

Dense Embeddings Preserving the Semantic Relationships in WordNet

1st Canlin Zhang

Circulo Health (present)

Department of Mathematics

Florida State University (sponsor)

Columbus, US

canlin.zhang@circulohealth.com

2nd Xiuwen Liu

Department of Computer Science

Florida State University

Tallahassee, US

liux@cs.fsu.edu

Abstract—In this paper, we provide a novel way to generate low dimensional vector embeddings for the noun and verb synsets in WordNet, where the hypernym-hyponym relationship is preserved in the embeddings. We call this embedding the Sense Spectrum (and Sense Spectra for embeddings). In order to create suitable labels for the training of sense spectra, we designed a new similarity measurement for noun and verb synsets in WordNet. We call this similarity measurement the Hypernym Intersection Similarity (HIS), since it compares the common and unique hypernyms between two synsets. Our experiments show that on the noun and verb pairs of the SimLex-999 dataset, HIS outperforms the three similarity measurements in WordNet. Moreover, to the best of our knowledge, the sense spectra provide the first dense synset embeddings that preserve the semantic relationships in WordNet.

Index Terms—Knowledge Representation, WordNet, Semantic Relationship, Embeddings

I. INTRODUCTION

WordNet is a lexical database for the English language [1], which groups English words into sets of synonyms called *synsets* [2]. Each synset is related to a specific semantic sense, and synsets related to the same semantic sense are usually ordered by their usage frequencies in English. There are four syntactic types of synsets in WordNet: noun (n), verb (v), adjective (a) and adverb (r). As a result, a synset in WordNet is represented in the form of “semantic sense.syntactic type.ordering”. For instance, *domestic_animal.n.01* means the first noun synset related to the semantic sense “domestic animal”, and *eat.v.03* means the third verb synset related to the semantic sense “eat”.

WordNet can be regarded as a dictionary, since it provides short definitions and usage examples for each synset. On the other hand, WordNet can also be regarded as a thesaurus [3], since it records a number of semantic relationships among synsets or their members (called *lemmas*). The most important relationship among synsets in WordNet is the Hypernym-Hyponym relationship [4], which indicates the generic term (hypernym) and a specific instance of it (hyponym). Only noun and verb synsets in WordNet possess the hypernym-hyponym relationship [5].

Since almost all the state-of-the-art Natural Language Processing (NLP) models are built on embeddings [6], [7], it is

desirable to represent the synsets in WordNet by embeddings as well. To be specific, low dimensional embeddings (dense embeddings) that can preserve the semantic relationships among synsets in WordNet are especially desired, which have not been realized. However, we note that the noun and verb synsets make up more than 80 percent of the synsets in WordNet, and the hypernym-hyponym relationship is the major relationship for nouns and verbs. Hence, it should be a good start of the research if we generate dense embeddings that preserve the hypernym-hyponym relationship for noun and verb synsets in WordNet. In this paper, we will not work with adjective or adverb synsets since they do not have the hypernym-hyponym relationship.

We first create a new similarity measurement called the Hypernym Intersection Similarity (HIS), by which the “commonness” and “differences” between two noun or verb synsets are measured according to the intersection situation of their hypernym sets. Then, using HIS as labels, we train our embedding vectors with a novel operation other than the inner product, which makes our embedding vectors look like a “spectrum of senses.” So, we call it the Sense Spectrum.

In the next section, we shall discuss the related work on creating embeddings for WordNet synsets. Then in Section 3, we shall introduce the architectures of our model. In Section 4, we will describe our implementations and provide experimental results. Then in Section 5, we will provide further discussions on our model. Finally, we will conclude the paper with a brief summary in Section 6.

II. RELATED WORK

Synset embeddings have been applied to improve the performances of NLP models on NLP tasks. For example, [8] combine word embeddings and synset embeddings to improve the performance on measuring the semantic similarity of words. Also, [9] apply a Lesk algorithm model [10] to do Word Sense Disambiguation (WSD) based on the embeddings of WordNet synsets. Almost all the NLP models evolved with WordNet represent synsets by embedding vectors.

Roughly speaking, there are two ways to create embeddings for WordNet synsets: One way is to combine the embeddings of words appeared in the definition or usage examples of

that synset, where pre-trained word embeddings from other models are required [11]. Another way is to keep each synset in one unique dimension when creating embeddings, and then create a binary matrix recording the existence (or not) of one specific semantic relationship between any two synsets (two dimensions) [12].

Synset embeddings created in the first way are dense and low dimensional, yet preserve no semantic relationships. This is because words in the definition and usage examples of a synset barely contain the information on semantic relationships. Synset embeddings created in the second way do preserve the semantic relationships, but these embeddings are sparse and have extremely high dimensions (10,000 dimensions or higher). When being used as neural network inputs, sparse and high dimensional embeddings require more weights in the low layers of neural networks, which often lead to over-fitting [13]. Hence, low dimensional embeddings that can preserve the semantic relationships among synsets in WordNet are desired, which motivates the proposed sense spectrum.

III. ARCHITECTURES

In the first subsection, we shall introduce the proposed HIS Similarity. Then, in the next subsection, we shall introduce the three basic similarity measurements in WordNet, which will be used as comparisons to our HIS similarity. After that, the formulas and training algorithms of sense spectra will be given.

