Appendices

In these appendices, we provide supplementary details on the experiments, mathematical framework, and detailed results in Section S1. In Section S2 we discuss computational aspects and the importance of clustering the contexts. Detailed results of the sentence representation and hypernymy detection experiments are listed on the following pages in Section S3 and S6 respectively. Then we describe a qualitative analysis of sentence similarity in Section S4, and finally discuss a qualitative analysis of hypernymy detection in Section S7.

Contents

S1 Technical specifications

In this Section, we give further details on the experimental framework in Section S1.1, on the PPMI formulation (Section S1.2), and on Optimal Transport (Section S1.3). In Section S1.5, we provide references for software release.

S1.1 Experimental Details

Sentence Representations. While using the Toronto Book Corpus, we remove the errors caused by crawling and pre-process the corpus by filtering out sentences longer than 300 words, thereby removing a very small portion (500 sentences out of the 70 million sentences). We utilize the code[S1] from GloVe for building the vocabulary of size 205513 (obtained by setting min_count=10) and the co-occurrence matrix (considering a symmetric window of size 10). Note that as in GloVe, the contribution from a context word is inversely weighted by the distance to the target word, while computing the co-occurrence. The vectors obtained via GloVe have 300 dimensions and were trained for 75 iterations at a learning rate of 0.005, other parameters being the default ones. The performance of these vectors from GloVe was verified on standard word similarity tasks.

Hypernymy Detection. The training of the entailment vector is performed on a Wikipedia dump from 2015 with 1.7B tokens that have been tokenized using the Stanford NLP library (Manning et al., 2014). In our experiments, we use a vocabulary with a size of 80'000 and word embeddings with 200 dimensions. We followed the same training procedure as described in Henderson (2017) and were able to reproduce their scores on the hypernymy detection task. For tuning the hyperparameters, we utilize the HypeNet training set of Shwartz et al. (2016) (from the random split), following the procedure indicated in Chang et al. (2017) for tuning DIVE and Gaussian embeddings.

S1.2 PPMI Details

Definition. The Positive Pointwise Mutual Information (PPMI) matrix (Church and Hanks, 1990; Levy et al., 2015) is defined as follows:

$$\text{PPMI}(w,c) := \max\left(\log\left(\frac{p(w,c)}{p(w)\times p(c)}\right), 0\right). \tag{5}$$

This means that the PPMI entries are non-zero when the joint probability of target and context words co-occurring is higher than the probability when they are independent.

Formulation and Variants. Typically, the probabilities used in PPMI are estimated from the co-occurrence counts $\#(w,c)$ in the corpus and lead to

$$\text{PPMI}(w,c) = \max\left(\log\left(\frac{\#(w,c)\times|Z|}{\#(w)\times\#(c)}\right), 0\right), \tag{6}$$

where, $\#(w) = \sum_c \#(w,c)$, $\#(c) = \sum_w \#(w,c)$ and $|Z| = \sum_w \sum_c \#(w,c)$. Also, it is known that PPMI is biased towards infrequent words and assigns them a higher value. A common solution is to smoothen[S2] the context probabilities by raising them to an exponent of α lying between 0 and 1. Levy and Goldberg (2014b) have also suggested the use of the shifted PPMI (SPPMI) matrix where the shift[S3] by $\log(s)$ acts like a prior on the probability of co-occurrence of target and context pairs. These variants of PPMI enable us to extract better semantic associations from the co-occurrence matrix. Finally, we have

$$\text{SPPMI}_{\alpha,s}(w,c) := \max\left(\log\left(\frac{\#(w,c)\times\sum_{c'}\#(c')^\alpha}{\#(w)\times\#(c)^\alpha}\right) - \log(s), 0\right),$$

[S1] https://github.com/stanfordnlp/GloVe

[S2] $p_\alpha(c) := \frac{\#(c)^\alpha}{\sum_{c'}\#(c')^\alpha}$.

[S3] Here, we denote the shift parameter by s instead of the k defined in (Levy et al., 2015) to avoid confusion with the other usage of k.

where α and s denote the smoothing and k-shift parameters. Hence, the bin values (at context c) for the histogram of word w in Eq. (3) can be written as:

$$(\mathrm{H}^{w})_{c} := \frac{\mathrm{SPPMI}_{\alpha,s}(w,c)}{\sum_{c\,\in\mathcal{C}} \mathrm{SPPMI}_{\alpha,s}(w,c)}. \tag{7}$$

Computational aspect. We utilize the sparse matrix support of Scipy[S4] for efficiently carrying out all the PPMI computations.

PPMI Column Normalizations. Now, instead of the individual contexts we consider the PPMI with respect to the representative contexts (i.e., cluster centers). In certain cases, when the PPMI contributions towards the partitions (or clusters) have a large variance, it can be helpful to consider the fraction of $\mathcal{C}_k$'s SPPMI (Eq. (8), (9)) that has been used towards a word w, instead of aggregate values used in (12). Otherwise the process of making the histogram unit sum might misrepresent the actual underlying contribution. We call this PPMI column normalization (β). In other words, the intuition is that the normalization will balance the effect of a possible non-uniform spread in total PPMI across the clusters. We observe that setting β to 0.5 or 1 help in boosting performance on the STS tasks. The basic form of column normalization is shown in (9).

$$(\tilde{\mathrm{H}}^{w})_{k} := \frac{(\bar{\mathrm{H}}^{w})_{k}}{\sum_{k=1}^{K} (\bar{\mathrm{H}}^{w})_{k}} \quad \text{with} \tag{8}$$

$$(\bar{\mathrm{H}}^{w})_{k} := \frac{\mathrm{SPPMI}_{\alpha,s}(w,\mathcal{C}_k)}{\sum_{w} \mathrm{SPPMI}_{\alpha,s}(w,\mathcal{C}_k)}. \tag{9}$$

Another possibility while considering the normalization to have an associated parameter β that can interpolate between the above normalization and normalization with respect to cluster size.

$$\begin{aligned} (\tilde{\mathrm{H}}^{w}_{\beta})_{k} &:= \frac{(\bar{\mathrm{H}}^{w}_{\beta})_{k}}{\sum_{k=1}^{K} (\bar{\mathrm{H}}^{w}_{\beta})_{k}}, \quad \text{where} \\ (\bar{\mathrm{H}}^{w}_{\beta})_{k} &:= \frac{\mathrm{SPPMI}_{\alpha,s}(w,\mathcal{C}_k)}{\sum_{w} \mathrm{SPPMI}_{\alpha,s}(w,\mathcal{C}_k)^{\beta}} \end{aligned} \tag{10}$$

In particular, when $\beta = 1$, we recover the equation for histograms as in (9), and $\beta = 0$ would imply normalization with respect to cluster sizes.

S1.3 Optimal Transport

Implementation aspects. We make use of the Python Optimal Transport (POT)[S5] for performing the computation of Wasserstein distances and barycenters on CPU. For more efficient GPU implementation, we built custom implementation using PyTorch. We also implement a batched version for barycenter computation, which to the best of our knowledge has not been done in the past. The batched barycenter computation relies on a viewing computations in the form of block-diagonal matrices. As an example, this batched mode can compute around 200 barycenters in 0.09 seconds, where each barycenter is of 50 histograms (of size 100) and usually gives a speedup of about 10x.

Scalability. For further scalability, an alternative is to consider *stochastic optimal transport* techniques (Genevay et al., 2016). Here, the idea would be to randomly sample a subset of contexts from the distributional estimate while considering this transport.

Stability of Sinkhorn Iterations. For all our computations involving optimal transport, we typically use λ around 0.1 and make use of log or median normalization as common in POT to stabilize the Sinkhorn iterations. Also, we observe that clipping the ground metric matrix (if it exceeds a particular large threshold) also sometimes results in performance gains.

