# Wasserstein is all you need

Anonymous EMNLP submission

## Abstract

We propose a unified framework for building unsupervised representations of individual objects or entities (and their compositions), by associating with each object both a distributional as well as a point estimate (vector embedding). This is made possible by the use of *optimal transport*, which allows us to build these associated estimates while harnessing the underlying geometry of the ground space. Our method gives a novel perspective for building rich and powerful feature representations that simultaneously capture uncertainty (via a distributional estimate) and interpretability (with the optimal transport map). As a guiding example, we formulate unsupervised representations for text, in particular for sentence representation and entailment detection. Empirical results show strong advantages gained through the proposed framework. This approach can be used for any unsupervised or supervised problem (on text or other modalities) with a co-occurrence structure, such as any sequence data. The key tools underlying the framework are Wasserstein distances and Wasserstein barycenters[1].

## 1 Introduction

One of the main driving factors behind the recent surge of interest and successes in natural language processing and machine learning has been the development of better representation methods for data modalities. Examples include continuous vector representations for language (Mikolov et al., 2013; Pennington et al., 2014), convolutional neural network (CNN) based text representations (Kim, 2014; Kalchbrenner et al., 2014; Severyn and Moschitti, 2015; Deriu et al., 2017), or via other neural architectures such as RNNs, LSTMs (Hochreiter and Schmidhuber, 1997; Collobert and Weston, 2008), all sharing one core idea – to map input entities to dense vector embeddings lying in a low-dimensional latent space where the semantics of the inputs are preserved.

While existing methods represent each entity of interest (e.g., a word) as a single point in space (e.g., its embedding vector), we here propose a fundamentally different approach. We represent each entity based on the *histogram of contexts* (co-occurring with it), with the contexts themselves being points in a suitable metric space. This allows us to cast the distance between histograms associated with the entities as an instance of the *optimal transport problem* (Monge, 1781; Kantorovich, 1942; Villani, 2008). For example, in the case of words as entities, the resulting framework then intuitively seeks to minimize the cost of moving the set of contexts of a given word to the contexts of another. Note that the contexts here can be words, phrases, sentences, or general entities co-occurring with our objects to be represented, and these objects further could be any type of events extracted from sequence data, including e.g., products such as movies or web-advertisements (Grbovic et al., 2015), nodes in a graph (Grover and Leskovec, 2016), or other entities (Wu et al., 2017). Any co-occurrence structure will allow the construction of the histogram information, which is the crucial building block for our approach.

A strong motivation for our proposed approach here comes from the domain of natural language, where the entities (words, phrases or sentences) generally have multiple semantics under which they are present. Hence, it is important that we consider representations that are able to effectively capture such inherent uncertainty and polysemy, and we will argue that histograms (or probability distributions) over embeddings allows to capture more of this information compared to point-wise embeddings alone. We will call the histogram as

[1] And, hence the title!

the *distributional estimate* of our object of interest, while we refer to the individual embeddings of single contexts as *point estimates*.

Next, for the sake of clarity, we discuss the framework in the concrete use-case of text representations, when the contexts are just words, by employing the well-known Positive Pointwise Mutual Information (PPMI) matrix to compute the histogram information for each word.

With the power of optimal transport, we show how this framework can be of significant use for a wide variety of important tasks in NLP, including word and sentence representations as well as hypernymy (entailment) detection, and can be readily employed on top of existing pre-trained embeddings for the contexts. The connection to optimal transport at the level of words and contexts paves the way to make better use of its vast toolkit (like Wasserstein distances, barycenters, etc.) for applications in NLP, which in the past has primarily been restricted to document distances (Kusner et al., 2015; Huang et al., 2016).

We demonstrate that building the required histograms comes at almost no additional cost, as the co-occurrence counts are obtained in a single pass over the corpus. Thanks to the entropic regularization introduced by Cuturi (2013), Optimal Transport distances can be computed efficiently in a parallel and batched manner on GPUs. Lastly, the obtained transport map (Figure 1) also provides for interpretability of the suggested framework.

## 2 Related Work

Most of the previous work in building representations for natural language has been focused towards vector space models, in particular, popularized through the groundbreaking work in Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). The key idea in these models has been to map words which are similar in meaning to nearby points in a latent space. Based on which, many works (Levy and Goldberg, 2014a; Melamud et al., 2015; Bojanowski et al., 2016) have suggested specializing the embeddings to capture some particular information required for the task at hand. One of the problems that still persists is the inability to capture, within just a point embedding, the various semantics and uncertainties associated with the occurrence of a particular word (Huang et al., 2012; Guo et al., 2014).

A recent line of work has proposed the view to represent words with Gaussian distributions or mixtures of Gaussian distributions (Vilnis and McCallum, 2014b; Athiwaratkun and Wilson, 2017), or hyperbolic cones (Ganea et al., 2018) for this purpose. Also, a concurrent work from Muzellec and Cuturi (2018) has suggested using elliptical distributions endowed with a Wasserstein metric. While these already provide richer information than typical vector embeddings, their form restricts what could be gained by allowing for arbitrary distributions. In addition, hyperbolic embeddings (Nickel and Kiela, 2017; Ganea et al., 2018) are so far restricted to supervised tasks (and even elliptical embeddings (Muzellec and Cuturi, 2018) to a most extent), not allowing unsupervised representation learning as in the focus of the paper here. To this end, we propose to associate with each word a distributional and a point estimate. These two estimates together play an important role and enable us to make use of optimal transport.

Amongst the few explorations of optimal transport in NLP, i.e., document distances (Kusner et al., 2015; Huang et al., 2016), document clustering (Ye et al., 2017), bilingual lexicon induction (Zhang et al., 2017), or learning an orthogonal Procrustes mapping in Wasserstein distance (Grave et al., 2018), the focus has been on transporting words directly. For example, the Word Mover's Distance (Kusner et al., 2015) casts finding the distance between documents as an optimal transport problem between their bag of words representation. Our approach is different as we consider the transport over contexts instead, and use it to propose a representation for words. This enables us to establish any kind of distance (even asymmetric) between words by defining a suitable underlying cost on the movement of contexts, as we show for the case of entailment. Another benefit of defining this transport over contexts is the added flexibility to extend the representation for sentences (or arbitrary length text) by utilizing the idea of Wasserstein barycenters, which to the best of our knowledge has never been considered in the past.

