https://fasttext.cc/>

Model	SST
Const. Tree LSTM (Tai et al., 2015)	88.0
DMN (Kumar et al., 2016)	88.6
DCG (Looks et al., 2017)	89.4
NSE (Munkhdalai and Yu, 2017)	89.7
GloVe BiLSTM-Max (4.1M)	88.0±.1
FastText BiLSTM-Max (4.1M)	86.7±.3
Naive baseline (5.4M)	88.5±.4
Unweighted DME (4.1M)	89.0±.2
DME (4.1M)	88.7±.6
CDME (4.1M)	89.2±.4
CDME*-Softmax (4.6M)	89.3±.5
CDME*-Sigmoid (4.6M)	89.8±.4

Table 2: Sentiment classification accuracy results on the binary SST task. For DCG we compare against their best single sentence model (Looks et al., 2017). *=multiple different embedding sets (see Section 4). Number of parameters included in parenthesis. Results averaged over ten runs with different random seeds.

including more than one embeddings can help performance. Next, we observe that the DME embeddings outperform the naive concatenation baselines, while having fewer parameters. Differences between the three DME variants are small and not significant, although we do note that we found the highest maximum performance for the contextualized version, which adds very few additional parameters. It is important to note that the imposition of weighting thus is not detrimental to performance, which means that DME and CDME provide additional interpretability without sacrificing performance.

Finally, we obtain results for using the six different embedding types (marked *), and show that adding in more embeddings increases performance further. To our knowledge, these numbers constitute the state of the art within the model class of single sentence encoders on these tasks.

5 Sentiment

To showcase the general applicability of the proposed approach, we also apply it to a case where we have to classify a single sentence, namely, sentiment classification. Sentiment analysis and opinion mining have become important applications for NLP research. We evaluate on the binary SST task (Socher et al., 2013), consisting of 70k sentences with a corresponding binary (positive or

Model R@:	Image		Caption	
	1	10	1	10
VSE++	32.3	72.1	43.7	82.1
FastText (15M)	35.6	74.7	47.1	82.7
ImageNet (29M)	25.6	63.1	36.6	72.2
Naive (32M)	34.4	73.9	46.4	82.2
Unweighted DME (15M)	35.9	75.0	48.9	83.7
DME (15M)	36.5	75.5	49.7	83.6
CDME (15M)	36.5	75.6	49.0	83.8

Table 3: Image and caption retrieval results (R@1 and R@10) on Flickr30k dataset, compared to VSE++ baseline (Faghri et al., 2017). VSE++ numbers in the table are with ResNet features and random cropping, but no fine-tuning. Number of parameters included in parenthesis; averaged over five runs with std omitted for brevity.

negative) sentiment label.

5.1 Implementation Details

We use 256-dimensional embedding projections, 512-dimensional BiLSTM encoders and an MLP with 512-dimensional hidden layer in the classifier. The initial learning rate is set to 0.0004 and dropped by a factor of 0.2 when dev accuracy stops improving, dropout to 0.5, and we use Adam for optimization. The loss is standard cross-entropy. We calculate the mean accuracy and standard deviation based on ten random seeds.

5.2 Results

Table 2 shows a similar pattern as we observed with NLI: the naive baseline outperforms the single-embedding encoders; the DME methods outperform the naive baseline, with the contextualized version appearing to work best. Finally, we experiment with replacing ϕ in Eq. 1 and 2 with a sigmoid gate instead of a softmax, and observe improved performance on this task, outperforming the comparable models listed in the table. These results further strengthen the point that having multiple different embeddings helps, and that we can learn to combine those different embeddings efficiently, in interpretable ways.

6 Image-Caption Retrieval

An advantage of the proposed approach is that it is inherently capable of dealing with multi-modal information. Multi-modal semantics (Bruni et al., 2014) often combines linguistic and visual representations via concatenation with a global weight

α , i.e., $\mathbf{v} = [\alpha \mathbf{v}_{ling}, (1 - \alpha) \mathbf{v}_{vis}]$. In DME we instead learn to combine embeddings dynamically, optionally based on context. The representation for a word then becomes grounded in the visual modality, and we encode on the word-level *what things look like*.

We evaluate this idea on the Flickr30k image-caption retrieval task: given an image, retrieve the correct caption; and vice versa. The intuition is that knowing what something looks like makes it easier to retrieve the correct image/caption. While this work was under review, a related method was published by [Kiros et al. \(2018\)](#), which takes a similar approach but evaluates its effectiveness on COCO and uses Google images. We obtain word-level *visual embeddings* by retrieving relevant images for a given label from ImageNet in the same manner as [Kielbaso and Bottou \(2014\)](#), taking the images’ ResNet-152 features ([He et al., 2016](#)) and subsequently averaging those. We then learn to combine textual (FastText) and visual (ImageNet) word representations in the caption encoder used for retrieving relevant images.

