ENCODING CATEGORY TREES INTO WORD-EMBEDDINGS USING GEOMETRIC APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a novel method to implicitly encode a tree-structured category information into word-embeddings, resulting in super-dimensional ball representations (n-ball embedding for short). Inclusion relations among n-balls precisely encode subordinate relations among categories. The cosine similarity function is enriched by category information. A large n-ball dataset is constructed using geometrical method, which achieves zero energy cost in embedding tree structures into word embedding. A new benchmark dataset is created for predicting the category of unknown words. Experiments show that n-ball embeddings, carried with category information, significantly out-perform word-embeddings in the neighbourhood test, while only slightly change the original word-embeddings. Experiment results also show that n-ball embeddings demonstrate surprisingly good performance in validating the category of unknown word. Source codes and data-sets are free for public access `https://github.com/gnodisnait/nball4tree.git` and `https://github.com/gnodisnait/bp94nball.git`.

1 INTRODUCTION

Words in similar contexts have similar semantic and syntactic information. Word embedding is the vector representation of words with this feature, e.g., Mikolov et al. (2013); Pennington et al. (2014). Word embedding has been widely used in AI applications, such as question-answering, e.g., Tellex et al. (2003), text classification, e.g., Sebastiani (2002), information retrieval, e.g.,Manning et al. (2008). Collobert et al. (2011) developed a unified NLP system based on word embedding to processing common NLP tasks. To enhance semantic reasoning, researchers proposed to represent words as regions, instead of vectors. For example, Erk (2009) extended a word vector into a region by estimating the log-linear probability of weighted feature distances, and found that hyponym regions often do not fall inside of their hypernym regions. By using external hyponymy relations, her experiment reached 95.2% precision and 43.4% recall in hypernym prediction for a small scale dataset. These results suggest that regions structured by hyponymy relations may not be located within the same dimension as the original space of word-embeddings, and that strict inclusion relation among regions is still an open task for representing hyponymy relations.

In this article, we restrict regions to be n-dimensional balls (n-ball for short) and propose a novel geometrical training approach to encode tree-structured category information into word-embeddings with two criteria as follow: (1) Subordinate relations among categories shall be implicitly represented by inclusion relations among corresponding n-balls; (2) The tree structure of category relations shall be precisely encoded. That is, the energy loss shall be zero. Our contributions are as follow: (1) A novel geometric approach is proposed to construct/train n-balls, so that tree-structured category relations are encoded as inclusion relations among n-balls with zero energy loss; (2) A new similarity measurement is proposed, considering both the location and the size of an n-ball, which is more precise and sensitivity as compared with the cosine similarity in the experiment; (3) A large dataset of n-ball embedding based on Glove with a category tree extracted from WordNet 3.0 is free for public access.

The rest of the article is structured as follows: Section 2 presents the structure of n-ball embeddings; Section 3 describes the geometric approach to construct n-ball embeddings; Section 4 reports experiment results; Section 5 briefly reviews some related works; Section 6 concludes the presented work, and lists on-going works.

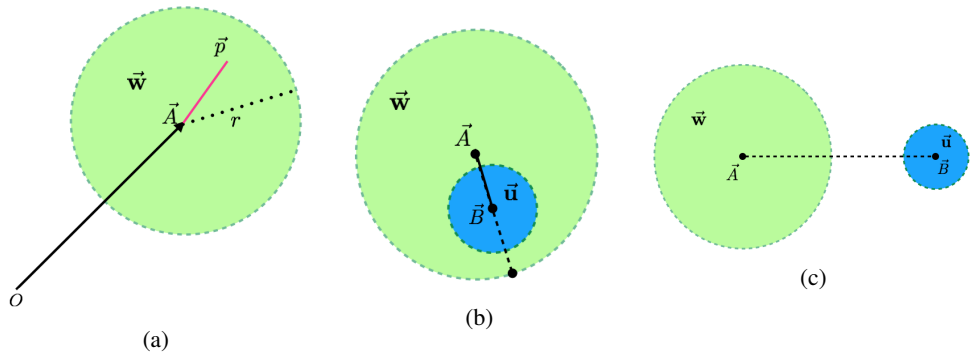


Figure 1: (a) the structure of n-ball; (b) $\vec{\mathbf{u}}$ is inside of $\vec{\mathbf{w}}$; (c) $\vec{\mathbf{u}}$ disconnects from $\vec{\mathbf{w}}$