Lastly, the proposed framework is not specific to words or sentences but holds for building unsupervised representations for any entity and composition of entities, where a co-occurrence structure can be devised between entites and their contexts.

## 3 Background on Optimal Transport

Optimal Transport (OT) provides a way to compare two probability distributions defined over a space $\mathcal{G}$, given an underlying distance on this space (or more generally a cost of moving one point to another). In other terms, it lifts distance between points to distance between distributions. Below, we give a short yet formal background description on optimal transport for the discrete case.

Let's consider an empirical probability measure of the form $\mu = \sum_{i=1}^{n} a_i \delta(x_i)$ where $X = (x_1, \ldots, x_n) \in \mathcal{G}^n$, $\delta(x)$ denotes the Dirac (unit mass) distribution at point $x \in \mathcal{G}$, and $(a_1, \ldots, a_n)$ lives in the probability simplex $\Sigma_n := \left\{p \in \mathbb{R}_+^n \mid \sum_{i=1}^{n} p_i = 1\right\}$.

Now consider a second empirical measure, $\nu = \sum_{j=1}^{m} b_j \delta(y_j)$, with $Y = (y_1, \ldots, y_m) \in \mathcal{G}^m$, and $(b_1, \ldots, b_m) \in \Sigma_m$. If the ground cost of moving from point $x_i$ to $y_j$ is denoted by $M_{ij}$, then the Optimal Transport distance between $\mu$ and $\nu$ is the solution to the following linear program.

$$\mathrm{OT}(\mu, \nu; M) := \min_{T \in \mathbb{R}_+^{n \times m}} \sum_{ij} T_{ij} M_{ij}$$

$$\text{such that } \forall i, \sum_j T_{ij} = a_i, \quad \forall j, \sum_i T_{ij} = b_j.$$

Here, the optimal $T \in \mathbb{R}^{n \times m}$ is referred to as the *transportation matrix*: $T_{ij}$ denotes the optimal amount of mass to move from point $x_i$ to point $y_i$. Intuitively, OT is concerned with the problem of moving goods from factories to shops in such a way that all the demands are satisfied and the overall transportation cost is minimal.

When $\mathcal{G} = \mathbb{R}^d$ and the cost is defined with respect to a metric $D_{\mathcal{G}}$ over $\mathcal{G}$ (i.e., $M_{ij} = D_{\mathcal{G}}(x_i, y_j)^p$ for any $i, j$), OT defines a distance between empirical probability distributions. This is the $p$-Wasserstein distance, defined as $W_p(\mu, \nu) := \mathrm{OT}(\mu, \nu; D_{\mathcal{G}}^p)^{1/p}$. In most cases, we are only concerned with the case where $p = 1$ or 2.

The cost of exactly solving OT problem scales at least in $\mathcal{O}(n^3 \log(n))$ ($n$ being the cardinality of the support of the empirical measure) when using network simplex or interior point methods. Following Cuturi (2013) we consider the entropy regularized Wasserstein distance, $W_p^\lambda(\mu, \nu)$. The above problem can then be solved efficiently using Sinkhorn iterations, albeit at the cost of some approximation error. The regularization strength $\lambda \geq 0$ controls the accuracy of approximation and recovers the true OT for $\lambda = 0$. The cost of the Sinkhorn algorithm is only quadratic in $n$ at each iteration.

Further on in our discussion, we will make use of the notion of averaging in the Wasserstein space. More precisely the Wasserstein barycenter, introduced by Agueh and Carlier (2011), is a probability measure that minimizes the sum of ($p$-th power) Wasserstein distances to the given measures. Formally, given $N$ measures $\{\nu_1, \ldots, \nu_N\}$ with corresponding weights $\eta = \{\eta_1, \ldots, \eta_N\} \in \Sigma_N$, the Wasserstein barycenter can be written as follows:

$$B_p(\nu_1, \ldots, \nu_N) = \arg\min_{\mu} \sum_{i=1}^{N} \eta_i W_p(\mu, \nu_i)^p. \tag{2}$$

We similarly consider the regularized barycenter $B_p^\lambda$, using entropy regularized Wasserstein distances $W_p^\lambda$ in the above minimization problem, following Cuturi and Doucet (2014). Employing the method of iterative Bregman projections (Benamou et al., 2015), we obtain an approximation of the solution at a reasonable computational cost.

## 4 Methodology

In this section, we elaborate on both the distributional and the point estimate that we attach to each word, as mentioned in the introduction. A common method in NLP to empirically estimate the probability $p(w|c)$ of occurrence of a word $w$ in some context $c$, is to compute the number of times the word $w$ co-occurs with context $c$ relative to the total number of times context $c$ appears in the corpus. The context $c$ could be a particular word, phrase, sentence or other definitions of co-occurrence of interest.

**Distributional Estimate.** For a word $w$, its distributional estimate is built from a histogram over the set of contexts $\mathcal{C}$, and an embedding of these contexts into a space $\mathcal{G}$.

A natural way to build this histogram is to maintain a co-occurrence matrix between words in our vocabulary and all possible contexts, such that its each entry indicates how often a word and context occur in an interval (or window) of a fixed size $L$. Then, the bin values $((H^w)_c)_{c \in \mathcal{C}}$ of the histogram $(H^w)$ for a word $w$, can be viewed as the row corresponding to $w$ in this co-occurrence matrix. In Section 5, we discuss how to reduce the number of bins in the histogram, and possible modifications of the co-occurrence matrix to improve associations.