[S4] https://docs.scipy.org/doc/scipy/reference/sparse.html

[S5] http://pot.readthedocs.io/en/stable/

Value of p. It has been shown in Agueh and Carlier (2011) that when the underlying space is Euclidean and $p = 2$, there exists a unique minimizer to the Wasserstein barycenter problem. But, since we are anyways solving the regularized Wasserstein barycenter (Cuturi and Doucet, 2014) problem over here instead of the exact one, the particular value of p seems less of an issue. Empirically in the sentence similarity experiments, we have observed $p = 1$ to perform better than $p = 2$ (by about 2-3 points).

S1.4 Empirical runtime

Starting from scratch, it takes less than 11 minutes to get the results on all STS tasks which contains 25,000 sentences. This includes about 3 minutes to cluster 200,000 words (1 GPU), 5 minutes to convert raw co-occurrences into histograms of size 300 (1 CPU core) and 3 minutes for STS (1 GPU).

S1.5 Software Release

Core code and histograms. Our code to build the ppmi-matrix, clusters, histograms as well computing Wasserstein distances and barycenters is publicly available on Github under `https://github.com/context-mover`. Precomputed histograms, clusters and point embeddings used in our experiments can also be downloaded from `https://drive.google.com/open?id=13stRuUd--71hc0q92yWUF-0iY15DYKNf`.

Standard evaluation suite for Hypernymy. To ease the evaluation pipeline, we have collected the most common benchmark datasets and compiled the code for assessing a model's performance on hypernymy detection or directionality into a Python package, called **HypEval**, which is publicly available at `https://github.com/context-mover/HypEval`. This also handles OOV (out-of-vocabulary) pairs in a standardized manner and allows for efficient, batched evaluation on GPU.

S2 Clustering the contexts

In this Section, we discuss computational aspects and how using clustering makes the problem scalable. We give precise definition of the distributional estimate in Section S2.1, and show how the number of clusters affects the performance in Section S2.3.

S2.1 Computational considerations.

The view of optimal transport between histograms of contexts introduced in Eq. (4) offers a pleasing interpretation (see Figure 2). However, it might be computationally intractable in its current formulation, since the number of possible contexts can be as large as the size of vocabulary (if the contexts are just single words) or even exponential (if contexts are considered to be phrases, sentences and otherwise). For instance, even with the use of SPPMI matrix, which also helps to sparsify the co-occurrences, the cardinality of the support of histograms still varies from 10^3 to 5×10^4 context words, when considering a vocabulary of size around 2×10^5.

This is problematic because the Sinkhorn algorithm for regularized optimal transport (Cuturi, 2013, see Section 3) scales roughly quadratically in the histogram size, and the ground cost matrix can also become prohibitive to store in memory. One possible fix is to instead consider a set of representative contexts in this ground space, for example via clustering. We believe that with dense low-dimensional embeddings and a meaningful metric between them, we may not require as many contexts as needed before. For instance, this can be achieved by clustering the contexts with respect to metric $D_{\mathcal{G}}$. Apart from the computational gain, the clustering will lead to transport between more abstract contexts. This will although come at the loss of some interpretability.

Now, consider that we have obtained K representative contexts, each covering some part $\mathcal{C}_k$ of the set of contexts $\mathcal{C}$. The histogram for word w with respect to these contexts can then be written as:

$$\tilde{\mathbb{P}}^w_{\tilde{V}} = \sum_{k=1}^{K} (\tilde{\mathrm{H}}^w)_k \; \delta(\tilde{\mathbf{v}}_k). \tag{11}$$

Here $\tilde{\mathbf{v}}_k \in \tilde{V}$ is the point estimate of the k^{th} representative context, and $(\tilde{\mathrm{H}}^w)_k$ denote the new histogram bin values with respect to the part $\mathcal{C}_k$,

$$(\tilde{\mathrm{H}}^w)_k := \frac{\mathrm{SPPMI}_{\alpha,s}(w, \mathcal{C}_k)}{\sum_{k=1}^K \mathrm{SPPMI}_{\alpha,s}(w, \mathcal{C}_k)}, \text{with} \tag{12}$$

$$\mathrm{SPPMI}_{\alpha,s}(w, \mathcal{C}_k) := \sum_{c \in \mathcal{C}_k} \mathrm{SPPMI}_{\alpha,s}(w, c). \tag{13}$$

In the following subsection, we show the effect of the number of clusters on the performance.

S2.2 Implementation.

For clustering, we make use of kmcuda's[S6] efficient implementation of K-Means algorithm on GPUs.

S2.3 Effect of number of clusters

Here, we analyze the impact of number of clusters on the performance of Context Mover's Barycenters (CoMB) for the sentence similarity experiments (cf. Section 5). In particular, we look at the three best performing variants (A, B, C) on the validation set (STS 16) as well as averaged across them.

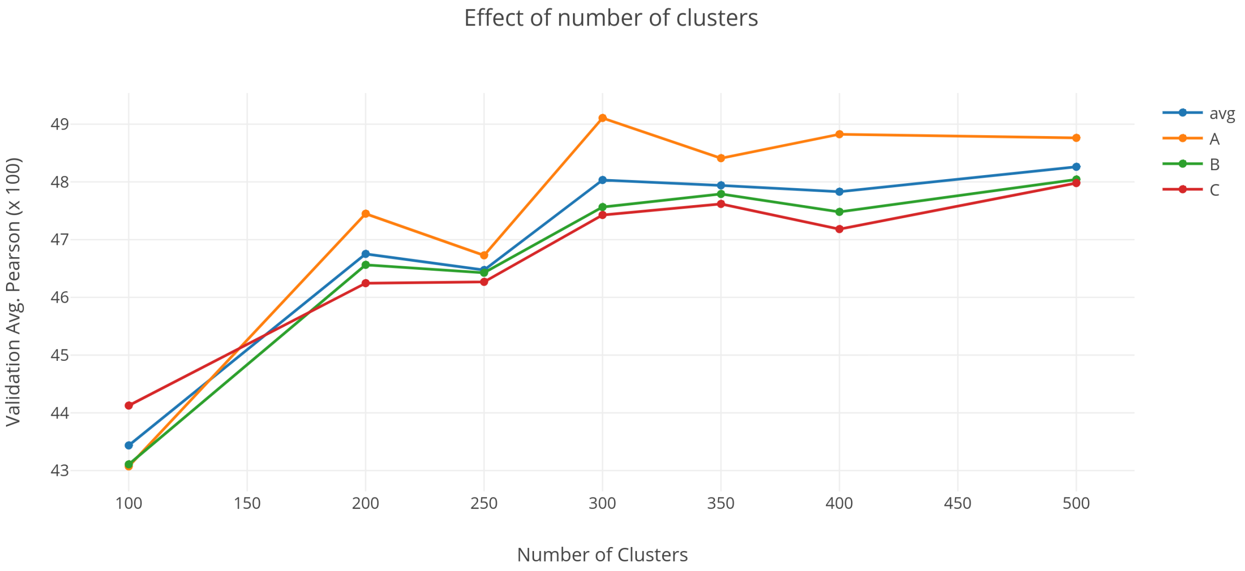

Figure S1: Effect of the number of clusters (K) on validation performance. A, B, C correspond to the three best performing variants of CoMB obtained as per validation on STS16 and as presented in the Table 1. In particular, A denotes the hyperparameter setting of [(α=0.55, β=1, s=5], B refers to [α=0.55, β=0.5, s=5] and C denotes [α=0.15, β=0.5, s=1]. The *'avg'* plot shows the average trend across these three configurations.

We observe in Figure S1 that on average the performance significantly improves when the number of clusters are increased until around $K = 300$, and beyond that mostly plateaus (± 0.5). But, as can be seen for variants B and C the performance typically continues to rise until $K = 500$. It seems that the amount of PPMI column normalization ($\beta = 0.5$ vs $\beta = 1$) might be at play here.