2 Region-Based Embeddings

Tree structures are widely used in many applications, e.g. file systems, the taxonomy of plants or animals, subordinate relations of governments. We use the tree structure of hyponymy relations among word senses as representative example to construct n-ball embeddings. Formally, an n-ball of word-sense w with the central point $\vec{A}_w$ and radius r_w is written as $\vec{\mathbf{w}} = \mathbb{B}(\vec{A}_w, r_w)$, defined as the set of vectors whose Euclidean distance to $\vec{A}_w$ is less than r_w: $\mathbb{B}(\vec{A}_w, r_w) \triangleq \{\vec{p} | dis(\vec{A}_w, \vec{p}) < r_w\}$, where $dis(\vec{A}_w, \vec{p}) = \sqrt{(\vec{A}_w - \vec{p})^\top (\vec{A}_w - \vec{p})}$ is the Euclidean distance between $\vec{A}_w$ and $\vec{p}$. We define n-ball as an open-space, as illustrated in Figure 1(a).

2.1 Relations between n-balls

We distinguish two relations between n-balls: *being inside of*, and *disconnecting from*, as illustrated in Figure 1(b-c). $\vec{\mathbf{u}} = \mathbb{B}(\vec{A}_u, r_u)$ being inside of $\vec{\mathbf{w}} = \mathbb{B}(\vec{A}_w, r_w)$ can be measured by the result of subtracting the sum of radius r_u and the distance between their central vectors from radius r_w. Formally, we define $\mathcal{D}_{inside}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \triangleq r_w - r_u - dis(\vec{A}_u, \vec{A}_w)$, That is, $\vec{\mathbf{u}}$ is inside of $\vec{\mathbf{w}}$, if and only if $\mathcal{D}_{inside}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \geq 0$. w is the hypernym of u, if and only if $\mathcal{D}_{inside}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \geq 0$. If v is the hypernym of w, w is the hypernym of u, we have

$$\mathcal{D}_{inside}(\vec{\mathbf{u}}, \vec{\mathbf{v}}) > \mathcal{D}_{inside}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) + \mathcal{D}_{inside}(\vec{\mathbf{w}}, \vec{\mathbf{v}}) \geq \mathcal{D}_{inside}(\vec{\mathbf{w}}, \vec{\mathbf{v}})$$

The direct hypernym of $\vec{\mathbf{v}}$, written as $\mathbf{dh}(\vec{\mathbf{v}})$, can be defined as the $\vec{\mathbf{x}}$ which produces the minimal positive value of $\mathcal{D}_{inside}(\vec{\mathbf{v}}, \vec{\mathbf{x}})$. Formally, $\mathbf{dh}(\vec{\mathbf{v}}) \triangleq \arg\min_{\vec{\mathbf{x}}, \vec{\mathbf{x}} \neq \vec{\mathbf{v}}} \mathcal{D}_{inside}(\vec{\mathbf{v}}, \vec{\mathbf{x}}) \geq 0$

Similarly, $\vec{\mathbf{u}}$ disconnecting from $\vec{\mathbf{w}}$ can be measured by the result of subtracting the distance between their center points from the sum of their radii. Formally, we define $\mathcal{D}_{disc}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \triangleq r_w + r_u - dis(\vec{A}_u, \vec{A}_w)$, That is, $\vec{\mathbf{u}}$ disconnects from $\vec{\mathbf{w}}$, if and only if $\mathcal{D}_{disc}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \leq 0$. We define $\mathcal{D}_{disc}(\vec{\mathbf{u}})$ as the $\mathcal{D}_{disc}$ measurement of $\vec{\mathbf{u}}$ to its nearest neighbor, that is, $\mathcal{D}_{disc}(\vec{\mathbf{u}}) \triangleq \max \mathcal{D}_{disc}(\vec{\mathbf{u}}, \vec{\mathbf{w}})$, for all $\mathcal{D}_{disc}(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \leq 0$.

2.2 Similarity Measurement

Similarity is normally measured by the cosine value of two vectors, Mikolov et al. (2013). For n-balls, the similarity between two balls can be approximated by the cosine value of their central vectors. Formally, given two n-balls $\vec{\mathbf{u}} = \mathbb{B}(\vec{A}_u, r_u), \vec{\mathbf{w}} = \mathbb{B}(\vec{A}_w, r_w)$, their cosine similarity can be defined as $\cos(\vec{A}_u, \vec{A}_w)$. One weakness of the method is that we do not know the boundary of the lowest $\cos$ value below which two word senses are not similar. Using category information, we

