Polar Embedding

Ran Iwamoto*, Ryosuke Kohita†,Akifumi Wachi†
* Keio University, † IBM Research - Tokyo
raniwamoto@gmail.com, kohi@ibm.com, akifumi.wachi@ibm.com

Abstract

Hierarchical relationships are invaluable information for many natural language processing (NLP) tasks. Distributional representation has become a fundamental approach for encoding word relationships, particularly embeddings in hyperbolic space showed great performance in representing hierarchies by taking advantage of their spatial properties. However, most machine learning systems do not suppose to use in such complex non-Euclidean geometries. To achieve hierarchy representations in commonly used Euclidean space, we propose *Polar Embedding* that learns word embeddings with the polar coordinate system. Utilizing characteristics of polar coordinates, the hierarchy of words is expressed with two independent variables: radius (*generality*) and angles (*similarity*), and their variables are optimized separately. Polar embedding shows word hierarchies explicitly and allows us to use beneficial resources such as word frequencies or word generality annotations for computing radiuses. We introduce an optimization method for learning angles in limited ranges of polar coordinates, which combining a loss function controlling gradient and distribution uniformization. Experimental results on hypernymy datasets indicate that our approach outperforms other embeddings in low-dimensional Euclidean space and competitively performs even with hyperbolic embeddings, which possess a geometric advantage.

1 Introduction

A hierarchy is structured information that enables us to understand a specific object in a general sense (e.g., *dog* is one instance of *mammal*). Such generalization capability is or will be a basis of intelligent systems such as comprehending causality (Hassanzadeh et al., 2019), common sense (Talmor et al., 2019), and logic (Yang et al., 2017). For example, species information (e.g., *carnivora* vs. *herbivore*) will be useful when predicting behavior of animals. Another example is when developing a question answering system. Hierarchical relations of words enable the system to cover diverse inputs from a user (e.g., *how many paw pads does* a (*cat* | *kitten* | *tabby*) → (*cat*) *have?*). Therefore, to deploy hierarchical information in such systems, it is critical to represent word generality and meaning efficiently in a machine-readable manner.

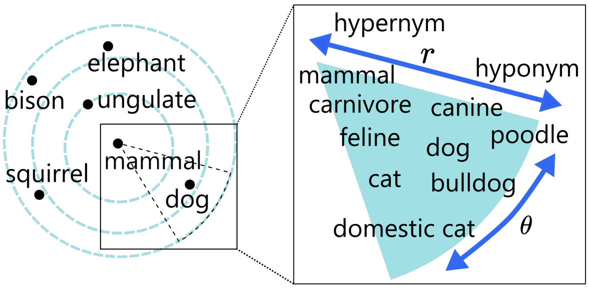


Figure 1: Conceptual illustration representing structures of word hierarchies.

For encoding word relationships, distributional representations are commonly used in natural language processing. Word embeddings such as Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and FastText (Bojanowski et al., 2017) express word similarities as continuous vectors in Euclidean space, and have brought significant advances to various applications (Collobert et al., 2011; Lample et al., 2016). Hierarchy-aware representations in Euclidean space have also been developed.

Gaussian embedding (Vilnis and McCallum, 2015) represents words as Gaussian distributions, whose mean vectors (50–100 dimension) encode word similarity, and whose variances encode word generality. Mean vectors have a similar role to existing word vectors, and word generalities are expressed by adding variance. On the other hand, order embedding (Vendrov et al., 2016) expresses both generality and similarity using word vectors, namely word positions. It models the partially or-

Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL), pages 470–480
November 10–11, 2021. ©2021 Association for Computational Linguistics

dered structure of a hierarchy between objects as inclusive relations of orthants in Euclidean space.

More recently, models using non-Euclidean space – hyperbolic, spherical space, or space of heterogeneous curvature have been gaining researchers' attention (Nickel and Kiela, 2017; Dhingra et al., 2018; Tifrea et al., 2019; Vilnis et al., 2018; Gu et al., 2019). Among them, hyperbolic geometry has good compatibility with spreading-out structures of hierarchies, since the volume of hyperbolic space increases exponentially to the direction of the radii (Sala et al., 2018).

For instance, Poincaré embedding (Nickel and Kiela, 2017), Lorentz embedding (Nickel and Kiela, 2018), and hyperbolic cone (Ganea et al., 2018b) perform excellently even with low dimensions by utilizing hyperbolic space. The idea of Poincaré embedding representing a hierarchy with a ball is intuitive and promising. The model learns embeddings of which (i) the distance from the origin of the ball represents *generality* of objects (e.g., *mammal* and *dog*) and (ii) the difference of angles represents *similarity* between objects (e.g., *dog* and *cat*), as shown in Figure 1.

However, hyperbolic embeddings require other components to also be developed under the same hyperbolic geometry (Ganea et al., 2018a). It may be challenging to apply the model to downstream applications that are mostly developed in Euclidean space (Du et al., 2018). Our goal is to achieve a low-dimensional, intuitive representation of hierarchical structures, such as poincare embedding, in commonly used Euclidean space.