6.1 Implementation Details

Our loss is a max-margin rank loss as in VSE++ ([Faghri et al., 2017](#)), a state-of-the-art method on this task. The network architecture is almost identical to that system, except that we use DME (with 256-dimensional embedding projection) and a 1024-dimensional caption encoder. For the Flickr30k images that we do retrieval over, we use random cropping during training for data augmentation and use a ResNet-152 for feature extraction. We tune the sizes of the encoders and use a learning rate of 0.0003 and a dropout rate of 0.1.

6.2 Results

Table 3 shows the results, comparing against VSE++. First, note that the ImageNet-only embeddings don’t work as well as the FastText ones, which is most likely due to poorer coverage. We observe that DME outperforms naive and FastText-only, and outperforms VSE++ by a large margin. These findings confirm the intuition that knowing what things look like (i.e., having a word-level visual representation) improves performance in visual retrieval tasks (i.e., where we need to find relevant images for phrases or sentences)—something that sounds obvious but has not really been explored before, to our knowledge. This

showcases DME’s usefulness for fusing embeddings in multi-modal tasks.

7 Discussion & Analysis

Aside from improved performance, an additional benefit of learning dynamic meta-embeddings is that they enable inspection of the weights that the network has learned to assign to the respective embeddings. In this section, we perform a variety of smaller experiments in order to highlight the usefulness of the technique for studying linguistic phenomena, determining appropriate training domains and evaluating word embeddings. We compute the contribution of each word embedding type as follows:

$$\beta_{i,j} = \frac{\|\alpha_{i,j} \mathbf{w}'_{i,j}\|_2}{\sum_{k=1}^n \|\alpha_{k,j} \mathbf{w}'_{k,j}\|_2}$$

7.1 Visualizing Attention

Figure 1 shows the attention weights for a CDME model trained on SNLI, using the aforementioned six embedding sets. The sentence is from the SNLI validation set. We observe that different embeddings are preferred for different words. The figure is meant to illustrate possibilities for analysis, which we turn to in the next section.

7.2 Linguistic Analysis

We perform a fine-grained analysis of the behavior of DME on the validation set of SNLI. Figure 3 shows a breakdown of the average attention weights per part of speech. Figure 4 shows a similar breakdown for open versus closed class. The analysis allows us to make several interesting observations: it appears that this model prefers GloVe embeddings, followed by the two FastText embeddings (trained on Wikipedia and Common Crawl). For open class words (e.g., nouns, verbs, adjectives and adverbs), those three embedding types are strongly preferred, while closed class words get more evenly divided attention. The embeddings from [Levy and Goldberg \(2014a\)](#) get low weights, possibly because the method is complementary with FastText-wiki, which was trained on a more recent version of Wikipedia.

We can further examine the attention weights by analyzing them in terms of frequency and concreteness. We use Norvig’s Google N-grams corpus frequency counts³ to divide the words into fre-

³<http://norvig.com/mayzner.html>

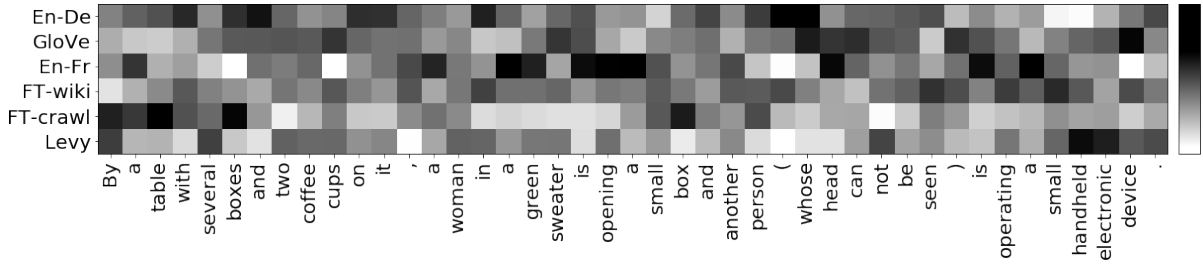


Figure 1: Example visualization of a sentence from the SNLI dev set.