As going from $K = 300$ to $K = 500$ comes at the cost of increased computation time, and doesn't lead to a substantial gain in performance. We use either $K = 300$ or 500 clusters, depending on validation results, for our results on sentence similarity tasks.

Such a trend seems to be in line with the ideal case where we wouldn't need to do any clustering and just take all possible contexts into account.

[S6] https://github.com/src-d/kmcuda

S3 Sentence Representation

S3.1 Detailed results

We provide detailed results of the **test set performance** of *Context Mover's Barycenters* (CoMB) and related baselines on the STS-12, 13, 14 and STS-15 tasks in Tables S2 and S3 and **validation set performance** in Table S4. Hyperparameters for all the methods are tuned on STS16 (validation set), and the best configuration so obtained is used for the other STS tasks.

The first 3 baselines (NBoW, SIF, SIF + PC removed) as well as the first three CoMB (first part of the Tables) are using Glove embeddings, while methods in the second part of the table use Sent2vec embeddings. The Sent2Vec embeddings that we use are the pre-trained ones available at `https://github.com/epfml/sent2vec`. The GloVe embeddings used are the ones described in the Section S1.1. We used SIF's publicly available implementation (`https://github.com/PrincetonML/SIF`) to obtain its scores. The numbers are average Pearson correlation x 100 (with respect to ground-truth scores).

	STS12					
Model	MSRpar	MSRvid	SMTeuroparl	WordNet	SMTnews	Average
NBoW	17.5	-6.4	25.4	37.2	31.9	21.1
SIF	12.1	51.6	23.5	55.1	19.9	32.4
SIF + PC removed	21.9	58.9	30.9	55.9	37.2	41.0
Euclidean avg	31.1	67.1	45.4	52.2	32.7	45.7
CoMB (GloVe)	31.3	61.5	**47.5**	54.5	**46.0**	48.2
CoMB (GloVe) + Mix	**35.8**	75.0	44.2	59.2	38.5	**50.5**
CoMB (GloVe) + Mix + PC removed	35.5	**78.2**	35.5	**60.9**	36.5	49.3
CoMB (GloVe) + Mix + PC rem. (TBC + News Crawl)	33.0	82.8	45.7	65.9	47.0	54.9
Sent2vec	37.7	78.7	49.3	70.2	42.3	55.6
CoMB (sent2vec) + Mix	40.7	78.9	**49.9**	68.0	43.0	56.1
CoMB (sent2vec) + Mix + PC removed	**44.3**	**82.3**	47.1	**68.8**	**47.0**	**57.9**

	STS13			
Model	FNWN	Headlines	WordNet	Average
NBoW	14.2	27.1	-0.8	13.5
SIF	8.5	54.1	6.3	23.0
SIF + PC removed	13.7	61.0	75.5	50.1
Euclidean avg.	1.9	50.9	64.3	39.0
CoMB (GloVe)	11.8	54.6	60.1	42.2
CoMB (GloVe) + Mix	22.3	58.5	72.3	51.0
CoMB (GloVe) + Mix + PC removed	**28.9**	**62.8**	**77.7**	**56.5**
CoMB (GloVe)+ Mix + PC rem. (TBC + News Crawl)	46.9	75.1	79.5	67.2
Sent2vec	42.4	66.2	62.7	57.1
CoMB (sent2vec) + Mix	42.5	67.6	69.1	59.7
CoMB (sent2vec) + Mix + PC removed	**43.3**	**69.4**	**80.0**	**64.2**

Table S1: Detailed **test set performance** of *Context Mover's Barycenters* (CoMB) and related baselines on the STS12 and STS13 tasks using Toronto Book Corpus. The numbers are average Pearson correlation x100 (with respect to ground truth scores). 'Mix' denotes the mixed distributional estimate. 'PC removed' refers to removing contribution along the principal component of point estimates as done in SIF. The part in brackets after CoMB refers to the underlying ground metric.

We observe empirically that the PPMI smoothing parameter α, which balances the bias of PPMI towards rare words, plays an important role. While its ideal value would vary on each task, we found the settings mentioned in the Table S5 to work well uniformly across the above spectrum of tasks. We also provide in Table S5 a comparison of the hyper-parameters used in each of the methods in Tables S1, S2, S3 and S4.

	STS14						
Model	Forum	News	Headlines	Images	WordNet	Twitter	Average
NBoW	18.2	37.6	24.0	14.9	17.1	38.0	25.0
SIF	21.1	29.4	50.7	34.3	22.4	46.5	34.1
SIF + PC removed	27.9	43.1	57.0	52.9	76.8	53.5	51.9
Euclidean avg.	39.2	52.3	40.6	54.5	64.8	48.1	49.9
CoMB (GloVe)	40.4	**64.9**	50.5	51.5	64.4	57.8	54.9
CoMB (GloVe) + Mix	**40.9**	62.7	53.9	59.7	73.7	58.8	58.3
CoMB (GloVe) + Mix + PC removed	40.0	60.8	**58.6**	**66.6**	**77.9**	**60.8**	**60.8**
CoMB (GloVe)+ Mix + PC rem. (TBC + News Crawl)	42.0	69.8	67.1	77.4	81.6	67.0	67.5
Sent2vec	49.1	67.2	63.9	**82.5**	72.4	**75.5**	68.4
CoMB (sent2vec) + Mix	52.1	**69.5**	63.2	78.3	75.1	74.5	68.8
CoMB (sent2vec) + Mix + PC removed	**52.5**	69.5	**64.4**	78.4	**81.6**	75.3	**70.3**

Table S2: (continued from Table S1) Detailed **test set performance** of *Context Mover's Barycenters* (CoMB) and related baselines on the STS14 using Toronto Book Corpus. The numbers are average Pearson correlation x 100 (with respect to groundtruth scores). 'Mix' denotes the mixed distributional estimate. 'PC removed' refers to removing contribution along the principal component of point estimates as done in SIF. The part in brackets after CoMB refers to the underlying ground metric.

S3.2 Hyperparameters

The hyperparameters for CoMB and related baselines are summarized in the Table S5. When tuning our results on the validation set (STS16), the main hyperparameters and their ranges that we consider are: α={0.15, 0.55, 0.95}, β={0, 0.5, 1.0}, s={1, 5, 15}, λ={0.05, 0.1}, m={0.4, 0.5, 0.6}, ground metric clipping ={10, 12, 14}, sinkhorn iterations = {100}, and K={300, 400, 500}.

	STS15					
Model	Forum	Students	Belief	Headlines	Images	Average
NBoW	18.6	43.7	28.5	37.1	25.8	30.7
SIF	23.9	33.8	30.2	57.6	31.1	35.3
SIF + PC removed	35.3	63.8	51.0	62.3	51.6	52.8
Euclidean avg.	42.4	59.8	48.0	53.6	63.6	53.5
CoMB (GloVe)	36.2	64.5	45.2	61.1	61.8	53.8
CoMB (GloVe) + Mix	51.0	**66.2**	54.4	62.5	68.2	60.5
CoMB (GloVe) + Mix + PC removed	**55.3**	61.3	**63.3**	**66.1**	**74.1**	**64.0**
CoMB (GloVe) + Mix + PC rem. (TBC + News Crawl)	65.9	66.3	71.6	74.3	81.8	71.9
Sent2vec	67.5	**73.9**	**77.1**	69.4	**82.6**	**74.1**
CoMB (sent2vec) + Mix	67.9	73.8	75.6	69.8	81.5	73.7
CoMB (sent2vec) + Mix + PC removed	**68.4**	69.6	74.6	**71.3**	81.9	73.1

Table S3: (continued from Table S2) Detailed **test set performance** of *Context Mover's Barycenters* (CoMB) and related baselines on the STS15 using Toronto Book Corpus. The numbers are average Pearson correlation x 100 (with respect to groundtruth scores). 'Mix' denotes the mixed distributional estimate. 'PC removed' refers to removing contribution along the principal component of point estimates as done in SIF. The part in brackets after CoMB refers to the underlying ground metric.