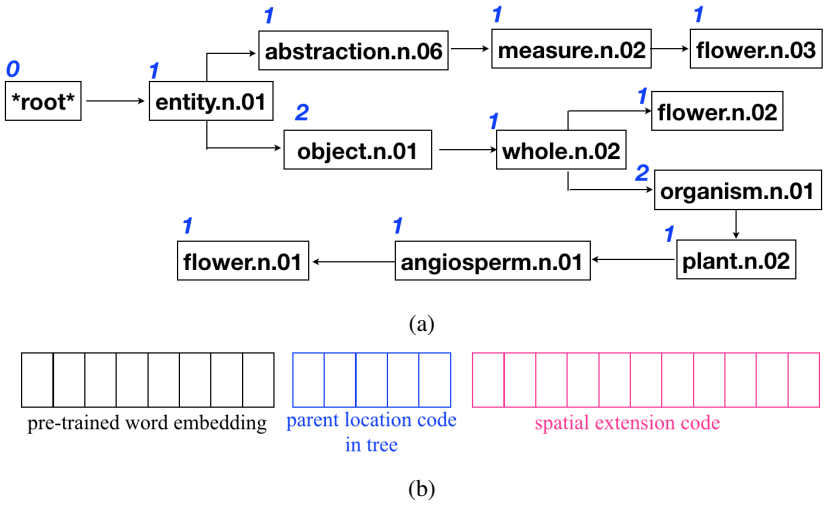


Figure 2: (a) Hypernymy relations of three word senses of flower in WordNet 3.0; (b) Three components of the center point of an n-ball

can define that two word senses are not similar, if they have different direct hypernyms. Formally,

$$Sim_0(\vec{\mathbf{u}}, \vec{\mathbf{w}}) \triangleq \begin{cases} \cos(\vec{A}_u, \vec{A}_w) & \mathbf{dh}(\vec{\mathbf{u}}) = \mathbf{dh}(\vec{\mathbf{w}}) \\ 0 & \text{otherwise} \end{cases}$$

3 CONSTRUCTING n-BALL EMBEDDINGS

$\mathcal{N}$-ball embeddings encode two types of information: (1) word-embeddings, (2) tree structures of hyponymy relations among word-senses. A word can have several word-senses. We need to create a unique vector to describe the location of each word-sense in trees. We introduce a virtual-root (*root*) to be the parent of all roots of the trees, fix word-senses in the alphabetic order in each layer, and number each word-sense based on the fixed order. A fragment structure is as illustrated in Figure 2(a). The path of a word-sense to *root* can be uniquely named by the numbers along the path. For example, the path of from *root* to *flower.n.03* is [*entity.n.01, abstraction.n.06, measure.n.02, flower.n.03*][1], which can be uniquely named as [1,1,1,1]; We call this vector *location code* of *flower.n.03*. The *location code* of the direct hypernym of *flower.n.03* is called *Parent Location Code (PLC)*. *PLC* of *flower.n.03* is [1,1,1], *PLC* of *whole.n.02* is [1,2].

As a word may not co-occur in the same context as its hypernym in the corpus, their co-occurrence relations can be weak, and the cosine similarity could even be less than zero. For example, in GloVE embedding, $\cos$(ice_cream, dessert)$= -0.1998$, $\cos$(tuberose, plant)$= -0.2191$. This follows that hypernym ball must contain the origin point of the n-dimensional space, if it contains its hyponym, as illustrated in Figure 3 (a). When this happens to two semantically unrelated hypernyms, n-balls of the two unrelated hypernyms shall share the original point. For example, the ball of desert shall share a common part with the ball of plant. This violates our second criteria. To avoid such case, we restrict that every n-ball shall not contain the origin point O. This can be computationally achieved by adding dimensions, and realised by introducing *spatial extension code*, a constant non-zero vector, as illustrated in Figure 2(b). To intuitively understand this, imagine that you stand in the middle of two objects A and B, and cannot see both of them. To see both of them without turning the head or the eyes, you shall walk several steps away, so that the angle between A and B is less than some degree, as illustrated in Figure 3(c).

[1]We exclude *root* from paths

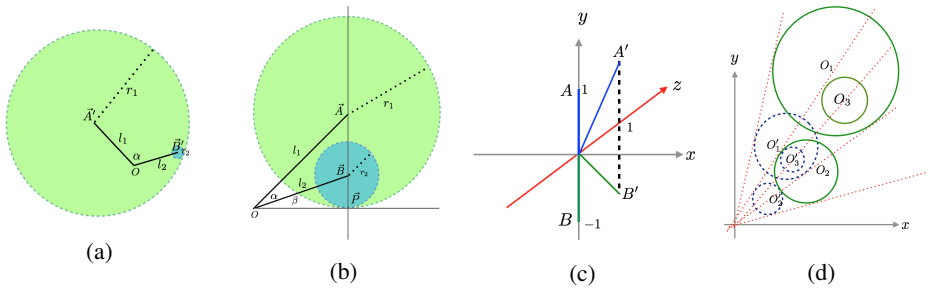


Figure 3: (a) The angle between ball $(\vec{A}', r_1)$ and ball $(\vec{B}', r_2)$ is greater than $90°$, so, ball $(\vec{A}', r_1)$ will contain O, if it contains ball $(\vec{B}', r_2)$; (b) If ball $(\vec{A}, r_1)$ contains ball $(\vec{B}, r_2)$, not containing O, the sum of α and β shall be less than $90°$; (c) The angle $\angle AOB = 180°$. If we shift A and B 1 unit on the new dimension, we have $A'(0, 1, 1)$ and $B'(0, -1, 1)$, the angle $\angle A'OB'$ will become $90°$; (d) O_1, O_2, O_3 are homothetically transformed into O'_1, O'_2, O'_3, inclusion relations among them are preserved