In this paper, we propose *polar embedding* for learning representations on the polar coordinate system. Polar embedding can show word hierarchies explicitly using two independent variables: word generality is expressed by radius, and word similarity is expressed by angles. Radius and angles can be optimized separately, which allows us to use beneficial resources such as word frequencies or word generality annotations for computing radiuses. In short, polar coordinates provide us with a useful system to achieve the intuitive distribution of a hierarchy.

We also introduce techniques for learning hierarchy-aware representations while efficiently using an area in low-dimensional Euclidean space with the polar coordinate system.

To sum up, the contributions of this paper are threefold;

1. We introduce polar coordinates to learn hierarchy-aware embedding in Euclidean space.

2. We introduce two methods for distributing word angles for fully using limited spaces, i.e., Welsch loss function (Dennis and Welsch, 1978) and minimization of Kullback-Leibler (KL) divergence with Stein variational gradient descent (SVGD; (Liu et al., 2016)).

3. We show polar embedding performs competitively or better than other models learned in low-dimensional Euclidean and hyperbolic spaces on the link prediction benchmark.

2 Related Work

Popular word embeddings train word vectors by minimizing the Euclidean distance between words appearing in the same context (Mikolov et al., 2013; Pennington et al., 2014). Word embeddings showed significant progress in numerous NLP research; however, those embeddings have two points to be improved regarding the vector norm.

Meng et al. (2019) pointed out the first problem that there is a gap between training space and usage space of word embedding. The distributed representation is trained to minimize the distance between vectors of similar words. Although, the word vectors are normalized to the unit length and measured word similarities using cosine distance (Mikolov et al., 2013; Levy et al., 2015). The usage space is a ball, which is different from the training space. To learn a distributed representation in a space suitable for evaluation with cosine similarity, Meng et al. (2019) proposed spherical text embedding that use Riemannian manifold to learn word vectors with unit-norm constraints. Their model achieves high performance in the document classification task by learning embeddings in the same space as the usage space.

An other problem of popular word embeddings is that there is no explicit modeling of hierarchical relationships. Subsequent studies have shed light on the issue and extended word embeddings to be aware of hierarchical information. Gaussian embedding (Vilnis and McCallum, 2015) considers a hierarchy as an inclusion relation and represents it as Gaussian distributions with different variances so that general words have higher variance.

Word norms are frequently utilized to represent hierarchical structures, because in popular word embeddings such as glove, word vectors are normalized and norms are not utilized effectively. Nguyen et al. (2017) introduced a loss function to reflect pairwise hypernymy relations in similarity of word vectors, and Vulić and Mrkšić (2018) proposed a post-processing method for adjusting vector norms to enhance hierarchical relationships. Order embedding (Vendrov et al., 2016) represents a hierarchy by preserving the partial order between words.

By learning similarity on the Euclidean sphere and by expressing word generality in norm, it is expected that the word hierarchy can be effectively represented in Euclidean space.

3 Polar Embedding

We propose polar embedding that learns word representations in the polar coordinate system. The most essential feature of polar coordinates is that it holds the radius and angles of a position vector as separate parameters. Given that the intuitive distribution of a hierarchical structure shown in Figure 1, we can naturally associate the radius with word generality and the angles with word similarity. In this section, we describe methods of optimizing the radius and angles towards the simple but efficient representation of a hierarchy in low-dimensional Euclidean space.

3.1 Angle Distributions in Polar Coordinates

First, we explain the characteristics of polar coordinates using a three-dimensional sphere. Left of Figure 2 illustrates three-dimensional polar coordinates. Their angles have range limitations: $\theta \in [0, 2\pi), \varphi \in (0, \pi)$, and the points of $\varphi = 0, \pi$ are called pole.

In cartesian coordinates, the volume of a differential cube does not depend on the value of each coordinate (the position of the cube). On the other hand, in polar coordinates, the differential cube is smaller when φ is close to the poles. More specifically, if we put the samples at equal intervals along θ and φ dimension (e.g. every $\pi/180$ radian), the Euclidean distance between two samples near the poles is closer than the distance between samples near $\varphi = \pi/2$. These properties need to be considered when optimizing angles.

3.2 Preliminaries

First, let us introduce the notations throughout the following sections. Let $\mathcal{W}^n = \{\mathbf{w} \in \mathbb{R}^n \mid \|\mathbf{w}\| < r_{\max}\}$ be the open n-dimensional ball where $r_{\max} \in \mathbb{R}$ is the radius and $\| \cdot \|$ denotes the Euclidean norm. In an n-dimensional ball $\mathcal{W}^n$, a word w is represented by a vector $\mathbf{w} = (r, \theta, \varphi^1, \varphi^2, ..., \varphi^{n-2})$, where $r \in (0, r_{\max}), \theta \in [0, 2\pi), \varphi^k \in (0, \pi)$, for $k = 1, 2, ..., n-2$.

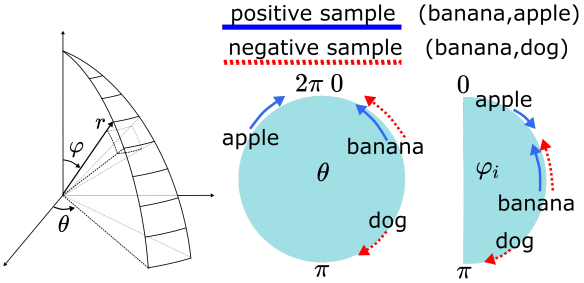


Figure 2: Three-dimensional polar coordinates and angle optimizations.