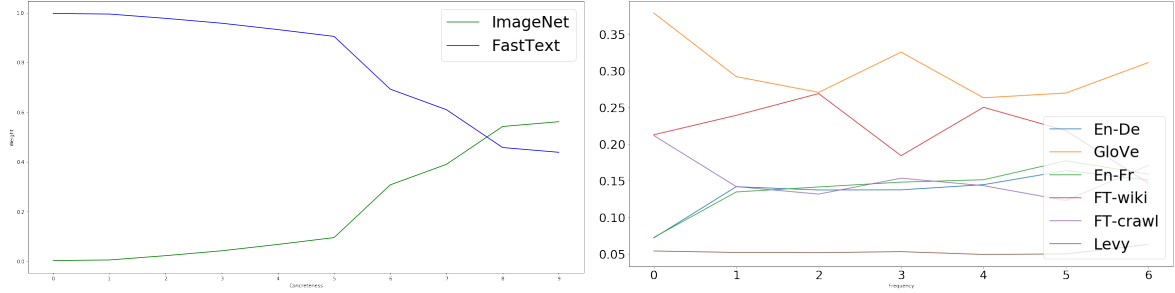


Figure 2: Concreteness weights (left) for Flickr30k model and Frequency weights (right) for SNLI model with multiple embeddings. Visual ImageNet embeddings are preferred for more concrete words. GloVe is strongly preferred for low-frequency words.

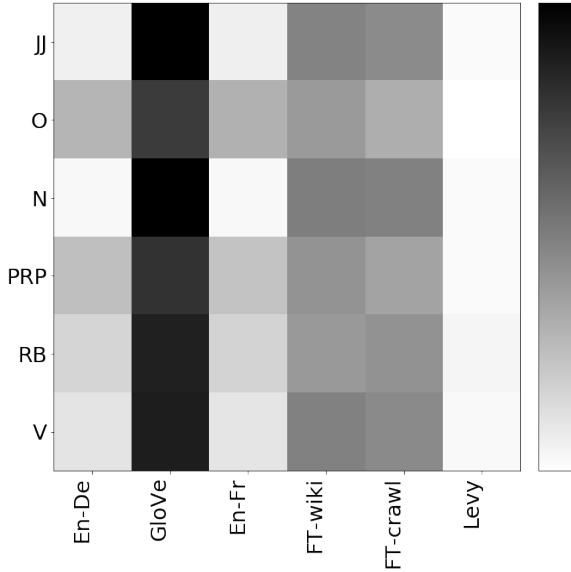


Figure 3: Pos tags and weights by embedding type.

quency bins. Figure 2 (right) shows the average attention weights per frequency bin, ranging from low to high. We observe a clear preference for GloVe, in particular for low-frequency words. For concreteness, we use the concreteness ratings from Brysbaert et al. (2014). Figure 2 (left) shows the average weights per concreteness bin for a model trained on Flickr30k. We can clearly see that visual embeddings get higher weights as the words

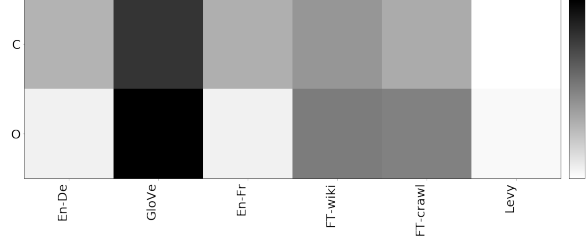


Figure 4: Weights for open/closed class words.

become more and more concrete.

There are of course intricate relationships between concreteness, frequency, POS tags and open/closed class words: closed class words are often frequent and abstract, while open class words could be more concrete, etc. It is beyond the scope of the current work to explore these further, but we hope that others will pursue this direction in future work.

7.3 Multi-domain Embeddings

The MultiNLI dataset consists of various genres. This allows us to inspect the applicability of source domain data for a specific genre. We train embeddings on three kinds of data: Wikipedia, the Toronto Books Corpus (Zhu et al., 2015) and the English OpenSubtitles⁴. We examine the atten-

⁴<http://opus.nlpl.eu/OpenSubtitles.php>

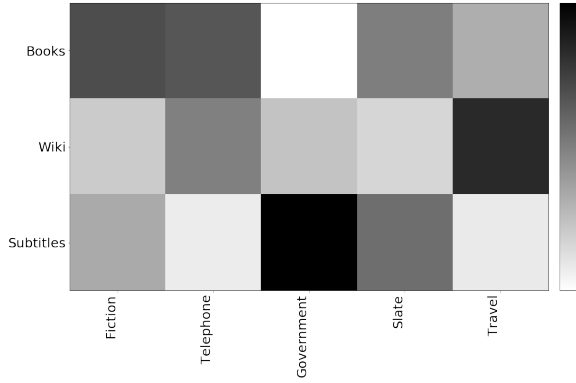


Figure 5: Multi-domain weights on MultiNLI.