3.1 The Structure of the Central Vector

Following Zeng et al. (2014) and Han et al. (2016), we structure the central vector of an n-ball by concatenating three vectors: (1) the pre-trained word embedding, (2) the parent location code in the hypernym tree (if the code is shorter than the max length, we append '0's till it reaches the fixed length), (3) spatial extension code.

3.2 Zero Energy Loss for n-Ball Construction Using Geometric Approach

Our second criteria is to perfectly encode subordinate relations among categories into inclusion relations among n-balls. This is a big challenge, as the widely adopted back-propagation training process, Rumelhart et al. (1988), quickly reaches a non-zero local minimal and terminates[2]. The problem is that when the location or the size of an n-ball is updated to improve its relation with a second ball, its relation to a third ball will be very easily deteriorated. We propose the classic depth-first recursion process, listed in **Algorithm 1**, to travel the category tree, and update sizes of locations of n-balls using three geometric transformations as follow.

Homothetic transformation (H-tran), which keeps the direction of the central point, and enlarges lengths of $\vec{A}$ and r with the same rate k. $\mathcal{H}(\mathbb{B}(\vec{A}, r), k) \triangleq \mathbb{B}(\vec{A}', r')$, such that $|\vec{A}|/|\vec{A}'| = r/r' = k$ and $\vec{A}/|\vec{A}| = \vec{A}'/|\vec{A}'|$;

Shift transformation (S-tran), which keeps the length of the radius r, and add a new vector $\vec{s}$ to $\vec{A}$. $\mathcal{S}(\mathbb{B}(\vec{A}, r), \vec{s}) \triangleq \mathbb{B}(\vec{A}', r)$, such that $\vec{A}' = \vec{A} + \vec{s}$;

Rotation transformation (R-tran), which keeps the length of the radius r, and rotates angle α of $\vec{A}$ at the plane spanned by the i-th and j-th element of $\vec{A}$. $\mathcal{R}(\mathbb{B}(\vec{A}, r), \alpha, i, j) \triangleq \mathbb{B}(\vec{A}', r)$, such that $\vec{A}_k = \vec{A}'_k$, $k \neq i, j$, $\vec{A}'_i = \vec{A}_i \cos\alpha + \vec{A}_j \sin\alpha$ and $\vec{A}'_j = \vec{A}_j \cos\alpha - \vec{A}_i \sin\alpha$.

To prevent from the deterioration of already improved relations, we use the principle of family action: *if a transformation is applied for one ball, the same transformation will be applied for all its child balls*. Among three transformations, only homothetic transformation preserves inclusion and disconnectedness relations among n-balls, as illustrated in Figure 3(d), therefore has the priority to be used.

In the process of adjusting sibling n-balls to be disconnected from each other, we apply homothetic transformation obeying the principle of family action. When an n-ball is too close to the origin of the space, a shift/rotate transformation will be applied, which will change the pre-trained word embeddings.

[2]This phenomenon appears even in very small dataset. We developed a visual simulation for illustration. The source code is available at `https://github.com/gnodisnait/bp94nball.git`

Following the depth-first procedure, a parent ball is constructed after all its child balls. Given a child n-ball $\mathbb{B}(\vec{B}, r_2)$, a candidate parent ball $\mathbb{B}(\vec{A}, r_1)$ is constructed as the minimal cover of $\mathbb{B}(\vec{B}, r_2)$, illustrated in Figure 3(b). The final parent ball $\mathbb{B}(\vec{P}, r_p)$ is the minimal ball which covers these already constructed candidate parent balls $\mathbb{B}(\vec{P}_i, r_{p_i})$.