	STS16					
Model	Answer	Headlines	Plagiarism	Postediting	Question	Average
NBoW	19.9	32.6	16.5	35.7	-8.9	19.2
SIF	35.1	55.1	14.6	31.7	-3.5	26.6
SIF + PC removed	42.4	60.0	58.5	71.7	55.4	57.6
Euclidean avg.	45.4	43.8	47.5	66.0	50.7	50.7
CoMB (GloVe)	38.7	55.4	50.2	67.6	50.1	52.4
CoMB (GloVe) + Mix	**50.5**	57.1	64.2	69.6	59.7	60.2
CoMB (GloVe) + Mix + PC removed	47.9	**60.6**	**70.0**	**76.6**	**59.9**	**63.0**
CoMB (GloVe) + Mix + PC rem. (TBC + News Crawl)	59.0	72.4	78.6	82.1	68.0	72.0
Sent2vec	62.5	68.3	**78.6**	82.5	53.5	69.1
CoMB (sent2vec) + Mix	62.6	69.0	76.4	83.0	59.5	70.1
CoMB (sent2vec) + Mix + PC removed	**63.3**	**69.7**	74.8	**83.9**	**61.1**	**70.6**

Table S4: Detailed **validation set performance** of *Context Mover's Barycenters* (CoMB) and related baselines on the STS16 using Toronto Book Corpus. The numbers are average Pearson correlation x100 (with respect to groundtruth scores). 'Mix' denotes the mixed distributional estimate. 'PC removed' refers to removing contribution along the principal component of point estimates as done in SIF. The part in brackets after CoMB refers to the underlying ground metric. Note that, STS16 was used as the validation set to obtain the best hyperparameters for all the methods in these experiments. As a result, high performance on STS16 may not be indicative of the overall performance.

	a			Clusters	PC removed	Mixing
SIF	$a = 10^{-4}$					
SIF + PC removed	$a = 10^{-4}$				✓	0.4
	α	β	s			
Euclidean avg.	0.55	1	5	300		
CoMB (GloVe)	0.55	1	5	300		
CoMB (GloVe) + Mix	0.95	1	1	500		0.4
CoMB (GloVe) + Mix + PC removed	0.95	1	1	500	✓	0.4
CoMB (GloVe) + Mix + PC rem. (TBC + News Crawl)	0.55	1	1	400	✓	0.6
CoMB (sent2vec) + Mix	0.15	1	1	300		0.4
CoMB (sent2vec) + Mix + PC removed	0.15	1	1	300	✓	0.4

Table S5: Detailed parameters for the methods presented in Tables S2, S2 and S4. The parameters for CoMB α, β, s denote the PPMI smoothing, column normalization exponent (Eq. (10)), and k-shift.

S4 Qualitative Analysis of Sentence Similarity

In this section, we aim to qualitatively analyse the particular examples where our method, Context Mover's Barycenters (CoMB), performs better or worse than the Smooth Inverse Frequency (SIF) approach from Arora et al. (2017).

S4.1 Evaluation Procedure

Comparing by rank. It doesn't make much sense to compare the raw distance values between two sentences as given by Context Mover's Distance (CMD) for CoMB and cosine distance for SIF. This is because the spread of distance values across sentence pairs can be quite different. Note that the quantitative evaluation of these tasks is also carried out by Pearson/Spearman rank correlation of the predicted distances/similarities with the ground-truth scores.

Thus, in accordance with this reasoning, we compare the similarity score of a sentence pair relative to its rank based on ground-truth score (amongst the sentence pairs for that dataset). So, the better method should rank sentence pairs closer to the ranking obtained via ground-truth scores.

Ground-Truth Score	*Implied meaning*
5	The two sentences are completely equivalent, as they mean the same thing.
4	The two sentences are mostly equivalent, but some unimportant details differ.
3	The two sentences are roughly equivalent, but some important information differs/missing.
2	The two sentences are not equivalent, but share some details.
1	The two sentences are not equivalent, but are on the same topic.
0	The two sentences are completely dissimilar.

Table S6: STS ground scores and their implied meanings, as taken from Agirre et al. (2015)

Ground-truth details. The ground-truth scores (can be fractional) and range from 0 to 5, and the meaning implied by the integral score values can be seen in the Table S6. In the case where different examples have the same ground-truth score, the ground-truth rank is then based on lexicographical ordering of sentences for our qualitative evaluation procedure. (This for instance means that sentence pairs ranging from 62 to 74 would correspond to the same ground-truth score of 4.6). The ranking is done in the descending order of sentence similarity, i.e., most similar to least similar.

Example selection criteria. For all the examples, we compare the best variants of CoMB and SIF on those datasets. We particularly choose those examples where there is the maximum difference in ranks according to CoMB and SIF, as they would be more indicative of where a method succeeds or fails. Nevertheless, such a qualitative evaluation is subjective and is meant to give a better understanding of things happening under the hood.

S4.2 Experiments and Observations

We look at examples from three datasets, namely: Images from STS15, News from STS14 and WordNet from STS14 to get a better idea of an overall behavior. In terms of aggregate quantitative performance, on Images and News datasets, CoMB is better than SIF, while the opposite is true for WordNet. These examples across the three datasets may not probably be exhaustive and are up to subjective interpretation, but hopefully will lend some indication as to where and why each method works.

S4.2.1 Task: STS14, Dataset: News

We look in detail at the examples in News dataset from STS 2014 (Agirre et al., 2014). The results of qualitative analysis on Images and WordNet datasets can be found in Section S4.5. For reference, CoMB results in a better performance overall with a Pearson correlation (x100) of 64.9 versus 43.0 for SIF, as presented in Table S2. The main observations are:

	Sentence 1	Sentence 2	Ground-Truth	Ground-Truth	CoMB	SIF
			Score	Ranking	Ranking	Ranking
1	the united states government and other nato members have refused to ratify the amended treaty until officials in moscow withdraw troops from the former soviet republics of moldova and georgia .	the united states and other nato members have refused ratify the amended treaty until russia completely withdraws from moldova and georgia .	4.6	30	**67**	152
2	jewish-american group the anti-defamation league (adl) published full-page advertisements in swiss and international papers in april 2008 accusing switzerland of funding terrorism through the deal .	the anti-defamation league took out full-page advertisments in swiss and international newspapers earlier in april 2008 accusing switzerland of funding terrorism through the deal .	4.4	36	**35**	128
3	the judicial order accused raghad of funding terrorism .	the court accused raghad saddam hussein of funding terrorism .	4.2	59	258	**124**
4	estonian officials stated that some of the cyber attacks that caused estonian government websites to shut down temporarily came from computers in the administration of russia including in the office of president vladimir putin .	officials in estonia including prime minister andrus ansip have claimed that some of the cyber attacks came from russian government computers including computers in the office of russian president vladimir putin .	3.8	86	**84**	206
5	the african union has proposed a peacekeeping mission to help somalia ' s struggling transitional government stabilize somalia .	the african union has proposed a peacekeeping mission to aid the struggling transitional government in stabilizing somalia , particularly after the withdrawal of ethiopian forces	3.6	119	**104**	262
6	some asean officials stated such standardization would be difficult due to different countries ' political systems .	some officials stated the task would be difficult for asean members because of varied legal and political systems .	3.6	117	244	**108**
7	nicaragua commemorated the 25th anniversary of the sandinista revolution .	nicaragua has not reconciled how to approach the anniversary of the sandinista revolution .	2.4	213	**250**	48
8	south korea launches new bullet train reaching 300 kph .	south korea has had a bullet train system since the 1980s .	2	232	**267**	130
9	south korea and israel oppose proliferation of weapons of mass destruction and an arms race .	china will resolutely oppose the proliferation of mass destructive weapons .	1.4	262	164	**235**
10	china is north korea ' s closest ally .	north korea is a reclusive state .	1.2	265	**279**	196
11	the chinese government gave active cooperation and assistance to the organization for the prohibition of chemical weapons inspections .	the ecuadorian foreign ministry said in a statement that delegates from the organization for the prohibition of chemical weapons (opaq) will also take part in the meeting .	1	277	158	**231**
12	do quy doan is a spokesman for the vietnamese ministry of culture and information .	grenell is spokesman for the u.s. mission to the united nations .	0.8	282	213	**292**