The simplest embedding of contexts is into the space of one-hot vectors of all the possible contexts. However, this induces a lot of redundancy in the representation and the distance between contexts does not reflect their semantics. A classical solution would be to instead find a dense low-dimensional embedding of contexts that captures the semantics, possibly using techniques such as SVD or deep neural networks. We denote by $V = (\mathbf{v}_c)_{c \in \mathcal{C}}$ an embedding of the contexts into this low-dimensional space $\mathcal{G} \subset \mathbb{R}^d$, which we refer to as the *ground space*. (We will consider prototypical cases of how this metric can be obtained in Sections 6 and 7.)

Combining the histogram $H^w$ and the embedding $V$, we represent the word $w$ by the following empirical distribution:

$$\mathbb{P}_V^w := \sum_{c \in \mathcal{C}} (\mathrm{H}^w)_c \, \delta(\mathbf{v_c}). \tag{3}$$

Recall that $\delta(\mathbf{v_c})$ denotes the Dirac measure at the position $\mathbf{v_c}$ of the context $c$. We refer to this representation (Eq. (3)) as the *distributional estimate* of the word.

Together with its distributional estimate, the word $w$ also has an associated *point estimate* $\mathbf{v}_w$ when it occurs in the sense of a context, in the form of its position (or embedding) in the ground space. This is what we mean by attaching the distributional and point estimate to each word.

**Distance.** If we equip the ground space $\mathcal{G}$ with a meaningful metric $D_{\mathcal{G}}$, then we can subsequently define a distance between the representations of two words $w_i$ and $w_j$, as the solution to the following optimal transport problem:

$$\mathrm{OT}(\mathbb{P}_V^{w_i}, \mathbb{P}_V^{w_j}; D_{\mathcal{G}}^p) \simeq W_p^{\lambda}(\mathbb{P}_V^{w_i}, \mathbb{P}_V^{w_j})^p. \tag{4}$$

Intuitively, two words are similar in meaning if the contexts of one word can be easily or cheaply transported to the contexts of the other word, with this ease of transportation being measured by $D_{\mathcal{G}}$. This idea still remains in line with the distributional hypothesis (Harris, 1954; Rubenstein and Goodenough, 1965) that words in similar contexts have similar meanings, but provides a unique way to quantify it.

**Interpretation.** In fact, both of these estimates are closely tied together and required to serve as an effective representation. For instance, if we only have the distributional estimates, then we may have

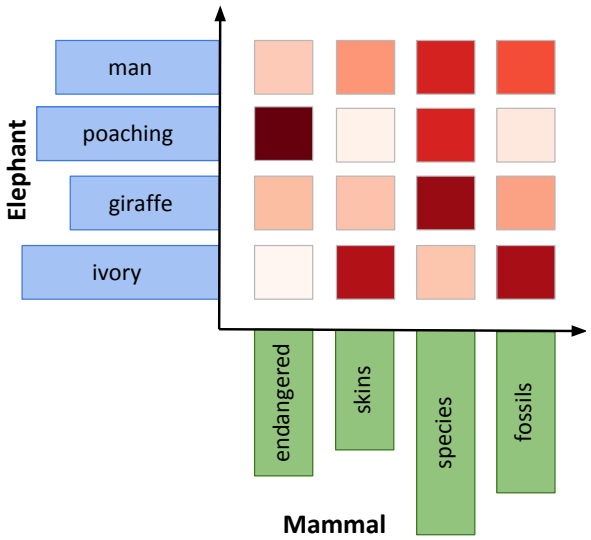


Figure 1: *Illustration of the optimal transport between the histograms of elephant and mammal.* Here, we pick four contexts at random from a list of top 20 contexts (in terms of PPMI) for the two histograms. Then using the regularized Wasserstein distance (as in Eq. (4)), we plot the obtained transportation matrix (or commonly called transport map) $T$ as above. Note how 'ivory' adjusts its movement towards 'skin' (as in skin color) to allow 'poaching' to be easily moved to 'endangered' as going to other contexts of 'mammal' is costly for 'poaching', thus capturing a global perspective.

two words such as '*tennis*' and '*football*' which occur in the contexts of {*court, penalty, judge*} and {*stadium, foul, referee*} respectively. While these contexts are mutually disjoint, they are quite close in meaning. Now there could be a third word such as '*law*' which occurs in the exact same contexts as tennis. So considering the distributional estimate alone, without making use of the point estimates of context, would lead us to have a smaller distance between tennis and law as compared to tennis and football. Whereas, if we only considered the point estimates, then we would lose much of the uncertainty associated about the contexts in which they occur, except for maybe the restricted information of neighboring points in the ground space. This is made clear in a related illustration shown in Figure 2.

The family of problems where such a representation can be used is not restricted to entities pertaining to NLP: the framework can be similarly used in any domain where a co-occurrence structure exists between entities and their contexts. For instance, in the case of movie recommendation where users correspond to the entities and movies to the contexts. Lastly, this connection with optimal transport

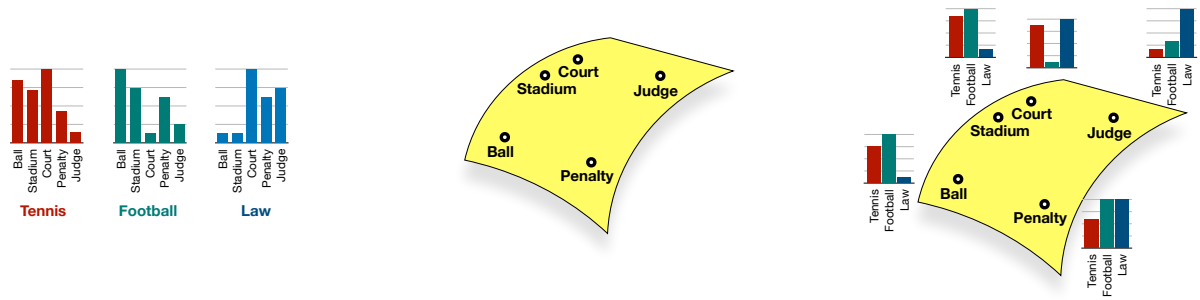


Figure 2: *Illustration of three words, each with their distributional estimates* (left)*, as well as the point estimates of the relevant contexts* (middle)*, as well as joint representation* (right).

allows us to utilize its rich theoretical and algorithmic toolkit towards important problems in NLP. In the next section, we discuss a concrete framework of how this can be applied and in Section 6 and 7, we detail how the tasks of sentence representation and hypernymy detection can be effectively carried out with this framework.