Algorithm 1: $training_one_family$(root): Depth first algorithm to construct n-balls of all nodes of a tree

input : a tree pointed by root; each node stores a word sense and its vector embedding
output: the n-ball embedding of each node

```
children ⟵ get_all_children_of(root)
if number_of(children) > 0 then
    foreach child ∈ children do
        // depth first
        training_one_family(child)
    end
    if number_of(children) > 1 then
        // training siblings to be disconnected from each other
        train_to_be_disconnected_among(children)
    end
    // create parent ball for all children
    root = create_parent_ball_of(children)
else
    initialize_ball(root)
end
```

4 Experiments and Evaluations

4.1 Experiment 1: Constructing n-ball Embeddings for Large-Scale Datasets

We use GloVE word embedding, ie. Pennington et al. (2014), as the pre-trained vector embedding of words, and extract trees of hyponymy relations among word-senses from Word-Net 3.0, ie. Miller (1995). Totally, we have $54,310$ word-senses, and 169 trees, the largest tree with root *entity.n.01* has $43,669$ word-senses. Source codes for training, along with input datasets, are free for public access `https://github.com/gnodisnait/nball4tree.git`. We proved that all relations in the category tree are preserved in n-ball embeddings. That is, zero energy cost is achieved by the geometric approach.

4.2 Experiment 2: Qualitative Evaluation of n-ball Embeddings

Following Levy & Goldberg (2014), we do qualitative evaluations. We manually inspect nearest neighbours, and compare results with pre-trained GloVE embeddings. A sample is listed in Table 1 - 2, and observed interesting results as follows.

Precise neighbourhood identification n-ball embeddings successfully separate word-senses of polysemy very precisely. For example, nearest neighbours of berlin.n.01 are all cities, nearest neighbours of berlin.n.02 are all names, as listed in Table 1;

Typed cosine similarity function better than the normal cosine function Sim_0 enriched by category information produces much better neighbourhood word-senses than the normal cosine measurement. For example, the top-5 nearest neighbours of *beijing* in GloVE using normal cosine measurement: *china, taiwan, seoul, taipei, chinese*, among which only *seoul* and *taipei* are cities. The top-10 nearest neighbors of *berlin* in GloVE using normal cosine measurement: *vienna, warsaw, munich, prague, germany, moscow, hamburg, bonn, copenhagen, cologne*, among which *germany* is a country. A worse problem is that neighbours of the word-sense berlin.n.02 as the family name do not appear.

Category information contributes to the sparse data problem Due to sparse data, some words with similar meanings have negative cosine similarity value. For example, *tiger* as a fierce or audacious person (tiger.n.01) and *linguist* as a specialist in linguistics (linguist.n.02) seldom appear in the same context, resulting -0.1 cosine similarity value using Glove word-embddings. However, they are hyponyms of person.n.01, using this information our geometrical process transform the n-balls of tiger.n.01 and linguist.n.02 into the inner side of the n-ball of person.n.01, resulting in high similarity value measured by the typed cosine function.

word-sense 1	word-sense 2 (Sim_0, cos)
beijing.n.01	london.n.01 (>0.99, 0.47), atlanta.n.01 (>0.99, 0.27) washington.n.01 (>0.99, **-0.11**), paris.n.0 (>0.99, 0.46), potomac.n.02 (>0.99, 0.18), boston.n.01(>0.99, 028)
berlin.n.01	madrid.n.01 (>0.99, 0.47), toronto.n.01 (>0.99, 0.46), rome.n.01 (>0.99, 0.68), columbia.n.03 (>0.99, 0.39), sydney.n.01 (>0.99, 0.52), dallas.n.01(>0.99, 0.28)
berlin.n.02	simon.n.02 (>0.99, 0.34), williams.n.01 (>0.99, 0.24), foster.n.01, (>0.99, 0.13), dylan.n.01 (>0.99, 0.10), mccartney.n.01 (>0.99, 0.23), lennon.n.01(>0.99, 0.25)
tiger.n.01	survivor.n.02 (>0.99, 0.40), neighbor.n.01 (>0.99, 0.36), immune.n.01(>0.99, 0.10), linguist.n.02 (>0.99, **-0.1**), bilingual.n.01 (>0.99, **-0.06**), warrior.n.01 (>0.99, 0.68)
france.n.02	white.n.07(>0.99, 0.31), woollcott.n.01(>0.99, **-0.12**), uhland.n.01(>0.99, **-0.32**), london.n.02(>0.99, 0.52), journalist.n.01(>0.99, 0.33), poet.n.01(>0.99, 0.20)
cat.n.01	tiger.n.02(>0.99, 0.62), fox.n.01(>0.99, 0.44), wolf.n.01(>0.99, 0.67), wildcat.n.03(>0.99, 0.16), tigress.n.01(>0.99, 0.40), vixen.n.02(>0.99, 0.38)
y.n.02	q.n.01(>0.99, 0.45), delta.n.03(>0.99,0.33), n.n.05(>0.99, 0.60), p.n.02(>0.99, 0.44), f.n.04(>0.99, 0.55), g.n.09(>0.99, 0.52)