Table S7: Examples of some indicative sentence pairs, from *News* dataset in *STS14*, with ground-truth scores and ranking as obtained via (best variants of) CoMB and SIF. The total number of sentences is ***300*** and the ranking is done in descending order of similarity. The method which ranks an example closer to the ground-truth rank is better and is highlighted in **blue**. CoMB ranking is the one produced when representing sentences via CoMB and then using CMD to compare them. SIF ranking is when sentences are represented via SIF and then employing cosine similarity.

Observation 1. Examples 1, 2, 4, 5 are sentence pairs which are equivalent in meaning (cf. Table S6), but typically have additional details in the predicates of the sentences. Here, CoMB is better than SIF at ranking the pairs closer to the ground-truth ranking. This probably suggests the averaging of word embeddings, which is the 1st step in SIF, is not as resilient to the presence of such details than the Wasserstein barycenter of distributional estimates in CoMB. We speculate that when having distributional estimates (where multiple senses or contexts are considered), adding details can help towards refining the particular meaning implied.

Observation 2. Let's consider the examples 3 and 6 where SIF is better than CoMB. These are sentence pairs which are equivalent or roughly equivalent in meanings, but with a few words substituted (typically subjects) like *“judicial order"* instead of *“court"* in example 3. Here it seems that the substitution is adverse for CoMB while considering varied senses through the distributional estimate, in comparison to looking at the “point" meaning given by SIF.

Observation 3. In 7, 8, and 10, each sentence pair is about a common topic, but the meaning of individual sentences is quite different. For instance, example 8: *“south korea launches new bullet train reaching 300 kph"* & *“south korea has had a bullet train system since the 1980s"*. Or like in example 10: *“china is north korea ’ s closest ally"* & *“north korea is a reclusive state"*. Note that typically in these examples, the subject is same in a sentence pair, and the difference is mainly in the predicate. Here, CoMB identifies the difference and ranks them closer to the ground-truth. Whereas, SIF fails to understand this and ranks them as more similar (and far away) than the ground-truth.

Observation 4. The examples 9, 11, and 12 are related sentences and differ mainly in details such as the name of the country, person, department, i.e. proper nouns. In particular, consider example 9: *“south korea and israel oppose proliferation of weapons of mass destruction and an arms race"* & *“china will resolutely oppose the proliferation of mass destructive weapons"*. The main difference in these examples stems from differences in the subject rather than the predicate. CoMB considers these sentence pairs to be more similar than suggested by ground-truth. Hence, in such scenarios where the subject (like the particular proper nouns) makes the most difference, SIF seems to be better.

S4.3 Conclusions from Qualitative Examples

Summarizing the observations from the above qualitative analysis on News dataset[S7], we conclude the following about the nature of success or failures of each method.

- When the subject of the sentence is similar and main difference stems from the predicate, CoMB is the winner. This can be seen for both the case when predicates are equivalent but described distinctly *(observation 1)* and when predicates are not equivalent *(observation 3)*.

- When the predicates are similar and the distinguishing factor is in the subject (or object), SIF takes the lead. This seems to be true for both scenarios when the subject used increases or decreases the similarity as measured by CoMB, *(observations 2 and 4)*.

- The above two points in a way also signify where having distributional estimates can be better or worse than point estimates.

- CoMB and SIF appear to be complementary in the kind of errors they make. Hence, combining the two is an exciting future avenue.

Lastly, it also seems worthwhile to explore having different ground metrics for CoMB and CMD (which are currently shared). The ground metric plays a crucial role in performance and the nature of these observations. Employing a ground metric(s) that better handles the above subtleties would be a useful research direction.

[S7] Similar findings can also be seen for the two other datasets in Section S4.5.

S4.4 Effect of Sentence Length

In this section, we look at the length of sentences across all the datasets in each of the STS tasks. Average sentence length is one measure of the complexity of a particular dataset. But looking at just sentence lengths may not give a complete picture, especially for the textual similarity tasks where there can be many words common between the sentence pairs. The Table S8 shows the various statistics of each dataset, with respect to the sentence lengths along with the better method on each of them (out of CoMB and SIF).

Task-Dataset	**# sentence pairs**	**Avg. sentence length**	**Avg. word overlap (per sentence pair)**	**Avg. effective sentence length (excluding common words)**	**Better method**
STS12-MSRpar	750	21.16	14.17	6.99	CoMB
STS12-MSRvid	750	7.65	4.70	2.95	CoMB
STS12-SMTeuroparl	459	12.33	8.11	4.22	CoMB
STS12-WordNet	750	8.82	5.03	3.79	SIF
STS12-SMTnews	399	13.62	8.66	4.96	SIF
STS13-FNWN	189	22.94	2.53	20.41	CoMB
STS13-Headlines	750	7.80	3.76	4.05	SIF
STS13-WordNet	561	8.17	4.64	3.53	SIF
STS14-Forum	450	10.48	7.03	3.45	CoMB
STS14-News	300	17.42	11.59	5.83	CoMB
STS14-Headlines	750	7.91	3.89	4.01	SIF
STS14-Images	750	10.18	6.20	3.98	SIF
STS14-WordNet	750	8.87	4.83	4.05	SIF
STS14-Twitter	750	12.25	4.85	7.40	(equal)
STS15-Forum	375	17.77	4.29	13.49	CoMB
STS15-Students	750	10.70	5.33	5.37	CoMB
STS15-Belief	375	16.53	6.27	10.26	SIF
STS15-Headlines	750	8.00	3.71	4.29	SIF
STS15-Images	750	10.66	6.07	4.59	CoMB

Table S8: Analysis of sentence lengths in each of the datasets from STS12, STS13, STS14, and STS15. Along with the average sentence lengths, we also measure average word overlap in the sentence pair and thus the average *effective sentence length (i.e., after excluding the overlapping/common words in the sentence pair)*. For reference, we also show which out of CoMB or SIF performs better. On STS14-Twitter, the difference in performance isn't significant and we thus write 'equal' in the corresponding cell.

Observations.

- We notice that on datasets with longer effective sentence lengths, CoMB performs better than SIF on average. There might be other factors at play here, but if one had to pick on the axis of effective sentence length, CoMB leads over SIF[S8].
- The above statement also aligns well with the *observation 1* from the qualitative analysis (cf. Section S4.2.1), that having more details can help in refining the particular meaning or sense implied by CoMB. (Effective sentence length can serve as a good proxy for indicating the amount of details.)
- It also seems to explain why both methods don't perform well (see Table S2) on STS13-FNWN, which has on average the maximum effective sentence length (of 20.4).
- To an extent, it also points towards the effect of corpora. For instance, in a corpus such as WordNet, which has a low average sentence length and with examples typically concerned about word definitions (see

[S8] Effective sentence length averaged across datasets where CoMB is better is **7.48**. Contrast this to an average effective sentence length of **5.03** across datasets where SIF is better.

Table S10), SIF seems to be better of the methods. On the other hand, CoMB seems to be better for News (Table S7), Image captions (Table S9) or Forum.