## 5 Concrete Framework

For the sake of brevity, we present the framework for the case where contexts consist of single words.

**Making associations better.** Let's say that a word is considered to be a context word if it appears in a symmetric window of size $L$ around the target word (the word whose distributional estimate we seek). Now, the co-occurrence matrix is between the words of our vocabulary, with rows and columns indicating target words and context words respectively. While each entry of this matrix reflects the co-occurrence count, it may not suggest a strong association between the target and its context. For instance, in the sentence *"She prefers her coffee to be brewed fast than being perfect"*, there is a stronger association between 'coffee' and 'brewed' rather than between 'coffee' and 'her', although the co-occurrence counts alone might imply the opposite. Hence, to handle this we consider the well-known Positive Pointwise Mutual Information (PPMI) matrix (Church and Hanks, 1990; Levy et al., 2015), whose entries are as follows:

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

The PPMI entries are non-zero when the joint probability of the target and context words co-occurring together is higher than the probability when they are independent. Typically, these probabilities are estimated from the co-occurrence counts $\#(w,c)$ in the corpus and lead to

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

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[2] the context probabilities by raising them to an exponent of $\alpha$ lying between 0 and 1. Levy and Goldberg (2014b) have also suggested the use of the shifted PPMI (SPPMI) matrix where the shift[3] 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 define

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

Hence, the bin values for our histogram in Eq. (3) are formed as:

$$(H^w)_c := \frac{\text{SPPMI}_s^\alpha(w,c)}{\sum_{c\in\mathcal{C}}\text{SPPMI}_s^\alpha(w,c)}. \qquad (5)$$

**Computational considerations.** The view of optimal transport between histograms of contexts introduced in Eq. (4) offers a pleasing interpretation (see Figure 1). However, it might still be a computationally intractable in its current formulation. Indeed 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

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

[3] 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 $k$.

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 the word histograms still varies from $10^3$ to $5 \times 10^4$ context words, when considering a vocabulary of size around $2 \times 10^5$.

This is a problem because the Sinkhorn algorithm for regularized optimal transport (Cuturi, 2013) (Section 3), scales roughly quadratically in the histogram size and the ground cost matrix can also become prohibitive to store in memory, for the range of histogram sizes mentioned. One possible fix is to instead consider a few representative contexts in this ground space. The hope is that with the 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 $D_{\mathcal{G}}$. Besides the computational gain, the clustering will lead us to consider this transport between more abstract contexts. This will although come at the loss of some interpretability. Another alternative for dealing with this computational issue could be to consider stochastic optimal transport techniques (Genevay et al., 2016), where the intuition would be to randomly sample a subset of contexts while considering this transport. But we leave that direction for a future work.

Now, consider that we have obtained $K$ contexts, each representing 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 $\tilde{\mathbb{P}}^w_{\tilde{V}} = \sum_{k=1}^{K} (\tilde{H}^w)_k \, \delta(\tilde{\mathbf{v}}_k)$. Here $\tilde{\mathbf{v}}_k \in \tilde{V}$ denotes the point estimate of the $k^{th}$ representative context, and $(\tilde{H}^w)_k$ are the new bin values for the histogram similar to that in Eq. (5), but with respect to these parts,

$$(\tilde{H}^w)_k := \frac{\text{SPPMI}^\alpha_s(w, \mathcal{C}_k)}{\sum_{k=1}^{K} \text{SPPMI}^\alpha_s(w, \mathcal{C}_k)}, \text{with}$$
$$\text{SPPMI}^\alpha_s(w, \mathcal{C}_k) := \sum_{c \in \mathcal{C}_k} \text{SPPMI}^\alpha_s(w, c).$$

Furthermore, in certain cases[4], it can be important to measure the relative portion of $\mathcal{C}_k$'s SPPMI (Eq. (7)) that has been used towards a word $w$. Otherwise the process of making the histogram unit sum in Eq. (6) will misrepresent the actual underlying contribution (check Eq. (10) in Appendix A for more details):

$$(\tilde{H}^w)_k := \frac{(\bar{H}^w)_k}{\sum_{k=1}^{K} (\bar{H}^w)_k} \quad \text{with} \tag{6}$$
$$(\bar{H}^w)_k := \frac{\text{SPPMI}^\alpha_s(w, \mathcal{C}_k)}{\sum_w \text{SPPMI}^\alpha_s(w, \mathcal{C}_k)}. \tag{7}$$

**Summary.** While we detailed the case of context as single words, this framework can be extended in a similar manner to take into account other contexts such as bi-grams, tri-grams, n-grams or other abstract semantic concepts. Building this suggested representation comes at almost free cost during the typical learning of point-estimates for an NLP task, as the co-occurrence counts can simply be maintained while going through the corpus. GloVe (Pennington et al., 2014) even constructs the co-occurrence matrix explicitly as a precursor to learning the point-estimates.

## 6 Sentence Representation

Traditionally, the goal of this task is to develop a representation for sentences, that captures the semantics conveyed by it. Most unsupervised representations proposed in the past rely on the composition of vector embeddings for the words, through either additive, multiplicative, or other ways (Mitchell and Lapata, 2008; Arora et al., 2017; Pagliardini et al., 2017). We propose to represent sentences as probability distributions to better capture the inherent uncertainty and polysemy.