Model	Levy	LEAR	SNLI
CDME	0.33	0.67	85.3±.9

Model	GloVe	Refined	SST
CDME	0.59	0.41	89.0±.4

Table 4: Accuracy and learned weights on SNLI using LEAR (Vulić and Mrkšić, 2017) or SST using sentiment-refined embeddings using the specialization method from Yu et al. (2017).

tion weights on the five genres in the in-domain (matched) set, consisting of fiction; transcriptions of spoken telephone conversations; government reports, speeches, letters and press releases; popular culture articles from the Slate Magazine archive; and travel guides.

Figure 5 shows the average attention weights for the three embedding types over the five genres. We observe that Toronto Books, which consists of fiction, is very appropriate for the fiction genre, while Wikipedia is highly preferred for the travel genre, perhaps because it contains a lot of factual information about geographical locations. The government genre makes more use of OpenSubtitles. The spoken telephone genre does not appear to prefer OpenSubtitles, which we might have expected given that that corpus would contain spoken dialogue, but Toronto books, which does include written dialogue.

7.4 Specialization

The above shows that we can use DME to analyze different embeddings on a task. Given the recent interest in the community in specializing, retro-fitting and counter-fitting word embeddings for given tasks, we examine whether the lexical-

level benefits of specialization extend to sentence-level downstream tasks. After all, one of the main motivations behind work on lexical entailment is that it allows for better downstream textual entailment. Hence, we take the LEAR embeddings by Vulić and Mrkšić (2017), which do very well on the HyperLex lexical entailment evaluation dataset (Vulić et al., 2017). We compare their best-performing embeddings against the original embeddings that were used for specialization, derived from the BOW2 embeddings of Levy and Goldberg (2014a). Similarly, we use the technique of Yu et al. (2017) for refining GloVe embeddings for sentiment, and evaluate model performance on the SST task.

Table 4 shows that LEAR embeddings get high weights compared to the original source embeddings (“Levy” in the table). Our analysis showed that LEAR was particularly favored for verbs (with average weights of 0.75). The sentiment-refined embeddings were less useful, with the original GloVe embeddings receiving higher weights. These preliminary experiments show how DME models can be used for analyzing the performance of specialized embeddings in downstream tasks.

Note that different weighting mechanisms might give different results—we found that the normalization strategy and the depth of the network significantly influenced weight assignments in our experiments with specialized embeddings.

7.5 Examining Contextualization

We examined models trained on SNLI and looked at the variance of the attention weights per word in the dev set. If contextualization is important for getting the classification decision correct, then we would expect big differences in the attention weights per word depending on the context. Upon examination, we only found relatively few differences. In part, this may be explained by the small size of the dev set, but for the GloVe+FastText model we inspected there were only around twenty words with any variance at all, which suggests that the field needs to work on more difficult semantic benchmark tasks. The words, however, were characterized by their polysemy, in particular by having both noun and verb senses. The following words were all in the top 20 most context-dependent words: *mob*, *boards*, *winds*, *trains*, *pitches*, *camp*.

8 Conclusion

We argue that the decision of which word embeddings to use in what setting should be left to the neural network. While people usually pick one type of word embeddings for their NLP systems and then stick with it, we find that dynamically learned meta-embeddings lead to improved results. In addition, we showed that the proposed mechanism leads to better interpretability and insightful linguistic analysis. We showed that the network learns to select different embeddings for different data, different domains and different tasks. We also investigated embedding specialization and examined more closely whether contextualization helps. To our knowledge, this work constitutes the first effort to incorporate multi-modal information on the language side of image-caption retrieval models; and the first attempt at incorporating meta-embeddings into large-scale sentence-level NLP tasks.

In future work, it would be interesting to apply this idea to different tasks, in order to explore what kinds of embeddings are most useful for core NLP tasks, such as tagging, chunking, named entity recognition, parsing and generation. It would also be interesting to further examine specialization and how it transfers to downstream tasks. Using this method for evaluating word embeddings in general, and how they relate to sentence representations in particular, seems a fruitful direction for further exploration. In addition, it would be interesting to explore how the attention weights change during training, and if, e.g., introducing entropy regularization (or even negative entropy) might improve results or interpretability further.