S4.5 Additional Qualitative Analysis

S4.5.1 Task: STS15, Dataset: Images

We consider the sentence pairs from Images dataset in STS15 task (Agirre et al., 2015), as presented in Table S9. As a reminder, CoMB outperforms SIF on this dataset with a Pearson correlation (x100) of 61.8 versus 51.7, as mentioned in Table S3. The main observations are:

	Sentence 1	Sentence 2	Ground-Truth	Ground-Truth	CoMB	SIF
			Score	Ranking	Ranking	Ranking
1	the man and two young boys jump on a trampoline .	a man and two boys are bouncing on a trampoline .	4.8	68	**74**	640
2	a boy waves around a sparkler .	a young boy is twisting a sparkler around in the air .	4.4	126	**195**	624
3	a dog jumps in midair to catch a frisbee .	the brown dog jumps for a pink frisbee .	4	184	**161**	481
4	a child is walking from one picnic table to another .	the boy hops from one picnic table to the other in the park .	3.2	287	**401**	737
5	three boys are running on the beach playing a game .	two young boys and one young man run on a beach with water behind them .	3.2	306	**260**	421
6	a boy swinging on a swing .	the girl is on a swing .	2.4	380	**410**	622
7	a man is swinging on a rope above the water .	a man in warm clothes swinging on monkey bars at night .	1.6	492	259	**606**
8	a skier wearing blue snow pants is flying through the air near a jump .	a skier stands on his hands in the snow in front of a movie camera .	1.4	514	264	**605**
9	two black and white dogs are playing together outside .	two children and a black dog are playing out in the snow .	1	570	185	**372**
10	three dogs running in the dirt .	the yellow dog is running on the dirt road .	1	524	303	**531**
11	a little girl and a little boy hold hands on a shiny slide .	a little girl in a paisley dress runs across a sandy playground .	0.4	629	**683**	354
12	a little girl walks on a boardwalk with blue domes in the background .	a man going over a jump on his bike with a river in the background .	0	696	310	**591**

Table S9: Examples of some indicative sentence pairs, from *Images* dataset in *STS15*, with ground-truth scores and ranking as obtained via (best variants of) CoMB and SIF. The total number of sentences is ***750*** and the ranking is done in descending order of similarity. The method which ranks an example closer to the ground-truth rank is better and is highlighted in **blue**. CoMB ranking is the one produced when representing sentences via CoMB and then using CMD to compare them. SIF ranking is when sentences are represented via SIF and then employing cosine similarity.

Observation A. Example 1 to 5 indicate pairs of sentences which are essentially equivalent in meaning, but with varying degrees of equivalence. Here, we can see that CoMB with CMD is able to rank the similarity between these pairs quite well in comparison to SIF, even when their way of describing is different. For instance, example 2 : *“a boy waves around a sparkler”* & *“a young boy is twisting a sparkler around in the air"*. This points towards the benefit of having multiple senses or contexts encoded through the distributional estimate in CoMB.

Observation B. Next, in the examples 7 to 10, which consist of sentence pairs that are not equivalent but have commonalities (about the topic). Here, SIF ranks the sentences closer to the ground-truth ranking while CoMB interprets these pairs as being more common in meaning than given by ground-truth. This could be the consequence of comparing the various senses or contexts implied by the sentence pairs via CMD. Take for instance, example 10, *“three dogs running in the dirt"* & *“the yellow dog is running on the dirt road"*. Since these sentences are about the similar topic (and the major difference is in their subject), this can result in CMD considering them more similar than cosine distance.

Observation C. For sentences which are completely dissimilar as per ground-truth, let's look at example 11 and 12. Consider 11, which is *“a little girl and a little boy hold hands on a shiny slide"* & *“a little girl in a paisley dress runs across a sandy playground",* the sentences meaning totally different things and CoMB seems to be better at ranking than SIF. But, consider example 12: *“a little girl walks on a boardwalk with blue domes in the background"* & *“a man going over a jump on his bike with a river in the background"*. One common theme[S9] can be thought as *“a person moving with something blue in the background"*, which can result in CoMB ranking the sentence as more similar. SIF also ranks it higher (at 591) than ground-truth (696), but is more closer than CoMB which ranks it at 310.

S4.5.2 Task: STS14, Dataset: WordNet

	Sentence 1	Sentence 2	Ground-Truth Score	Ground-Truth Ranking	CoMB Ranking	SIF Ranking
1	combine so as to form a more complex product .	combine so as to form a whole ; mix .	4.6	127	**142**	335
2	(cause to) sully the good name and reputation of .	charge falsely or with malicious intent ; attack the good name and reputation of someone .	4.4	176	**235**	534
3	a person or thing in the role of being a replacement for something else	a person or thing that takes or can take the place of another .	4.2	248	**270**	535
4	create something in the mind .	form a mental image of something that is not present or that is not the case .	3.6	340	**443**	683
5	the act of surrendering an asset	the act of losing or surrendering something as a penalty for a mistake or fault or failure to perform etc .	3	405	**445**	639
6	(attempt to) convince to enroll , join or participate	register formally as a participant or member .	2.8	406	**423**	507
7	return to a prior state .	return to an original state .	4.4	219	384	**231**
8	give away something that is not needed .	give up what is not strictly needed .	4.2	261	709	**383**
9	a person who is a member of the senate .	a person who is a member of a partnership .	0.4	553	260	**429**
10	the context or setting in which something takes place .	the act of starting something .	0	717	485	**707**
11	a spatial terminus or farthest boundary of something .	a relation that provides the foundation for something .	0	620	500	**623**
12	the act of beginning something new .	the act of rejecting something .	0	670	**677**	539

Table S10: Examples of some indicative sentence pairs, from *WordNet* dataset in *STS14*, with ground-truth scores and ranking as obtained via (best variants of) CoMB and SIF. The total number of sentences is ***750*** and the ranking is done in descending order of similarity. The method which ranks an example closer to the ground-truth rank is better and is highlighted in **blue**. CoMB ranking is the one produced when representing sentences via CoMB and then using CMD to compare them. SIF ranking is when sentences are represented via SIF and then employing cosine similarity.

Lastly, we discuss the examples and observations derived from the qualitative analysis on WordNet dataset from STS14 (Agirre et al., 2014). This dataset is comprised of sentences which are the definitions of words/phrases, and sentence length is typically smaller than the datasets discussed before. For reference, SIF (76.8) does better than CoMB (64.4) in terms of average Pearson correlation (x100), as mentioned in Table S2.

Observation D. Consider examples 1 to 6 as shown in Table S10, which fall in the category of equivalent sentences but in varying degrees. The sentence pairs essentially indicate different ways of characterizing equivalent

[S9] Of course, this is upto subjective interpretation.

things. Here, CoMB is able to rank the similarity between sentences in a better manner than SIF. Specifically, see example 2: *"(cause to) sully the good name and reputation of"* & *"charge falsely or with malicious intent ; attack the good name and reputation of someone"*. It seems that SIF is not able to properly handle the additional definition present in sentence 2 and ranks this pair much lower in similarity at 534 versus 235 for CoMB. This is also in line with observation 1 about added details in the Section S4.2.1.

Observation E. The examples 7 to 9, where CoMB doesn't do well in comparison to SIF, mainly have a slight difference in the object of the sentence. For instance, in example 9: *"a person who is a member of the senate"* & *"a person who is a member of a partnership"*. So based on the kind of substituted word, looking at its various contexts via the distributional estimate can make it more or less similar than desired. In such cases, using the "point" meanings of the objects seems to fare better. This also aligns with the observations 2 and 4 in the Section S4.2.1.