Our belief is that the meaning of a sentence can be understood as a concept that best explains the simultaneous occurrence of the words in it. We hypothesize that a sentence, $S = (w_1, w_2, \ldots, w_N)$, can be efficiently represented via the Wasserstein barycenter (see Eq. (2)) of distributional estimates of the words in the sentence, i.e.,

$$\tilde{\mathbb{P}}_S := B^\lambda_p \left( \tilde{\mathbb{P}}^{w_1}_V, \tilde{\mathbb{P}}^{w_2}_V, \ldots, \tilde{\mathbb{P}}^{w_N}_V \right), \tag{8}$$

which is itself again a distribution over $\mathcal{G}$.

Yet another interesting property is the non-associativity[5] of the barycenter operation. This can be utilized to take into account the order of the words in a sentence. For now, we restrict our focus on exploring how well barycenters of words taken all at once can represent sentences and this direction is left for future work.

Interestingly, the classical weighted averaging of point-estimates (Arora et al., 2017) can be seen as

[4] when the SPPMI contributions towards the partitions (or clusters) have a large variance.

[5] $B_p(\mu, B_p(\nu, \xi)) \neq B_p(B_p(\mu, \nu), \xi)$.

| | Dataset | | | | |
|---|---|---|---|---|---|
| Model | STS12 | STS13 | STS14 | STS15 | STS16 |
| BOW | 22.1 | 18.8 | 27.2 | 29.1 | 21.4 |
| SIF (with no PC removed) | 32.9 | 21.4 | 33.4 | 37.8 | 22.7 |
| SIF (with 1st PC removed) | 34.4 | **43.0** | 45.2 | **48.1** | 41.2 |
| WB ($K$=250) | 43.3 | 35.3 | 45.2 | 45.0 | 42.0 |
| WB ($K$=300) | **44.3** | 35.6 | **45.7** | 46.4 | **43.2** |

Table 1: Performance on the STS tasks using $K$=250 and $K$=300 clusters (Avg. Pearson correlation x 100).

a special case of Wasserstein barycenter, when the distribution associated to a word is just a Dirac at its point estimate. It becomes apparent that having a rich distributional estimate for a word can be advantageous.

**Evaluation.** To validate Wasserstein Barycenters (WB) as effective sentence representations, we consider the task of Semantic Textual Similarity (STS) (Agirre et al., 2012, 2013, 2014, 2015, 2016). The objective here is to predict how similar or dissimilar are two sentences in their meanings. Since with barycenter representation as in Eq. (8), each sentence is also a histogram of contexts, we can again make use of optimal transport to define the distance between two sentences $S_1$ and $S_2$,

$$\mathrm{OT}(\tilde{\mathbb{P}}_V^{S_1}, \tilde{\mathbb{P}}_V^{S_2}; D_{\mathcal{G}}^p) \simeq W_p^{\lambda}(\tilde{\mathbb{P}}_V^{S_1}, \tilde{\mathbb{P}}_V^{S_2})^p.$$

As a ground metric, we consider the Euclidean distance between the point estimates of words. This point estimate for a word is its embedding in the context space and can be obtained with the help of Word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). For, this task we train the word embeddings on the Toronto Book Corpus (Kiros et al., 2015) via GloVe and in the process also gain the distributional estimates of words for free. Since the word embeddings in these methods are constructed so that similar meaning words are close in cosine similarity, we find the representative points by performing K-means clustering with respect to this similarity.

We benchmark our performance against SIF (Smooth Inverse Frequency) method from Arora et al. (2017) who regard it as a "simple but tough-to-beat baseline", as well as against the common Bag of Words (BoW) averaging. For this experiment, we use SIF's publicly available implementation[6] and perform the evaluation using SentEval (Conneau and Kiela, 2018). Table 1 shows that we always beat BoW and SIF with weighted averaging on all tasks. Further, we perform better than the best variant of SIF (which in addition removes the 1st principal component) on 3 out of 5 tasks. Also, on the other two tasks we still perform competitively and achieve an overall gain over their best variant with $K = 300$ clusters (refer to Table 5 in Appendix A for detailed results). Note that, the hyperparameters for SIF are taken to be the best ones separately for each task. Whereas we used the same set of hyperparameters for all the above tasks and the PPMI specific hyperparameters haven't been tuned much, but this should not give us an edge over them.

In our comparison, we do not include methods such as Sent2vec (Pagliardini et al., 2017), as they are specifically trained to work well on the given task of sentence representation, and such an approach for training remains outside the scope of current work. Our approach for representing barycenters does not require any additional training and is still able to match and outperform strong baselines for the task of semantic similarity. This highlights the efficacy of proposed representation.

## 7 Hypernymy Detection

In linguistics, hypernymy is a relation between words (or sentences) where the semantics of one word (the *hyponym*) are contained within that of another word (the *hypernym*). A simple form of this relation is the *is-a* relation, e.g., *cat* is an *animal*. Hypernymy is a special case of the more general concept of lexical entailment which may be broadly defined as any semantic relations between two lexical items where the meaning of one is implied by the meaning of the other. Detecting lexical entailment relations is relevant for numerous tasks in NLP. Given a database of lexical entailment relations, e.g., containing *Roger Federer* is a *tennis player* might help a question answering system an-

[6] https://github.com/PrincetonML/SIF

| Method | Dataset | | | | | |
|---|---|---|---|---|---|---|
| | BLESS | EVALution | LenciBenotto | Weeds | Henderson | Baroni |
| Henderson. et. al | 6.4 | 31.6 | 44.8 | 60.8 | 70.5 | **78.3** |
| WE ($\alpha$=0.15, $s$=15) | **7.0** | 39.8 | 48.5 | 64.7 | **75.0** | 65.6 |
| WE ($\alpha$=0.5, $s$=15) | 5.5 | **40.5** | **49.5** | **66.2** | 72.8 | 67.4 |

Table 2: Comparison between entailment vectors and optimal transport / Wasserstein based entailment measure (WE). The scores are AP@all (%). The hyperparameter $\alpha$ refers to the smoothing exponent and $s$ to the shift in the PPMI computation. More datasets are presented in Table 4 in the Appendix A.