Given two words w_i and w_j, in the range of $\theta \in [0, 2\pi)$ which forms a circle by regarding $\theta = 2\pi$ as $\theta = 0$ (the center of Figure 2), the distance between θ_{w_i} and θ_{w_j} is defined with an absolute difference:

$$d(\theta_{w_i}, \theta_{w_j}) = \min\left(2\pi - |\theta_{w_i} - \theta_{w_j}|, |\theta_{w_i} - \theta_{w_j}|\right), \tag{1}$$

where $\min(\cdot, \cdot)$ selects the shorter arc. In the range of $\varphi^k \in (0, \pi)$ which forms a half-circle (the right of Figure 2), the distance between $\varphi^k_{w_i}$ and $\varphi^k_{w_j}$ is defined as an absolute difference:

$$d(\varphi^k_{w_i}, \varphi^k_{w_j}) = |\varphi^k_{w_i} - \varphi^k_{w_j}|, \tag{2}$$

where $k \in \{1, \ldots, n-2\}$.

Note that the maximum distance is bounded by at most π in the θ dimension and less than π in the φ^k dimensions according to the above definitions.

In representation learning, distances are optimized so that semantically relevant words become closer and irrelevant words become farther apart. Let us define $w_{(t)}$ as a target word, $w_{(+)}$ as a relevant word to $w_{(t)}$, and $w_{(-)}$ as an irrelevant word to $w_{(t)}$. Common approaches such as Skip-gram with negative sampling (Mikolov et al., 2013) minimize a loss function $\mathcal{L} = \mathcal{L}_{pos} - \mathcal{L}_{neg}$, where $\mathcal{L}_{pos}$ is a cumulative loss of positive samples (a set of relevant pairs), and $\mathcal{L}_{neg}$ is of negative samples (a set of irrelevant pairs). Given a word hierarchical tree, a word pair connected by an edge is a positive sample, and a non-connected word pair is a negative sample. An example of the angle update with these positive and negative samples is illustrated in Figure 2.

3.3 Radius

Radius (r) is expected to represent word generality. Specifically, general words (e.g., *mammal*, *furniture*) should have smaller values of r (i.e., near the origin) and specific words (e.g., *bulldog*, *wooden chair*) should have larger values (i.e., far from the origin). The radius r can be defined in arbitrary ways as long as it satisfies the above characteristics. In the case of learning embeddings from word pairs in a hierarchical tree, for example, the number of edges of a target word (i.e., how many words are connected to the target word) can be used as a definition of r because a word at an upper level in a hierarchy is likely to be connected to more words. If a whole or partial hierarchical tree(s) is available, information related to hierarchical levels such as node height and number of descendants, can represent generality more precisely.

3.4 Angles

Angles (θ, φ^k) are expected to represent the similarity of words. We optimize them basically with the same approach as most embeddings; making angles closer for positive samples and far for negative samples as shown in Figure 2. However, the polar coordinate system has limits with respect to the value ranges in the optimization; a word can move on the circle of θ and on the half-circle of φ^k. Note that the learning of θ and φ^k is independent from r, and we fix r to 1 during the process of updating angles. Given the characteristics of polar coordinates, we propose optimization methods to utilize a whole sphere broadly for the effective use of a limited space. More specifically, we embed the similarity of words while maintaining a uniform distribution on a sphere.

3.4.1 Optimization

We now introduce a method to optimize the angle vectors. A conventional approach for optimizing the embedding vectors is to use the squared loss function. In polar coordinates, however, the conventional approach results in a highly biased distribution over words in terms that majority of the words is likely to accumulate near the limits of the angle ranges. Specifically, words were likely to gather at the positions at which the distance becomes near the maximum value (i.e., π). This is because the squared loss function has large gradients for distantly separated samples but their angles have value range limitations. We describe those details in Section 3.4.2.

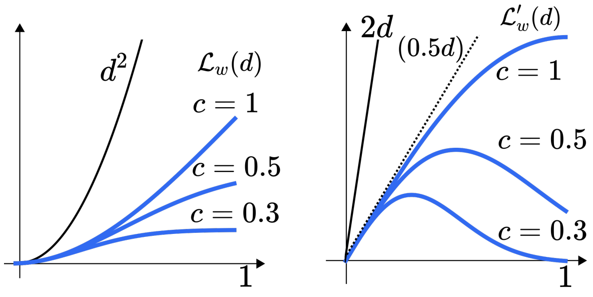


Figure 3: Loss functions (left) and their gradients (right) of Welsch loss (blue) and squared loss (black).

To address the above issue, we adopt two techniques for optimizing angle vectors. First, we train polar embedding using the Welsch loss function (Dennis and Welsch, 1978). This function is characterized by the following fact; the gradient is bounded and takes small value with large d. Hence, the Welsch loss function prevents words from gathering at certain positions by decreasing the gradient for negative samples. Second, we use stein variational gradient descent (SVGD, Liu et al. (2016)) algorithm to correct the embedding vectors to the uniform distribution. Intuitively, SVGD is used to reduce the KL divergence between the true uniform distribution, $p(\cdot)$ and current distribution, $q(\cdot)$. This uniformization by SVGD is conducted during the training of the embedding vectors once in every specific iterations. The pseudo code is described in Algorithm 1.