S5 Sentence completion: nearest neighbor analysis

Here, we would like to qualitatively probe the kind of results obtained when computing Wasserstein barycenter of the distributional estimates, in particular, when using CoMB to represent sentences. To this end, we consider a few simple sentences and find the closest word in the vocabulary for CoMB (with respect to CMD) and contrast it to SIF with cosine distance.

Query	**CoMB** (with CMD)	**SIF** (with cosine, no PC removal)
['i', 'love', 'her']	love, hope, *always*, *actually*, *because*, *doubt*, *imagine*, *but*, *never*, *simply*	love, loved, breep-breep, *want*, clash-clash-clang, thysel, *know*, think, nope, *life*
['my', 'favorite', 'sport']	sport, *costume*, circus, *costumes*, *outfits*, super, sports, *tennis*, *brand*, fabulous	favorite, favourite, sport, wiccan-type, *pastime*, pastimes, sports, best, *hangout*, spectator
['best', 'day', 'of', 'my', 'life']	best, *for*, *also*, only, or, *anymore*, *all*, *is*, *having*, *especially*	life, day, best, c.5, writer/mummy, days, margin-bottom, time, margin-left,night
['he', 'lives, 'in', 'europe', 'for']	america, europe, *decades*, asia, *millenium*, preserve, *masters*, *majority*, elsewhere, *commerce*	lives, europe, life, america, lived, world, england, france, people, c.5
['he', 'may', 'not', 'live']	*unless*, *perhaps*, must, may, *anymore*, will, likely, youll, would, certainly	may, live, should, will, might, must, margin-left, henreeeee, 0618082132, think
['can', 'you', 'help', me', 'shopping']	*anytime*, *yesterday*, *skip*, *overnight*, *wed*, *afterward*, choosing, figuring, deciding, shopping	help, can, going, want, *go*, *do*, think, need, able, take
['he', 'likes', 'to', 'sleep', 'a', 'lot']	*whenever*, forgetting, *afterward*, *pretending*, rowan, eden, *casper*, nash, annabelle, savannah,	lot, sleep, much, *besides*, better, likes, *really*, think, *probably*, talk

Table S11: Top 10 closest neighbors for CoMB and SIF (no PC removed) found across the vocabulary, and sorted in ascending order of distance from the query sentence. Words in *italics* are those which in our opinion would fit well when added to one of the places in the query sentence. Note that, both CoMB (under current formulation) and SIF don't take the word order into account.

Observations. We find that closest neighbors (see Table S11) for CoMB consist of a relatively more diverse set of words which fit well in the context of a given sentence. For example, take the sentence "i love her", where CoMB captures a wide range of contexts, for example, "i *actually* love her", "i love her *because*", "i *doubt* her love" and more. Also for an ambiguous sentence "he lives in europe for", the obtained closest neighbors for CoMB include: 'decades', 'masters', 'majority', 'commerce' , etc., while with SIF the closest neighbors are mostly words similar to one of the query words. Further, if you look at the last three sentences in the Table S11, the first closest neighbor for CoMB even acts as a good next word for the given query. This suggests that CoMB might perform well on the task of sentence completion, but this additional evaluation is beyond the scope of this paper.

S6 Hypernymy Detection

In this Section, we provide detailed results for the hypernymy detection in Section S6.2 and mention the corresponding hyperparamters in Section S6.3. We also mention the effect of PPMI parameters on Hypernymy results in Section S6.4.

S6.1 Corpora

All the methods use a Wikipedia dump as a training corpus. In particular, GE and DIVE employWaCkypedia (a 2009 Wikipedia dump) from Baroni et al. (2009), and $D^{Hend.}$ and CMD are based on a 2015 Wikipedia dump.

S6.2 Detailed Results

	Dataset					
Method	BLESS	EVALution	LenciBenotto	Weeds	BIBLESS	Baroni
Henderson et al. ($D^{\text{Hend.}}$)	**6.4**	31.6	44.8	60.8	70.5	**78.3**
CMD (K=200) + $D^{\text{Hend.}}$	5.8	**38.1**	**50.1**	**63.9**	74.0	**67.5**
CMD (K=250) + $D^{\text{Hend.}}$	5.8	37.1	49.9	63.8	**74.9**	67.3

	Dataset				
Method	Kotlerman	Levy	HypeNet-Test	Turney	**Avg.Gain**
Henderson et al. ($D^{\text{Hend.}}$)	34.0	11.7	28.8	**56.6**	-
CMD (K=200) + $D^{\text{Hend.}}$	**34.7**	12.2	53.4	56.0	+3.2
CMD (K=250) + $D^{\text{Hend.}}$	34.4	**12.9**	**53.7**	56.3	+ **3.3**

Table S12: Comparison of the entailment vectors alone (Hend.) and when used together with our Context Mover's Distance, CMD(K) (where K is the number of clusters), in the form of ground cost $D^{\text{Hend.}}$. We also indicate the average gain in performance across these 10 datasets by using CMD along with the entailment vectors. All scores are AP at all (%).

S6.3 Hyperparameters

The above listed variants of CMD are the ones with best validation performance on HypeNet-Train (Shwartz et al., 2016). The other hyperparameters (common) for both of them are as follows:

- PPMI smoothing, $\alpha = 0.5$.
- PPMI column normalization exponent, β=0.5.
- PPMI k-shift, s=1.
- Regularization constant for Wasserstein distance, λ=0.1
- Number of Sinkhorn iterations = 500.
- Log normalization of Ground Metric.

Out of Vocabulary Details. Following Chang et al. (2017) we pushed any OOV (out-of-vocabulary) words in the test data to the bottom of the list, effectively assuming that the word pairs do not have ahypernym relation. Table S13 shows the out of vocabulary information for entailment experiments.

Dataset	Number of pairs (N)	Out of vocabulary pairs (OOV)
BLESS	26554	1504
EVALution	13675	92
LenciBenotto	5010	1172
Weeds	2928	354
BIBLESS	1668	33
Baroni	2770	37
Kotlerman	2940	172
Levy	12602	4926
HypeNet-Test	17670	11334
Turney	2188	173

Table S13: Dataset sizes. N is the number of word pairs in the dataset, and OOV denotes how many word pairs are not processed.

S6.4 Effect of PPMI parameters for Hypernymy Detection

This table was generated during an earlier version of the paper, when we were not considering the validation on HypeNet-Train. Hence, the above table doesn't contain numbers on HypeNet-Test, but an indication of performance on it can be seen in Section S12. In any case, this table suggests that our method works well for several PPMI hyper-parameter configurations.

	Dataset										
Method	BLESS	EVALution	LenciBenotto	Weeds	BIBLESS	Baroni	Kotlerman	Levy	Turney	**Avg.Gain**	**Avg. Gain (w/o Baroni)**
Henderson et al. ($D^{\text{Hend.}}$)	6.4	31.6	44.8	60.8	70.5	**78.3**	34.0	11.7	56.6	-	-
CMD (α=0.15, s=1) + $D^{\text{Hend.}}$	**7.3**	37.7	49.0	63.6	74.8	64.4	33.9	10.8	57.2	+0.5	+2.2
CMD (α=0.15, s=5) + $D^{\text{Hend.}}$	6.9	39.1	49.4	64.3	74.0	65.2	34.2	11.6	57.0	+0.8	+2.5
CMD (α=0.15, s=15) + $D^{\text{Hend.}}$	7.0	39.8	48.5	64.7	75.0	65.6	34.9	12.3	**57.3**	+1.2	**+2.9**
CMD (α=0.5, s=1) + $D^{\text{Hend.}}$	6.6	39.2	48.6	62.9	**76.1**	64.6	34.7	10.2	56.8	+0.6	+2.4
CMD (α=0.5, s=5) + $D^{\text{Hend.}}$	5.9	40.4	**49.9**	65.7	73.9	67.2	34.6	11.3	56.5	+1.2	+2.7
CMD (α=0.5, s=15) + $D^{\text{Hend.}}$	5.5	**40.5**	49.5	**66.2**	72.8	67.4	**35.6**	**12.6**	56.1	**+1.3**	+2.8

Table S14: Comparison of the entailment vectors alone (Hend.) and when used together with our Context Mover's Distance, CMD(α, s) (where α and s are the PPMI smoothing and shift parameters), in the form of ground cost $D^{\text{Hend.}}$. All of the CMD variants use $K = 100$ clusters. We observe that using our method with the entailment vectors performs better on 8 out of 9 datasets in comparison to just using these vectors alone. Avg. gain refers to the average gain in performance relative to the entailment vectors. Avg. gain w/o Baroni refers to the average performance gain excluding the Baroni dataset. The hyperparameter α refers to the smoothing exponent and s to the shift in the PPMI computation. All scores are AP at all (%).