| Method | Dataset | | | | |
|---|---|---|---|---|---|
| | EVALution | LenciBenotto | Weeds | Turney | Baroni |
| GE + C | 26.7 | 43.3 | 52.0 | 53.9 | 69.7 |
| GE + KL | 29.6 | 45.1 | 51.3 | 52.0 | 64.6 |
| DIVE + C·$\Delta$S | 33.0 | **50.4** | 65.5 | 57.2 | **83.5** |
| Henderson. et. al | 31.6 | 44.8 | 60.8 | 56.6 | 78.3 |
| WE ($\alpha$=0.15, $s$=15) | 39.8 | 48.5 | 64.7 | **57.3** | 65.5 |
| WE ($\alpha$=0.5, $s$=15) | **40.5** | 49.5 | **66.2** | 56.1 | 67.4 |

Table 3: Comparison between entailment vectors, optimal transport / Wasserstein based entailment measure (WE) and other state-of-the-art methods. GE+C and GE+KL are Gaussian embeddings with cosine similarity and negative KL-divergence. The scores for GE+C, GE+KL, and DIVE + C·$\Delta S$ are taken from (Chang et al., 2017) as we use the same evaluation setup. The scores are again AP@all (%).

swering the question *"Who is Switzerland's most successful tennis player?"*.

First distributional approaches to detect hyponymy were unsupervised and exploited different linguistic properties of hypernymy (Weeds and Weir, 2003; Kotlerman et al., 2010; Santus et al., 2014; Rimell, 2014). While most of these methods are count-based, word embedding based methods (Chang et al., 2017; Nickel and Kiela, 2017; Henderson and Popa, 2016) have become more popular in recent years. Other approaches represent words by Gaussian distributions and use KL-divergence as a measure of entailment (Vilnis and McCallum, 2014a; Athiwaratkun and Wilson, 2017). Especially for tasks like hypernymy detection, these methods have proven to be powerful as they not only capture the semantics but also the uncertainty about various concepts in which the word appears.

Using the framework presented in Section 4, we define a measure of entailment as the optimal transport cost (see Eq. (4)) between associated distributions under a suitable ground cost.

For this purpose, we rely on a model that was recently proposed by (Henderson and Popa, 2016; Henderson, 2017) which explicitly models what information is known about a word by interpreting each entry of the embedding as the degree to which a certain feature is present. Based on the logical definition of entailment they derive an approximate inference procedure and an operator measuring the degree of entailment between two so-called entailment vectors defined as follows: $\vec{v}_y \oslash \vec{v}_x = \sigma(-\vec{v}_y) \cdot \log \sigma(-\vec{v}_x)$, where the sigmoid function $\sigma$ and log are applied component-wise on the embeddings $\vec{v}_y, \vec{v}_x$. Thus, our choice for the ground cost $D$ on the basis of this entailment operator is

$$D_{ij}^{\text{Hend.}} := -\vec{v}_i \oslash \vec{v}_j. \tag{9}$$

This asymmetric and not necessarily positive ground cost illustrates that our framework can be flexibly used with an arbitrary cost function defined on the ground space.

**Evaluation.** In total, we evaluated our method on 9 standard datasets: BLESS (Baroni and Lenci, 2011), EVALution (Santus et al., 2015), Lenci/Benotto (Benotto, 2015), Weeds (Weeds et al., 2014), Henderson[7] (Henderson, 2017), Baroni (Baroni et al., 2012), Kotlerman (Kotlerman et al., 2010), Levy (Levy et al., 2014) and Turney

[7]This dataset is a subset of the Weeds dataset (https://github.com/julieweeds/BLESS).

(Turney and Mohammad, 2015). As an evaluation metric, we use average precision AP@all (Zhu, 2004).

For comparison we also report the performance of the entailment embeddings that were trained as described in (Henderson, 2017)[8]. 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 a hypernym relation.

Table 2 compares the performance[9] of entailment embeddings and the optimal transport measure based on the ground cost defined in Eq. (9). Our method yields significant improvements over the entailment embeddings by (Henderson, 2017) on almost all of the datasets. Only on the Baroni dataset, our method performs worse but nevertheless still achieves similar performance as other state-of-the-art methods. It confirms the findings of (Shwartz et al., 2016) and (Chang et al., 2017): there is no single hypernymy scoring function that performs best on all datasets. Furthermore, on some datasets (EVALution, LenciBenotto, Weeds, Turney) we even outperform or match state-of-the-art performance (cf. Table 3), by simply using our framework together with ground cost as defined in Eq. (9).

Notably, our method is not specific to the entailment vectors by (Henderson, 2017). It can be used with any embedding vectors and ground cost measuring the degree of entailment, without requiring any additional training. A more accurate ground cost or embedding vectors might even further improve the performance. Furthermore, our training dataset (Wikipedia with 1.7B tokens) and our vocabulary with only 80'000 words are rather small compared to the datasets used, e.g., by (Vilnis and McCallum, 2014a). We expect to get even better results by using a larger vocabulary on a larger corpus.

## 8 Conclusion

To sum up, we advocate for associating both a distributional and point estimate as a representation for each entity. We show how this allows us to use optimal transport over the set of contexts associated with these entities, in problems with a co-occurrence structure. Further, the framework enables efficient combination with existing point-estimates and embeddings, and we demonstrate its performance on several NLP tasks. In the end, our method offers a unique perspective for building rich feature representations that simultaneously capture uncertainty and interpretability.

[8] More details about the training setup can be found in section A.1 of the Appendix.

[9] We also illustrate the effect of several hyperparameters used in the PPMI computation in Table 4 of the Appendix A.

## References


Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 32–43.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

Martial Agueh and Guillaume Carlier. 2011. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. *ICLR*.

Ben Athiwaratkun and Andrew Gordon Wilson. 2017. Multimodal word distributions. *arXiv preprint arXiv:1704.08424*.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138.