3.4.2 Optimization by Welsch loss function

In this section, we explain why the squared loss function does not work in polar coordinates and how we overcome the issue with the Welsch loss function.

Welsch loss function. The Welsch loss function is defined as follows:

$$\mathcal{L}_w(d) = \frac{c^2}{2}\left[1 - \exp\left(-\frac{d^2}{2c^2}\right)\right], \tag{3}$$

where d is the angle distance of two words described in Equations (1) and (2), and c is a hyperparameter. The gradient is represented as:

$$\frac{\partial \mathcal{L}_w(d)}{\partial d} = \frac{d}{2}\exp\left(-\frac{d^2}{2c^2}\right). \tag{4}$$

As seen in Figure 3, the gradient is bounded and takes a small value with large d.

Why does the squared loss function not work? The issue stems from the two facts; (i) the gradient of the squared loss function can be arbitrarily large for negative samples, and (ii) the maximum distance between two points in each angle dimension is bounded in the polar coordinate system. In the squared loss function, the larger value (i.e., longer distance) gives a large gradient (the black lines in Figure 3). Therefore, pairs of negative samples distribute farther and farther from each other during learning. In the learning of standard Euclidean embeddings, it is not problematic because they are allowed to use the space infinitely (e.g., the vector norm can be as large as needed). However, polar coordinates has the limits in the value range: $\theta \in [0, 2\pi)$ and $\varphi^k \in (0, \pi)$, and the maximum distance in each dimension is at most or less than π (see Equations (1) and (2)). In other words, the angles of negative samples cannot be apart more than π, and their optimization will stop after reaching it. The squared loss function particularly causes words to be accumulated because it keeps negative samples away, as mentioned above, which results in the biased distributions. Therefore, with the squared loss function, it is difficult to obtain uniform distributions on a sphere or even learn appropriate angles in polar coordinates.

How does the Welsch loss function solve the issue? As discussed above, the policy of the squared loss function for negative samples — farther is better — causes a biased distribution. Therefore, we need to modify it so that negative samples are regarded as *sufficiently distant* if they are a certain distance apart. Although it is possible to take a heuristic approach, such as clipping squared loss function, the Welsch loss function can naturally satisfy this requirement because the gradient is bounded and takes a small value with large d. The peak of the gradient can be considered a threshold in which the gradient increases when approaching the boundary then decreases after passing it (blue lines in the right of Figure 3). In other words, the Welsch loss function does not eagerly move negative pairs of which distance is beyond the threshold. Hence, we can suppress the accumulation problem of words with the appropriate threshold given by adjusting c. For example, we can define $\varphi^k_{w_{(t)}}$ and $\varphi^k_{w_{(-)}}$ as sufficiently distant when $d(\varphi^k_{w_{(t)}}, \varphi^k_{w_{(-)}}) = 0.5\pi$.

3.4.3 Uniformization by SVGD

To further mitigate the issue of word accumulation discussed in the previous section, we use SVGD for achieving a more uniform distribution of the embedding vectors because the Welsch loss function does not directly take into account uniformity. SVGD is a deterministic, gradient-based sampling algorithm, which minimizes the KL divergence between the target (uniform) distribution p and trained distribution q. With SVGD, we define the following Kernelized Stein Discrepancy (KSD) $S(\cdot,\cdot)$ between the true posterior distribution $p(x)$ and approximated posterior distribution $q(x)$, in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}^d$.

$$\mathcal{S}(q, p) = \max_{\phi \in \mathcal{H}^d} \left\{ \mathbb{E}_{x \sim q} \left[\mathcal{A}_p \phi(x) \right] \right\}, \tag{5}$$

where $\mathcal{A}_p \phi(x) = \phi(x) \nabla_x \log p(x) + \nabla_x \phi(x)$, and $\phi(x)$ is a smooth vector function. The optimal solution of (5) is given by

$$\phi^*_p(x') = \mathbb{E}_{x \sim q} \left[\kappa(x, x') \nabla_x \log p(x') + \nabla_x \kappa(x, x') \right], \tag{6}$$

where $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel satisfying a certain condition on the expectation value of differential of κ (Stein, 1972), and the radial basis function $\kappa(x, x') = \exp\left(-\gamma \left|\left|x - x'\right|\right|^2\right)$ satisfies this condition. Liu and Wang (2016) theoretically analyzed the relationship between KSD and KL divergence and proved that

$$\nabla_\epsilon \mathrm{KL}(q_\epsilon \,||\, p) \,|_{\epsilon=0} = -\mathbb{E}_{x \sim q} \left[\mathcal{A}_p \phi(x) \right],$$

where ϵ is a perturbation and q_ϵ is the perturbed density of the distribution x. This equation means that ϕ^*_p in (6) is the optimal perturbation direction providing the largest descent of the KL divergence.