S6.5 Computational Considerations

Table S15 presents the required time for the hypernymy evaluation task using HypEval.

Dataset	Dataset Size	Time (in seconds)
LenciBenotto	5'010	7
BIBLESS	1'668	5
EVALution	13'675	30
Weeds	2'928	5
Baroni	2'770	5
HypeNet-Train	49'475	49
HypeNet-Test	17'670	13
Turney	2'188	4
Levy	12'602	15
Kotlerman	2'940	5
Total	**110'926**	**138**

Table S15: Required evaluation time using the evaluation toolkit HypEval.

S6.6 Detection Accuracy on WBLESS

In this task, the goal is to detect whether a word pair (w_1, w_2) is in a hyponym-hypernym relationship. The detection accuracy for Henderson embeddings alone and CMD are reported in Table **??**.

Method	Accuracy (%)
Poincaré GloVe	65.2
$D_{Hend.}$	67.7
CMD $(K = 200) + D_{Hend.}$	**75.4**
CMD $(K = 250) + D_{Hend.}$	75.2

Table S16: Detection accuracy on WBLESS. CMD performs better than the state-of-the-art fully unsupervised method as reported in Tifrea et al. (2018).

Method	Spearman Correlation
Poincaré GloVe	0.341
$D_{Hend.}$	0.316
CMD $(K = 200) + D_{Hend.}$	0.336
CMD $(K = 250) + D_{Hend.}$	0.338

Table S17: Spearman correlation on HyperLex Vulić et al. (2017)

S7 Qualitative Analysis of Hypernymy detection

Here, our objective is to qualitatively analyse the particular examples where our method of using Context Mover's Distance (CMD) along with embeddings from Henderson (2017) performs better or worse than just using these entailment embeddings alone.

S7.1 Evaluation Procedure

Comparing by rank. Again as in the qualitative analysis with sentence similarity, it doesn't make much sense to compare the raw distance/similarity values between two words as their spread across word pairs can be quite different. We thus compare the ranks assigned to each word pair by both the methods.

Ground-truth details. In contrast to graded ground-truth scores in the previous analysis, here we just have a binary ground truth: 'True' if the hyponym-hypernym relation exists and 'False' when it doesn't. We consider the BIBLESS dataset (Kiela et al., 2015) for this analysis, which has a total of 1668 examples. Out of these, 33 word pairs are not in the vocabulary (see Table S13), so we ignore them for this analysis. Amongst the 1635 examples left, 814 are 'True' and 821 are 'False'. A perfect method should rank the examples labeled as 'True' from 1 to 814 and the 'False' examples from 815 to 1635. Of course, achieving this is quite hard, but the better of the methods should rank as many examples in the desired ranges.

Example selection criteria. We look at the examples where the difference in ranks as per the two methods is the largest. Also, for a few words, we also look at how each method ranks when present as a hypernym and a hyponym. If the difference in ranks is defined as, *CMD rank - Henderson Rank*, we present the top pairs where this difference is most positive and most negative.

S7.2 Results

For reference on the BIBLESS dataset, CMD performs better than Henderson embeddings quantitatively (cf. Table 2). Let's take a look at some word pairs to get a better understanding.

S7.2.1 Maximum Positive Difference in Ranks

These are essentially examples where CMD considers the entailment relation as 'False' while the Henderson embeddings predict it as 'True', and both are most certain about their decisions. Table S18 shows these pairs, along with ranks assigned by the two methods and the ground-truth label for reference.

Some quick observations: many of the word pairs that the Henderson's method gets wrong are co-hyponym pairs, such as: ('banjo', 'flute'), ('guitar', 'trumpet'), ('turnip, 'radish'). Additionally, ('bass', 'cello'), ('creature', 'gorilla'), etc., are examples where the method has to assess not just if the relation exists, but also take into account the directionality between the pair, which the Henderson's method seems unable to do.

S7.2.2 Maximum Negative Difference in Ranks

Now the other way around, these are examples where CMD considers the entailment relation as 'True' while the Henderson embeddings predict it as 'False', and both are most certain about their decisions. Table S19 shows these pairs. The examples where CMD performs poorly like, ('box', 'mortality'), ('pistol', 'initiative') seem to be unrelated and we speculate that matching the various contexts or senses of the distributional estimate causes this behavior. One possibility to deal with this can be to take into account the similarity between word pairs in the ground metric. Overall, CMD does a good job of handling these pairs in comparison to the Henderson method.

Hypernym candidate	Hypernym candidate	Ground Truth	CMD rank	Henderson rank	Better Method
bass	cello	FALSE	1346	56	CMD
banjo	flute	FALSE	1312	108	CMD
guitar	trumpet	FALSE	1249	52	CMD
trumpet	violin	FALSE	1351	165	CMD
gill	goldfish	FALSE	1202	21	CMD
topside	battleship	FALSE	1508	345	CMD
trumpet	piano	FALSE	1289	126	CMD
washer	dishwasher	FALSE	1339	234	CMD
gun	pistol	FALSE	1270	166	CMD
cauliflower	rainbow	FALSE	1197	136	CMD
hawk	woodpecker	FALSE	1265	210	CMD
garlic	spice	TRUE	1248	204	Henderson
coyote	beast	TRUE	1096	57	Henderson
lizard	beast	TRUE	1231	201	Henderson
turnip	radish	FALSE	1060	39	CMD
creature	gorilla	FALSE	1558	543	CMD
rabbit	squirrel	FALSE	1260	249	CMD
ship	battleship	FALSE	1577	571	CMD
giraffe	beast	TRUE	1220	220	Henderson
coyote	elephant	FALSE	1017	28	CMD

Table S18: The top 20 word pairs with **maximum positive difference** in ranks (CMD rank - Henderson rank). The rank is given out of 1635.

Hyponym candidate	Hypernym candidate	Ground Truth	CMD rank	Henderson rank	Better Method
box	mortality	FALSE	116	1534	Henderson
radio	device	TRUE	110	1483	CMD
television	system	TRUE	5	1354	CMD
elephant	hospital	FALSE	52	1355	Henderson
pistol	initiative	FALSE	40	1316	Henderson
library	construction	TRUE	71	1335	CMD
radio	system	TRUE	6	1266	CMD
bowl	artifact	TRUE	223	1448	CMD
oven	device	TRUE	88	1279	CMD
bear	creature	TRUE	324	1513	CMD
stove	device	TRUE	167	1356	CMD
saw	tool	TRUE	461	1620	CMD
television	equipment	TRUE	104	1244	CMD
library	site	TRUE	87	1217	CMD
battleship	bus	FALSE	292	1418	Henderson
pistol	device	TRUE	70	1187	CMD
battleship	vehicle	TRUE	77	1175	CMD
bowl	container	TRUE	333	1431	CMD
pub	construction	TRUE	19	1116	CMD
bowl	object	TRUE	261	1334	CMD

Table S19: The top 20 word pairs with **maximum negative difference** in ranks (CMD rank - Henderson rank). The rank is given out of 1635.