Table 1: Top-6 nearest neighbors based on Sim_0, the $\cos$ value of word stems are listed, e.g. $\cos$(beijing, london)$= 0.47$, tiger.n.01 refers to a fierce or audacious person

Upper category identification Using n-ball embeddings, we can find upper-categories of a word-sense. Given word-sense ws, we collect all those cats satisfying $\mathcal{D}_{inside}(ws, cat) > 0$. These cats shall be upper-categories of ws. If we sort them in the increasing order, and mark the first cat with $+_1$, the second with $+_2$, so on, the cat marked with $+_1$ is the direct upper-category of ws, the cat marked with $+_2$ is the direct upper-category of the cat with $+_1$, so on, as listed in Table 2;

word-sense 1	word-sense 2
beijing.n.01	city.n.01$+_1$, municipality.n.01$+_2$, region.n.03$+_3$, location.n.01$+_4$, object.n.01$+_5$, entity.n.01$+_6$
berlin.n.02	songwriter.n.01$+_1$, composer.n.01$+_2$, musician.n.02$+_3$, artist.n.01$+_4$, creator.n.02$+_5$
tiger.n.01	person.n.01$+_1$, organism.n.01$+_2$, whole.n.02$+_3$, object.n.01$+_4$, entity.n.01$+_5$
france.n.02	writer.n.01$+_1$, communicator.n.01$+_2$, person.n.01$+_3$, organism.n.01$+_4$
cat.n.01	wildcat.n.03$+_1$, lynx.n.02$+_2$, cougar.n.01$+_3$, bobcat.n.01$+_4$, caracal.n.01$+_5$, ocelot.n.01$+_6$, feline.n.01$+_7$,jaguarundi.n.01$+_8$
y.n.02	letter.n.02$+_0$ character.n.08$+_1$ symbol.n.01$+_2$, signal.n.01$+_3$ communication.n.02$+_4$

Table 2: $+_k$ represents the kth minimal positive value of $\mathcal{D}_{inside}$

4.3 Experiment 3: Consistency to Pre-Trained Word-Embedding

The n-ball representation method takes pre-trained word-embedding, and applies homothetic, shifting, and rotating transformations in the construction/training process. Shifting and rotating transformations may change the original direction of word-embedding. The aim of this experiment is to examine the effect of the n-ball training process to pre-trained word-embeddings.

Method 1 A word-embedding of word w is used by the n-ball representation of its word-sense, $w_1, \ldots, w_n$. Each can change the pre-trained value in its own way. As homothetic transformation has the highest priority, and does not change pre-trained word-embeddings. We examine the standard deviation (std) of the pre-trained word-embedding in n-ball representations of its word-senses. The less the std is, the better it is preserved.

The trained n-ball representations have 32,503 word-stems. For each word-stem, we extract word-embedding parts from n-ball representations, normalise them, minus pre-train word-embeddings, and compute standard deviation.

The maximum std is 0.7666. There are 417 stds greater than 0.2, 6 stds in the range of $(0.1, 0.2]$, 9 stds in the range of $(10^{-12}, 0.1]$, 9699 stds in the range of $(0, 10^{-12}]$, 22,372 stds equals 0. With this statistics we conclude that only a tiny portion (1.3%) of pre-trained word-embeddings have a small change (std $\in (0.1, 0.7666]$).

Method 2 The quality of word-embedding is evaluated by computing the correspondence (spearman co-relation) between similarities between human-judged word relations and vector-based word similarity relations. The standard datasets in the literature are WordSim353 dataset, which consists of 353 pairs of words, each pair is associated with a human-judged value about the co-relation between the two words, Finkelstein et al. (2001), and Stanfords Contextual Word Similarities (SCWS) dataset, which contains 2003 word pairs, each with 10 human judgments on similarity, Huang et al. (2012). Based on the result of Method 1, we extract the word-embedding of w from n-ball representations of w's word-senses, and use the average value as w's word-embedding.

Unfortunately, both datasets cannot be directly used within our experiment setting, as some words do not appear in the ball-embedding, due to several reasons as follow: (1) words whose word-senses having neither hypernym, nor hyponym in WordNet 3.0, e.g. *holy*; (2) words whose word-senses having different word stems, e.g. *laboratory, midday, graveyard, percent, zoo, FBI,* ...; (3) words having no word-senses, e.g. *Maradona*; (4) words whose word-senses using its basic form as word stems, e.g. *clothes, troops, earning, fighting, children*. After removing all the missing words, we have 318 paired words from WordSim353 and 1719 pairs from SCWS dataset for the evaluation.