To use this optimization method for our purpose, we need a mathematical representation of the probability density of the uniform distribution on the sphere in the polar coordinate system. However, because there is no such analytical expression to our best knowledge, we approximate it by a Gaussian Mixture model (GMM) with an appropriate kernel function (e.g. a radial basis function).

4 Experiments

Following previous studies (Nickel and Kiela, 2017; Ganea et al., 2018b), we used the transitive closure in WordNet (Miller, 1995) as our experimental dataset and compared polar embedding with

Algorithm 1 Learning procedure of angles

Input for the main loop: Iteration N, Dataset D, Vocabulary V, Learning rate α, Weight for negative samples β, SVGD Interval S

Input for SVGD: Iteration M, Learning rate η, Early stopping criterion $\gamma \in [0, 1]$

```
$p_\theta, p_{\varphi^k} \leftarrow$ approximate angle uniform distributions on a sphere with GMM
$\theta, \varphi^k \leftarrow$ Initialize vectors in Cartesian coordinates with a normal distribution, then convert them into polar coordinates for starting with a uniform distribution on a sphere (Miller, 1995)
for $n = 0$ to $N$ do
    $w_{(t)}, w_{(+)}, w_{(-)} \leftarrow$ sample from $D$

    // Update of $\theta$ with the Welsch loss
    $\hat{\theta}_{w_{(t)}} = \theta_{w_{(t)}} + \alpha\{\mathcal{L}'_w(d(\theta_{w_{(t)}}, \theta_{w_{(+)}})) - \beta\mathcal{L}'_w(d(\theta_{w_{(t)}}, \theta_{w_{(-)}}))\}$
    $\theta_{w_{(t)}} \leftarrow \hat{\theta}_{w_{(t)}} \bmod 2\pi$    ▷ $\mathrm{mod} 2\pi$ for the update across $2\pi$

    // Update of $\varphi^k$ ($\forall k \in \{1, \ldots, n-2\}$) with the Welsch loss
    $\hat{\varphi}^k_{w_{(t)}} \leftarrow \varphi^k_{w_{(t)}} + \alpha\{\mathcal{L}'_w(d(\varphi^k_{w_{(t)}}, \varphi^k_{w_{(+)}})) - \beta\mathcal{L}'_w(d(\varphi^k_{w_{(t)}}, \varphi^k_{w_{(-)}}))\}$
    if $\hat{\varphi}^k_{w_{(t)}} \in (0, \pi)$ then $\varphi^k_{w_{(t)}} \leftarrow \hat{\varphi}^k_{w_{(t)}}$    ▷ No update if $\hat{\varphi}^k_{w_{(t)}}$ overflows from the range

    // Update of $\theta$ and $\varphi^k$ ($\forall k \in \{1, n-2\}$) with SVGD
    if $n \equiv 0 \bmod S$ then
        $\hat{\theta}, \hat{\varphi}^k \leftarrow \mathrm{SVGD}(\theta, p_\theta), \mathrm{SVGD}(\varphi^k, p_\varphi)$
        for $w_i \in V$ do    ▷ Update $\theta$ and $\varphi^k$ for each word
            $\theta_{w_i} \leftarrow \hat{\theta}_{w_i} \bmod 2\pi$
            if $\hat{\varphi}^k_{w_i} \in (0, \pi)$ then $\varphi^k_{w_i} \leftarrow \hat{\varphi}^k_{w_i}$

procedure SVGD$(x, p)$    ▷ $x$ is $\theta$ or $\varphi^k$
    $s \leftarrow$ Compute a validation score before SVGD
    $X \leftarrow$ Create a set of batched samples from $x$
    for $m = 0$ to $M$ do
        for $x' \in X$ do $x' \leftarrow x' + \eta\phi^*_p(x')$
        $s' \leftarrow$ Compute a validation score with the latest representation
        if $s' < \gamma s$ then break    ▷ Stop SVGD if the validation score drops much
    return $x'$    ▷ $x'$ denotes updated values of $x$
```

other hierarchy-aware embeddings. First, our embedding trained on the WordNet mammal subtree was shown to understand our method intuitively, and then we evaluated polar embedding with a link prediction task for quantitative evaluation.

4.1 Settings

Dataset and Task. WordNet is a directed acyclic graph (DAG) consisting of edges that represent is-a relations of words. Each word w_i in WordNet represents a single node in the DAG. An edge represents a word pair (w_i, w_j) where w_i is a hypernym of w_j. A model is expected to embed such hierarchical relations in a latent space appropriately. In our experiment, we used the preprocessed mammal/noun hierarchy provided by (Ganea et al., 2018b). The number of words in mammal hierarchy is 1180, and in noun hierarchy is 82114.

On the WordNet noun hierarchy, we evaluated models with the link prediction task, which is a binary classification to predict if an hypernym-hyponym edge exists between two words. We first learned embeddings with the training set then classified edges in the validation/test set into existent or non-existent edges by using the embeddings with a scoring function as described in next paragraph. We evaluated models with the F1 score.