Benotto. 2015. *Distributional Models for Semantic Relations: A Study on Hyponymy and Antonymy : PhD thesis*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Haw-Shiuan Chang, ZiYun Wang, Luke Vilnis, and Andrew McCallum. 2017. Distributional inclusion vector embedding for unsupervised hypernymy detection.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

A. Conneau and D. Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *ArXiv e-prints*.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300.

Marco Cuturi and Arnaud Doucet. 2014. Fast computation of wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Bejing, China. PMLR.

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification. In *WWW 2017 - International World Wide Web Conference*, pages 1045–1052, Perth, Australia.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. *arXiv preprint arXiv:1804.01882*.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. 2016. Stochastic optimization for large-scale optimal transport. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3440–3448. Curran Associates, Inc.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2018. Unsupervised Alignment of Embeddings with Wasserstein Procrustes. *arXiv*.

Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1809–1818. ACM.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD 2016 - Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 855–864. ACM.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 497–507.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

James Henderson. 2017. Learning word embeddings for hyponymy with entailment-based distributional semantics. *arXiv preprint arXiv:1710.02437*.

James Henderson and Diana Nicoleta Popa. 2016. A vector space for distributional semantics for entailment. *arXiv preprint arXiv:1607.03780*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. 2016. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*, pages 4862–4870.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *ACL - Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.

Leonid V Kantorovich. 1942. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP 2014 - Empirical Methods in Natural Language Processing*, pages 1746–1751.

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.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Omer Levy, Ido Dagan, and Jacob Goldberger. 2014. Focused entailment graphs for open ie propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97.

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.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7.

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.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244.

Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Boris Muzellec and Marco Cuturi. 2018. Generalizing Point Embeddings using the Wasserstein Space of Elliptical Distributions . *arXiv*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6341–6350.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

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

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 511–519.

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

Enrico Santus, Alessandro Lenci, Qin Lu, and S Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 38–42. EACL (European chapter of the Association for Computational Linguistics).

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In *38th International ACM SIGIR Conference*, pages 959–962.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2016. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. *arXiv preprint arXiv:1612.04460*.

Peter D Turney and Saif M Mohammad. 2015. Experiments with three approaches to recognizing lexical entailment. *Natural Language Engineering*, 21(3):437–476.

Cédric Villani. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Luke Vilnis and Andrew McCallum. 2014a. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*.

Luke Vilnis and Andrew D McCallum. 2014b. Word representations via gaussian embedding. *CoRR*, abs/1412.6623.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259. Dublin City University and Association for Computational Linguistics.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 81–88. Association for Computational Linguistics.

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2017. Starspace: Embed all the things! *arXiv preprint arXiv:1709.03856*.

Jianbo Ye, Yanran Li, Zhaohui Wu, James Z Wang, Wenjie Li, and Jia Li. 2017. Determining gains acquired from word embedding quantitatively using discrete distribution clustering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1847–1856.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945. Association for Computational Linguistics.

Mu Zhu. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2:30.

# A Supplementary Material

## A.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[10] 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 and 100 cluster centers. We followed the same training procedure as described in (Henderson, 2017) and were able to reproduce their scores on the hypernymy detection task.

## A.2 Miscellaneous

**PPMI Computation:** We utilize the sparse matrix support of Scipy[11] for efficiently carrying out all the PPMI computations.

**PPMI Normalizations:** Another possibility while considering the normalization to have an associated parameter $\beta$ that can interpolate between two extremes.

$$
\begin{aligned}
(\tilde{H}^w)_k^\beta &:= \frac{(\bar{H}^w)_k^\beta}{\sum_{k=1}^K (\bar{H}^w)_k^\beta}, \quad \text{where} \\
(\bar{H}^w)_k^\beta : g &= \frac{\text{SPPMI}_s^\alpha(w, \mathcal{C}_k)}{\sum_w \text{SPPMI}_s^{\alpha,\beta}(w, \mathcal{C}_k)} \qquad (10) \\
&= \frac{\sum_{c \in \mathcal{C}_k} \text{SPPMI}_s^\alpha(w, c)}{\sum_w \sum_{c \in \mathcal{C}_k} (\text{SPPMI}_s^\alpha(w, c))^\beta}.
\end{aligned}
$$

In particular, when $\beta = 1$, we recover the equation for histograms as in Section 5, and $\beta = 0$ would imply normalization with respect to cluster sizes.

**Optimal Transport Computation:** We make use of the Python Optimal Transport (POT)[12] 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. For all our computations involving optimal transport, we typically use $\lambda$ around 0.1 and make use of log or median normalization as common in POT to stabilize the Sinkhorn iterations.

**Clustering:** For clustering, we make use of kmcuda's[13] efficient implementation of K-Means algorithm on GPUs.

## A.3 Software Release

We plan to make all our code (for all these parts) and our pre-computed histograms (for the mentioned datasets) publicly available on GitHub soon.

## A.4 Detailed Results

Detailed results of the sentence representation and hypernymy detection experiments are listed on the following pages.

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

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

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

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

Table 4: Comparison between entailment vectors and optimal transport / Wasserstein based entailment measure (WE). Avg. gain refers to the average difference relative to the entailment vectors. Avg. gain w/o Baroni refers to the average difference while neglecting the Baroni dataset. The hyperparameter $\alpha$ refers to the smoothing exponent and $s$ to the shift in the PPMI computation. All scores are AP@all (%).