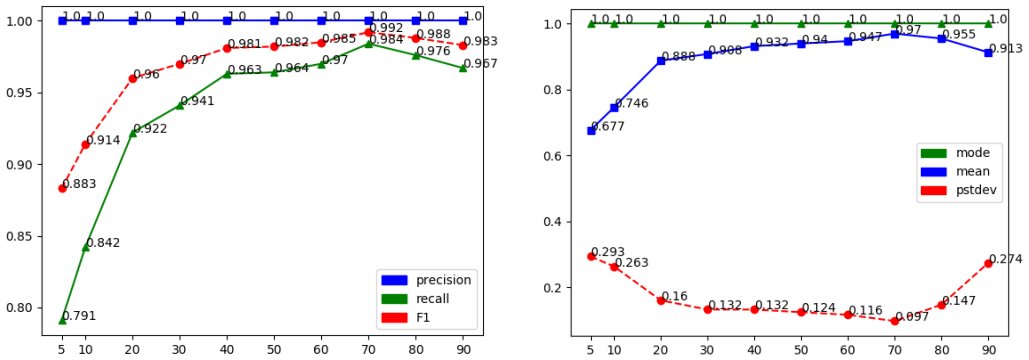
We get exactly the same spearman co-relation values in all 11 testing cases: the spearman co-relation on WordSim318 is 76.08%; each test on spearman co-relations using SCWS1719 is also the same. We conclude that n-ball representation is a loyal extension to word-embeddings.

4.4 Experiment 4: Membership Validation

The fourth experiment is to validate whether an unknown word belongs to a category, which is a basic task for prediction, e.g. Baroni et al. (2014). We describe the task as follows: Given pre-trained word-embeddings $\mathcal{E}_N$ with vocabulary size N, and a tree structure of hyponymy relations $\mathcal{T}_K$ on vocabulary W_K, $(K < N)$. Given $w_x \notin W_K$ and $c \in W_K$, we need to decide $w_x \in c$, or $w_x \notin c$? For example, when we read *mwanza is also experiencing*, we may guess *mwanza* a person; if we continue to read *mwanza is also experiencing major infrastructural development*, we would say *mwanza* is a city. Whether *mwanza* is a city, or a person?

Dataset From 54,311 word-senses, we randomly selected 1,000 word-senses of nouns and verbs as target categories, with the condition that each of them has at least 10 direct hyponyms; For each target category, we randomly select $p\%$ ($p \in [5, 10, 20, 30, 40, 50, 60, 70, 80, 90]$) from its direct hyponyms as training data. The test datasets are generated from three sources: (1) the rest $1 - p\%$ as true value, (2) randomly choose 1,000 false value from W_K; (3) 1,000 words from W_N which does not exist in W_K. Totally, we created 118,938 hyponymy relations in the training set, and 17,975,042 hyponymy relations in the testing set.

Method We develop an n-ball solution to solve the membership validation task as follows: suppose c has a hypernym path $[c, h_1, h_2, \ldots, h_m]$ and has several known members (direct hyponyms) $t_1, \ldots, t_s$. For example, *city.n.01* has a hypernym path [*city.n.01, municipality.n.01, urban_area.n.01, ..., entity.n.01*] and several known members *oxford.n.01, banff.n.01, chicago.n.01*. We construct n-ball embeddings for this small tree with the stem $[c, h_1, h_2, \ldots, h_m]$ and leaves $t_1, \ldots, t_s$, and record the geometric transformations. Suppose that w_x be a member of c, we initialise the n-ball of w_x using the same parameter as t_1, and apply the geometric transformations of t_1's n-ball for w_x's n-ball. If the final n-ball of w_x is located inside of the n-ball of c, we will decide that w_x is the member of c, otherwise not. This method can be explained in terms of Support Vector Machine, i.e., Shawe-Taylor & Cristianini (2004), as follows: the boundary of c's n-ball is supported by all n-balls of its known members. If the unknown word were a member of c, according to the principle of family action, its n-ball shall have the same geometric transformations as the



(a) (b)

Figure 4: (a) Precision, recall, F1 score of hypernym prediction, when $5\%, 10\%, 20\%, \ldots, 90\%$ of the members are used for training; (b) Mode, mean, and population standard deviation of the recall, when $5\%, 10\%, 20\%, \ldots, 90\%$ of the members are used for training

other known members of c, and contribute one candidate parent ball (section 3.2). Theoretically, we can introduce a ratio $\gamma \geq 1$ to zoom-out the boundary of c's n-ball created by its known members.