Polar Embedding. We determined r in a deterministic manner and trained angles as explained

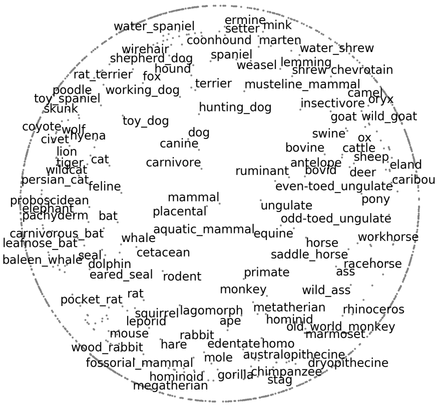


Figure 4: Two-dimensional polar embedding trained on the mammal subtree.

	Dimension = 5			Dimension = 10		
Model	Percentage of Available Edges in Training					
(Space)	10%	25%	50%	10%	25%	50%
Polar r^g (E)	**78.5%**	**79.9%**	**81.8%**	**82.2%**	81.6%	82.3%
Polar r^e (E)	77.6%	78.8%	78.8%	81.8%	81.2%	82.4%
Simple (E)	71.3%	73.8%	72.8%	75.4%	78.4%	78.1%
Order (E)	70.2%	75.9%	81.7%	69.7%	79.4%	**84.1%**
Cone (E)	69.7%	75.0%	77.4%	81.5%	**84.5%**	81.6%
Disk (E)	38.9%	42.5%	45.1%	54.0%	65.8%	72.0%
Poincare (H)	70.2%	78.2%	83.6%	71.4%	82.0%	85.3%
Cone (H)	80.1%	86.0%	92.8%	85.9%	91.0%	94.5%
Disk (H)	69.1%	81.3%	83.1%	79.7%	90.5%	94.2%

Table 1: Experimental results from link-prediction task on WordNet noun hierarchy. (E) and (H) denote Euclidean and hyperbolic spaces.

in Section 3.4. We tested two types of r for simulating different scenarios; r^e for the situation in which no hierarchical information is available and only word pairs are given, and r^g for the situation in which hierarchical information is available. On the training set of the WordNet noun hierarchy, r^e is defined as $r^e_i = 1 - z(\log(e_i + 1))$ where e_i is the number of edges of the i-th word. r^g is defined as $r^g_i = 1 - z(h_i + \log(l_i + 1))$ where h_i is a maximum height and l_i is the number of descendants of the i-th word in the hierarchy. The notation z is a min-max normalization function; hence, $r^e \in [0, 1]$ and $r^g \in [0, 1]$. The intuitions of those definitions are simple. For r^e, a word connected with many words is likely to be placed at an upper level in a hierarchy. For r^g, a word with a larger height and more descendants is likely to be placed at an upper level in a hierarchy. The r^g is expected to be more precise with respect to word generality because of the direct usage of hierarchical relationships while r^e is only aware of local connections between words. For practical use, we can set r from reliable word generality resources.

Finally, we introduced the following two scoring functions:

$$\begin{aligned} s_a(w_i, w_j) &= \frac{f(w_i) \cdot f(w_j)}{\|f(w_i)\| \|f(w_j)\|}, \\ s_r(w_i, w_j) &= |r_i - r_j|, \end{aligned}$$

where f is a conversion function from polar to cartesian coordinates (Blumenson, 1960).

We then defined the criterion as:

$$s(w_i, w_j) = \begin{cases} 1 & \text{if } s_a(w_i, w_j) > 1 - \tau s_r(w_i, w_j)^2 \\ 0 & \text{otherwise,} \end{cases}$$

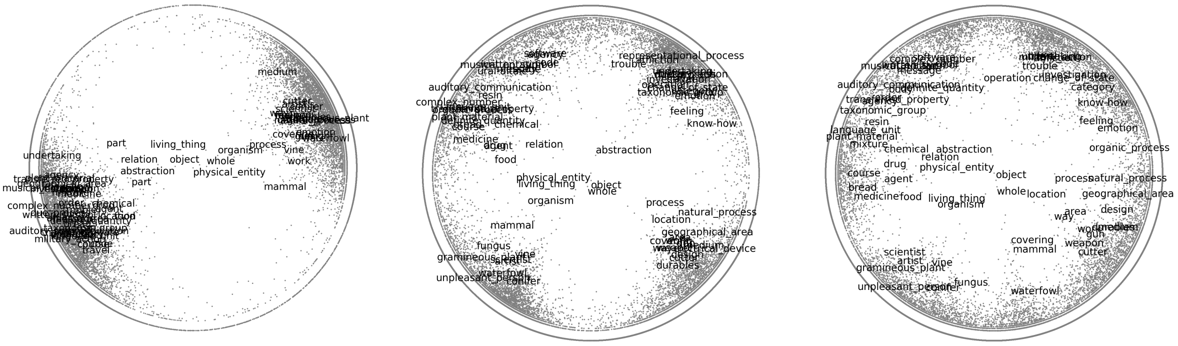
where τ is a hyperparameter tuned in the validation set. This function detects an edge between w_i and w_j when their angles are closer (i.e., higher similarity) and their radii are different (i.e., one is more general than the other). Considering the spreading-out structure of a hierarchy, this scoring function relaxes the condition for the angle similarity along with the increase in the radius difference. We tested all models with 5 or 10 dimensions as in a previous study (Ganea et al., 2018b).