Acknowledgments

We thank the anonymous reviewers for their comments. We also thank Marcus Rohrbach, Laurens van der Maaten, Ivan Vulić, Edouard Grave, Tomas Mikolov and Maximilian Nickel for helpful suggestions and discussions with regard to this work.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Amir Bakarov. 2017. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*.
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Marco Baroni. 2016. Grounding distributional semantics in the visual world. *Language and Linguistics Compass*, 10(1):3–13.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2017. Think globally, embed locally - locally linear meta-embedding of words. *arxiv preprint arXiv:1709.06671*.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.
- Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.
- Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.
- Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.
- Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. 2016. Context-dependent word representation for neural machine translation. *arXiv preprint arXiv:1607.00578*.
- Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2017. Unsupervised learning of task-specific tree structures with tree-lstms. *arXiv preprint arXiv:1707.02786*.
- Stephen Clark. 2015. Vector space models of lexical meaning. *Handbook of Contemporary Semantic Theory, The*, pages 493–522.

- Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding—computing meta-embeddings by averaging source word embeddings. In *Proceedings of NAACL-HLT*.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP*.
- Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.
- Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139.
- Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*.
- Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.
- Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.
- Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *J. Artif. Intell. Res.(JAIR)*, 57:345–420.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Felix Hill, Kyunghyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. 2014. Embedding word similarity with neural machine translation. *arXiv preprint arXiv:1412.6448*.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.
- Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*, volume 1, pages 95–105.
- Douwe Kiela and Léon Bottou. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45.
- Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of EMNLP*, pages 2044–2048.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Jamie Kiros, William Chan, and Geoffrey Hinton. 2018. Illustrative language understanding: Large-scale visual grounding with image search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 922–933.
- Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.
- Tom Kocmi and Ondřej Bojar. 2017. An exploration of word embedding initialization in deep-learning tasks. *arXiv preprint arXiv:1711.09160*.
- Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.
- Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. *arXiv preprint arXiv:1501.02598*.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017*.
- 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)*, volume 2, 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.

- Jianqiang Li, Jing Li, Xianghua Fu, Md Abdul Masud, and Joshua Zhexue Huang. 2016. Learning distributed word representation with multi-contextual mixed embedding. *Knowledge-Based Systems*, 106:220–230.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.
- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2015. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of IJCAI*, pages 1284–1290.
- Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. *CoRR*, abs/1605.09090.
- Moshe Looks, Marcello Herreshoff, DeLesley Hutchins, and Peter Norvig. 2017. Deep learning with dynamic computation graphs. *arXiv preprint arXiv:1702.02181*.
- Yong Luo, Jian Tang, Jun Yan, Chao Xu, and Zheng Chen. 2014. Pre-trained multi-view word embedding using two-side neural network. In *Proceedings of AAAI*.
- Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, pages 142–150.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.
- Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.
- Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhersch, and Armand Joulin. 2018. Learning word vectors for 157 languages. In *Proceedings of LREC*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. 2014. Evaluating neural word representations in tensor-based compositional settings. *arXiv preprint arXiv:1408.6179*.
- Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. *arXiv preprint arXiv:1606.01700*.
- Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL*.
- Tsendsuren Munkhdalai and Hong Yu. 2017. Neural semantic encoders. In *Proceedings of ACL*, volume 1, page 397.
- Avo Muromägi, Kairit Sirts, and Sven Laur. 2017. Linear ensembles of word embedding models. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 96–104.
- Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2015. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.
- Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. *arXiv preprint arXiv:1708.02312*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Mohammad Taher Pilehvar, José Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a seamless integration of word senses into downstream NLP applications. *arXiv preprint arXiv:1710.06632*.
- Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-dependent sense embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 183–191.
- Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of EMNLP*, pages 298–307.
- Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. *arXiv preprint arXiv:1801.10296*.
- 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 EMNLP*, pages 1631–1642.
- Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

- Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of EMNLP*, pages 2049–2054.
- Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.
- Thuy Vu and D Stott Parker. 2016. *k*-embeddings: Learning conceptual embeddings for words using context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1262–1267.
- Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.
- Ivan Vulić and Nikola Mrkšić. 2017. Specialising word vectors for lexical entailment. *arXiv preprint arXiv:1710.06371*.
- Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2018. Learning multimodal word representation via dynamic fusion methods. *arXiv preprint arXiv:1801.00532*.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W. Cohen, and Ruslan Salakhutdinov. 2016. Words or characters? fine-grained gating for reading comprehension. *arXiv preprint arXiv:1611.01724*.
- Wenpeng Yin and Hinrich Schütze. 2015. Learning word meta-embeddings by using ensembles of embedding sets. *arXiv preprint arXiv:1508.04257*.
- Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 534–539.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of ICCV*, pages 19–27.