| | Dataset | | | | | |
|---|---|---|---|---|---|---|
| Method | BLESS | EVALution | LenciBenotto | Weeds | Henderson | Baroni |
| Henderson et al. | 6.4 | 31.6 | 44.8 | 60.8 | 70.5 | **78.3** |
| WE ($\alpha$=0.15, $s$=1) | 7.3 | 37.7 | 49.0 | 63.6 | 74.8 | 64.4 |
| WE ($\alpha$=0.15, $s$=5) | 6.9 | 39.1 | 49.4 | 64.3 | 74.0 | 65.2 |
| WE ($\alpha$=0.15, $s$=15) | 7.0 | 39.8 | 48.5 | 64.7 | 75.0 | 65.6 |
| WE ($\alpha$=0.5, $s$=1) | 6.6 | 39.2 | 48.6 | 62.9 | 76.1 | 64.6 |
| WE ($\alpha$=0.5, $s$=5) | 5.9 | 40.4 | 49.9 | 65.7 | 73.9 | 67.2 |
| WE ($\alpha$=0.5, $s$=15) | 5.5 | **40.5** | **49.5** | **66.2** | 72.8 | 67.4 |
| WE ($\alpha$=0.95, $s$=1) | **7.5** | 31.1 | 40.9 | 52.2 | **76.9** | 56.7 |

| | Dataset | | | | |
|---|---|---|---|---|---|
| Method | Kotlerman | Levy | Turney | **Avg.Gain** | **Avg. Gain (w/o Baroni)** |
| Henderson et al. | 34.0 | 11.7 | 56.6 | - | - |
| WE ($\alpha$=0.15, $s$=1) | 33.9 | 10.8 | 57.2 | +0.5 | +2.2 |
| WE ($\alpha$=0.15, $s$=5) | 34.2 | 11.6 | 57.0 | +0.8 | +2.5 |
| WE ($\alpha$=0.15, $s$=15) | 34.9 | 12.3 | **57.3** | +1.2 | **+2.9** |
| WE ($\alpha$=0.5, $s$=1) | 34.7 | 10.2 | 56.8 | +0.6 | +2.4 |
| WE ($\alpha$=0.5, $s$=5) | 34.6 | 11.3 | 56.5 | +1.2 | +2.7 |
| WE ($\alpha$=0.5, $s$=15) | **35.6** | **12.6** | 56.1 | **+1.3** | +2.8 |
| WE ($\alpha$=0.95, $s$=1) | 31.2 | 8.1 | 56.7 | -3.7 | -1.5 |

Table 5: **Evaluation of Unsupervised Sentence Representation**: Comparison of the performance of our model (WB) with SIF and Bag of Words (BoW) averaging on Semantic Textual Similarity (STS) tasks. Reported scores are average Pearson correlation x 100 on all instances of the STS dataset. We compare the results of our model (WB) for $K = 250$ and $K = 300$ clusters. The PPMI hyperparameters haven't been tuned much and the results below are for $\alpha$=0.95 and no log shift (i.e. $s$=1). The results for SIF are shown for the variants with no principal component (PC) removed and the $1^{st}$ PC removed. Both of these SIF results are for the best selected hyperparameter value of $a = 0.001$ obtained by running their evaluation script.

| | STS12 | | | | |
|---|---|---|---|---|---|
| Model | MSRpar | MSRvid | SMTeuroparl | WordNet | SMTnews |
| Bag of Words | 17.3 | -4.2 | 27.0 | 35.1 | 35.0 |
| SIF (with no PC removed) | 19.5 | 41.7 | 24.3 | 54.0 | 25.0 |
| SIF (with $1^{\text{st}}$ PC removed) | 21.0 | 36.5 | 31.0 | **55.4** | 27.9 |
| WB ($K$=250) | 30.5 | 57.4 | 45.3 | 44.5 | **38.8** |
| WB ($K$=300) | **32.7** | **57.5** | **48.0** | 45.0 | 38.1 |

| | STS13 | | |
|---|---|---|---|
| Model | FNWN | Headlines | WordNet |
| Bag of Words | **19.2** | 24.2 | 12.9 |
| SIF (with no PC removed) | 11.5 | 46.1 | 6.8 |
| SIF (with $1^{\text{st}}$ PC removed) | 14.3 | **54.3** | 60.4 |
| WB ($K$=250) | -1.1 | 44.1 | **62.8** |
| WB ($K$=300) | 1.1 | 44.7 | 61.1 |

| | STS14 | | | | | |
|---|---|---|---|---|---|---|
| Model | Forum | News | Headlines | Images | WordNet | Twitter |
| Bag of Words | 15.5 | 37.0 | 23.9 | 19.0 | 29.9 | 37.6 |
| SIF (with no PC removed) | 15.8 | 31.7 | 44.6 | 38.0 | 26.7 | 43.6 |
| SIF (with $1^{\text{st}}$ PC removed) | 15.2 | 35.7 | **52.1** | **47.4** | **62.6** | **58.0** |
| WB ($K$=250) | 32.6 | 54.9 | 39.5 | 37.0 | 58.6 | 48.3 |
| WB ($K$=300) | **33.1** | **56.3** | 39.2 | 36.6 | 58.6 | 50.2 |

| | STS15 | | | | |
|---|---|---|---|---|---|
| Model | Forum | Students | Belief | Headlines | Images |
| Bag of Words | 19.0 | 42.5 | 22.3 | 34.6 | 27.0 |
| SIF (with no PC removed) | 26.4 | 38.3 | 31.6 | 52.3 | 40.4 |
| SIF (with $1^{\text{st}}$ PC removed) | 30.0 | **62.0** | 39.0 | **59.1** | 50.6 |
| WB ($K$=250) | 30.4 | 50.6 | **40.1** | 53.0 | **50.8** |
| WB ($K$=300) | **37.6** | 52.5 | 39.1 | 53.1 | 50.0 |

| | STS16 | | | | |
|---|---|---|---|---|---|
| Model | Answer | Headlines | Plagiarism | Postediting | Question |
| Bag of Words | 15.7 | 32.0 | 26.1 | 34.8 | -1.6 |
| SIF (with no PC removed) | 21.3 | 49.1 | 14.2 | 35.5 | -6.4 |
| SIF (with $1^{\text{st}}$ PC removed) | **26.0** | **57.0** | 43.4 | 61.5 | 18.2 |
| WB ($K$=250) | 20.8 | 39.8 | 48.5 | 63.8 | 37.5 |
| WB ($K$=300) | 20.9 | 40.6 | **50.1** | **65.2** | **39.4** |