Baselines. We compared polar embedding with four Euclidean (Simple, Order, Cone, Disk) and three hyperbolic models (Poincaré, Cone, Disk) (Vendrov et al., 2016; Nickel and Kiela, 2017; Ganea et al., 2018b; Suzuki et al., 2019). The Euclidean simple model learns embeddings by minimizing Euclidean distance in Cartesian coordinates. The results of compared methods are from (Ganea et al., 2018b).

4.2 Results: Mammal Subtree

Figure 4 illustrates two-dimensional polar embedding trained on the WordNet mammal subtree. Thanks to our uniformization methods, words scattered in the circle all around, and apparent hierarchies could be found. For example, *cat* and *dog* are species of *carnivore*, the relationship was reflected with polar embedding. Also, it created a sub-hierarchy; we could find *hunting dog* and *terrier* at the outer of *dog*, and *lion* and *wildcat* at the outer of *cat*. Such species hierarchies were well embedded for others as well (e.g., *aquatic mammal* → *cetacean* → *seal*, *dolphin*, and *primate* → *monkey* → *gorilla*, *ape*).

Practically, we can extract those hierarchical or-



(a) Squared Loss

(b) Welsch Loss

(c) Welsch Loss + SVGD

Figure 5: Distributions of noun hierarchy with polar embedding.

ders with r; obtaining hypernyms by increasing r and hyponyms by decreasing r. We can also extract similar words by using cosine similarity. For example, we can collect similar words to *dog* such as *canine*, *toy dog*, and *fox* regardless of the hierarchical orders. In addition, by combining cosine similarity with r, we can filter similar words with the levels in a hierarchy.

4.3 Results: Noun Subtree

The F1 scores on the WordNet noun benchmark are listed in Table 1. When the dimension was 5, polar embedding exhibited superior performance in most cases with both r^g and r^e. It also performed better than or competitively with hyperbolic models. When we increased the dimension to 10, however, the performance gain of polar embedding was not large compared to the other Euclidean models though it still showed competitive performance.

Table 2 shows the ablation study of the Welsch loss function and SVGD. The Welsch loss function significantly increased the score compared to the squared loss function. In short, the gradient adjustment for defining "sufficiently distant" was critical for learning angles in the polar coordinates. As expected, SVGD enhanced the performance for the one using the Welsch loss function.

Figure 5 illustrates actual θ distributions of the models used in our ablation study. First, if we simply used the squared loss function, most words gathered at either the left or right side (Figure 5a). The distribution was highly biased, and the embedding trained with squared loss function failed to use the full space effectively. By changing the loss function to the Welsch loss function, the bias largely decreased, and the model used the broader area (Figure 5b). Finally, SVGD further improved the biased distribution, and words distributed almost uniformly on the sphere (Figure 5c). While the Welsch loss function implicitly prevents the biased distribution, SVGD more explicitly forces word angles to distribute in uniform on a sphere. It enabled the model to use Euclidean space more broadly, which resulted in better performance. We also found the same trend for φ^k.

Loss - SVGD	F1
Welsch - w/	78.5%
Welsch - w/o	74.9%
Squared - w/	69.1%
Squared - w/o	65.5%

Table 2: Ablation study of Welsch loss function and SVGD. Settings were as follows: dimension = 5, percentage of training edges = 10, and $r = r^g$.

5 Conclusion

We proposed polar embedding, which represents hierarchical structures in low-dimensional Euclidean space. Word generalities and similarities are intuitively expressed using radius and angles in polar coordinates. We introduced the Welsch loss function and SVGD for training embeddings in the angle limit of polar coordinates, which keeps angle distributions uniform and enables a model to leverage a whole space effectively. Experimental results indicated that polar embedding outperformed other embeddings in Euclidean space.

Acknowledgements

The first author would like to acknowledge colleagues from her internship at IBM Research - Tokyo for their valuable discussion.

Ethical Considerations

Polar Embedding is a method for creating distributed representations of words, which does not produce ethically problematic expressions such as hate speech thus it has a low ethical risk.

References

LE Blumenson. 1960. A derivation of n-dimensional spherical coordinates. *The American Mathematical Monthly*, 67(1):63–66.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *TACL*, 5:135–146.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *JMLR*, 12:2493–2537.

John E. Dennis and Roy E. Welsch. 1978. Techniques for nonlinear least squares and robust regression. *Communications in Statistics - Simulation and Computation*, 7(4):345–359.

Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. 2018. Embedding Text in Hyperbolic Spaces. In *Proceedings of the Workshop on Graph-Based Methods for Natural Language Processing*, pages 59–69.

Lun Du, Zhicong Lu, Yun Wang, Guojie Song, Yiming Wang, and Wei Chen. 2018. Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach. In *IJCAI*, pages 2079–2085.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. 2018a. Hyperbolic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, pages 5345–5355.

Octavian.-E. Ganea, Gary. Becigneul, and Thomas. Hofmann. 2018b. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *ICML*, pages 1646–1655.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. 2019. Learning Mixed-Curvature Representations in Product Spaces. In *ICLR*.

Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2019. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. In *IJCAI*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *NAACL*, pages 260–270.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *TACL*, 3:211–225.