Evaluation and Analysis In the experiment, results show that the n-ball method is very precise and robust, as shown in Figure 4: the precision is always 100%, even only 5% from all members is selected as training set. The method is quite robust: If we select 5% as training set, the recall of prediction reaches 76.8%; if we select 50% as training set, the recall of prediction reaches 96.7%. Theoretically, the n-ball method can not guarantee 100% recall, as shown in Figure 4(b). If $p < 70\%$, the population standard deviation (pstdev) decreases, when more members are selected as training data. When $p > 70\%$, there is a slight increase of pstdev. The reason is that in the experiment setting, if more than 80% of the children are selected, it can happen that only one unknown member left for validating. If this single member is excluded outside of thecategory's n-ball, the recall is dropped to 0, which increases pstdev. The experiment result can be downloaded at `https://drive.google.com/file/d/1szwjyZhLQ75LE04kL2BpcszgJ0XuOYgi/view?usp=sharing`.

5 Related Work

Lenci & Benotto (2012) explored the possibility of identifying hypernyms in distributional semantic model; Santus et al. (2014) presented an entropy-based model to identify hypernyms in an unsupervised manner; Kruszewski et al. (2015) induced mappings from words/sentences embeddings into Boolean structure, with the aim to narrow the gap between co-occurrence based embeddings and logic-based structures. There are some works in word-embedding and knowledge graph embedding using regions to represent words or entities. Athiwaratkun & Wilson (2017) used multi-modal Gaussian distribution to represent words; KG2E model He et al. (2015) embeds entities using Gaussian distributions; Xiao et al. (2016) used manifolds to represent entities and relations; Nickel & Kiela (2017) used Poincaré ball to embed tree structures. In these works, relations, if represented, are explicit, and zero-energy cost is not targeted in the training process.

6 Conclusion and On-going Work

We proposed a novel geometrical method to seamlessly and precisely integrate external tree-structured category information into word-embeddings, resulting in a region-based (n-ball representation) word-sense embedding. Promising experiment results are presented. $\mathcal{N}$-ball method demonstrated great performance in validating category of unknown words, the reason behind is under further investigation. Our on-going work also include to develop n-ball approach for knowledge graph embedding, where multiple relations on directed acyclic graphs need to be considered.

REFERENCES

Ben Athiwaratkun and Andrew Wilson. Multimodal word distributions. In *ACL'17*, pp. 1645–1656. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1151. URL http://www.aclweb.org/anthology/P17-1151.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL'14*, pp. 238–247. Association for Computational Linguistics, 2014. doi: 10.3115/v1/P14-1023. URL http://www.aclweb.org/anthology/P14-1023.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2078186.

Katrin Erk. Supporting inferences in semantic space: Representing words as regions. In *IWCS-8 '09*, pp. 104–115, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-90-74029-34-6. URL http://dl.acm.org/citation.cfm?id=1693756.1693769.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: the concept revisited. In *WWW*, pp. 406–414, 2001.

Xu Han, Zhiyuan Liu, and Maosong Sun. Joint representation learning of text and knowledge for knowledge graph completion. *CoRR*, abs/1611.04125, 2016. URL http://arxiv.org/abs/1611.04125.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *CIKM '15*, pp. 623–632, New York, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806502. URL http://doi.acm.org/10.1145/2806416.2806502.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *ACL'12: Long Papers - Volume 1*, pp. 873–882, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390524.2390645.

Germán Kruszewski, Denis Paperno, and Marco Baroni. Deriving boolean structures from distributional vectors. *Transactions of the Association of Computational Linguistics*, 3:375–388, 2015. URL http://www.aclweb.org/anthology/Q15-1027.

Alessandro Lenci and Giulia Benotto. Identifying hypernyms in distributional semantic spaces. In *SemEval '12*, pp. 75–79, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2387636.2387650.

Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *ACL'14, Volume 2: Short Papers*, pp. 302–308, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P14/P14-2050.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *CoRR*, abs/1705.08039, 2017. URL http://arxiv.org/abs/1705.08039.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP'14*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0-262-01097-6. URL http://dl.acm.org/citation.cfm?id=65669.104451.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. Chasing hypernyms in vector spaces with entropy. In *EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pp. 38–42, 2014. URL http://aclweb.org/anthology/E/E14/E14-4008.pdf.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1): 1–47, March 2002. ISSN 0360-0300. doi: 10.1145/505282.505283. URL http://doi.acm.org/10.1145/505282.505283.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. ISBN 0521813972.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR '03*, pp. 41–47, NY, USA, 2003. ACM. ISBN 1-58113-646-3. doi: 10.1145/860435.860445. URL http://doi.acm.org/10.1145/860435.860445.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. From one point to a manifold: Knowledge graph embedding for precise link prediction. In *IJCAI*, pp. 1315–1321, 2016.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *COLING*, pp. 2335–2344, 2014. URL http://aclweb.org/anthology/C/C14/C14-1220.pdf.