Qiang Liu, Jason Lee, and Michael Jordan. 2016. A kernelized stein discrepancy for goodness-of-fit tests. In *ICML*, pages 276–284.

Qiang Liu and Dilin Wang. 2016. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *NeurIPS*, pages 2378–2386.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical Text Embedding. In *NeurIPS*, pages 8208–8217.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS*, pages 3111–3119.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. In *EMNLP*, pages 233–243.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NeurIPS*, pages 6338–6347.

Maximillian Nickel and Douwe Kiela. 2018. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *ICML*, pages 3779–3788.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*, pages 1532–1543.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. 2018. Representation Tradeoffs for Hyperbolic Embeddings. volume 80 of *PMLR*, pages 4460–4469.

Charles Stein. 1972. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pages 583–602.

Ryota Suzuki, Ryusuke Takahama, and Shun Onoda. 2019. Hyperbolic Disk Embeddings for Directed Acyclic Graphs. In *ICML*, pages 6066–6075.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *ACL*, page 4149–4158.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincare Glove: Hyperbolic Word Embeddings. In *ICLR*.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-Embeddings of Images and Language. In *ICLR*.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. In *ACL*, pages 263–272.

Luke Vilnis and Andrew McCallum. 2015. Word Representations via Gaussian Embedding. In *ICLR*.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising Word Vectors for Lexical Entailment. In *NAACL*, pages 1134–1145.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *NeurIPS*, pages 2319–2328.

A Appendix

A.1 Convert Polar Coordinates to Cartesian Coordinates

We relate the definition of polar coordinates given in Section 3.2 to the definition of Cartesian coordinates. Polar coordinates can be converted to Cartesian coordinates (Blumenson, 1960) and the information expressed in embedding is not missing in the conversion. For example, it is easier to use orthogonal coordinates to calculate cosine similarity. Let $\mathbf{w} = (r, \theta, \varphi^1, \varphi^2, ..., \varphi^{n-2})$ be a n-dimension vector in polar coordinates. Suppose $\bar{\mathbf{w}} = \{x_1, x_2, \ldots, x_n\}$ is the corresponding vector of $\mathbf{w}$ in Cartesian coordinates. Here, the angles θ and φ^k in n-dimensional polar coordinates are represented as follows:

$$\begin{aligned} \theta &= 2 \operatorname{arccot} \frac{x_{n-1} + \sqrt{{x_n}^2 + x_{n-1}^2}}{x_n}, \\ \varphi^k &= \arccos \frac{x_k}{\sqrt{{x_n}^2 + x_{n-1}^2 + \cdots + x_k^2}}. \end{aligned}$$

A.2 Angle Distributions in Polar Coordinates

As mentioned in Section 3.4.3, achieving a uniform distributions on a sphere with polar coordinates is complicated. Figure 6 shows distributions of θ and φ^k where samples uniformly distribute on a sphere at $r = 1$. Whereas the distribution of the θ dimension is intuitive, samples do not equally distribute in the φ^k dimensions; fewer samples around the poles and more samples around the center. Also, the distributions differ depending on the dimensions. This is stem from the fact that volume around the center and the poles is different on a sphere.

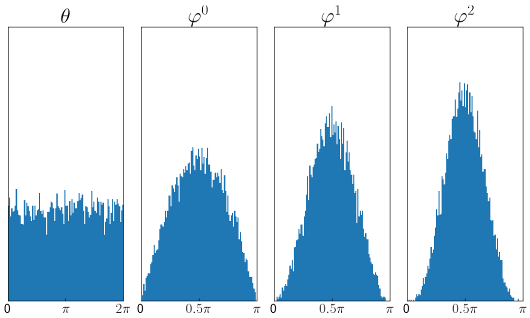


Figure 6: Angle distributions in five-dimensional sphere.

A.3 Hyperparameters

We list hyperparameters in Table 3 and Table 4. In the experiment, we reported the results with the best one on the validation set for each model and each dataset. Training time is 2–6 hours at 500000 iterationson a single NVIDIA GeForce GTX 1080 Ti. The score of the validation set was approximately same value as the score of the test set. We used grid search to find the best parameters, and performed one trial for each parameter.

Hyperparameter		Searched values
General		
Dimension	-	5, 10
Batch size	-	128, 1024
Iteration	N	500000
Learning rate	α	0.1, 0.3
Learning rate decay	-	0.95, 0.99
Negative sampling rate	β	0.1, 0.3, 0.5
Welsch loss parameter	c	0.4
SVGD		
Batch size	-	128
Interval	S	1000,2000, 5000
Iteration	M	2, 5, 20
Learning rate	η	1
Early stopping criterion	γ	0.99

Table 3: Hyperparameters for the noun subtree models

Hyperparameter		Searched values
General		
Dimension	-	2
Batch size	-	128
Iteration	N	10000
Learning rate	α	0.1, 0.3, 0.5
Learning rate decay	-	0.95, 0.99
Negative sampling rate	β	0.1, 0.3, 0.5
Welsch loss parameter	c	0.4
SVGD		
Batch size	-	128
Interval	S	500
Iteration	M	5
Learning rate	η	1
Early stopping criterion	γ	0.99

Table 4: Hyperparameters for the mammal subtree models