Besides, we note that it is not very meaningful to compare a noun synset with a verb one. So, whenever we mention “two (noun or verb) synsets a and b ” in this paper, we assume that either both a and b are noun synsets, or both of them are verb ones.

A. Hypernym intersection similarity

Primarily, we note that WordNet not only provides the direct hypernym for each noun and verb synset, but also provides its **hyponym closure** [1]: Suppose h_1 is a direct hypernym of the synset a , and h_2 is a direct hypernym of h_1 . Then, the hypernym closure of a will contain both h_1 and h_2 . That is, the hypernym closure consists of “all the hypernyms of all the hypernyms” for the synset a , which is denoted as H_a in this paper.

For example, if we set synset a to be *man.n.01* and synset b to be *woman.n.01*, their hypernym closures H_a and H_b are then shown in Figure 1:

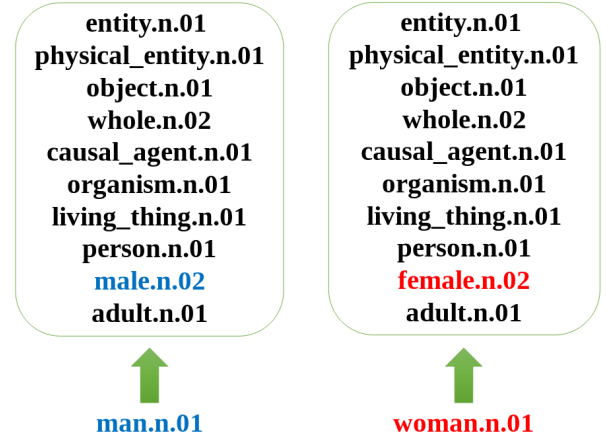


Fig. 1. (To be viewed in color) The hypernym closures of synsets *man.n.01* and *woman.n.01*.

The synset *man.n.01* denotes the common sense of “man”, whose definition in WordNet is “An adult person who is male (as opposed to a woman)”. Accordingly, *woman.n.01* is defined as “an adult female person (as opposed to a man)”. They have the same direct hypernym *adult.n.01*. We can see from Figure 1 that all the hypernyms of *man.n.01* and *woman.n.01* are the same, except that *male.n.02* is unique to *man.n.01* and *female.n.02* is unique to *woman.n.01*. Hence, the hypernym closures are the key to describe the hypernym-hyponym relationship between two synsets a and b .

However, we will not use the hypernym closure directly in the HIS Similarity. This is because the semantic field of a synset should be smaller than that of its hypernym closure [14]. Hence, based on the hypernym-hyponym relationship, the precise representation of a synset should be its hypernym closure plus the synset itself, which is the **hyponym set** $S_a = H_a \cup \{a\}$. We shall build the HIS Similarity based on the hypernym set.

Then, for two synsets a and b , we define the “commonness” between them as $S_a \cap S_b$, which can also be denoted as $S_{a \cap b}$. And the “uniqueness” of synset a is defined as $S_a / S_{a \cap b}$, which consists of the hypernyms unique to the synset a (the ones not in $S_{a \cap b}$). We denote $S_a / S_{a \cap b}$ as $S_{a/b}$. Similarly, the “uniqueness” of synset b is defined as $S_b / S_{a \cap b}$. We call $S_{a \cap b}$, $S_{a/b}$ and $S_{b/a}$ the Hypernym Representation Sets.

Taking $a = \textit{man.n.01}$ and $b = \textit{woman.n.01}$ as our example again, we can see from Figure 1 that

$$\begin{aligned} S_{a \cap b} &= \{\textit{adult.n.01}, \textit{person.n.01}, \textit{living_thing.n.01}, \\ &\quad \textit{organism.n.01}, \textit{causal_agent.n.01}, \textit{whole.n.02}, \\ &\quad \textit{object.n.01}, \textit{physical_entity.n.01}, \textit{entity.n.01}\}, \\ S_{a/b} &= \{\textit{man.n.01}, \textit{male.n.02}\}, \\ S_{b/a} &= \{\textit{woman.n.01}, \textit{female.n.02}\} \end{aligned}$$

Then, we set $\alpha = |S_{a/b}|$, $\beta = |S_{b/a}|$ and $\gamma = |S_{a \cap b}|$, which denotes the size of each hypernym representation set. Again, if $a = \textit{man.n.01}$ and $b = \textit{woman.n.01}$, we have that $\alpha = \beta = 2$ and $\gamma = 9$.

Finally, for any two noun or verb synsets a and b , the **Hypernym Intersection Similarity (HIS)** is defined as:

$$\mathcal{H}_{a,b}^{HIS} = \frac{\gamma^{0.2}}{\gamma^{0.3} + 0.5(\alpha^{0.3} + \beta^{0.3})}, \quad (1)$$

Here, the exponent parameters $\{0.2, 0.3\}$ and the scalar parameter 0.5 are designed empirically, inspired by the unigram distribution with 3/4rd power in word2vec [6].

The initial scalars α, β, γ of the HIS Similarity will be used as labels in the training of sense spectra, which is introduced in Subsection 3.3.

B. Three basic synset similarities

There are three basic measurements on the similarity between two noun or verb synsets in WordNet: The Shortest Path Similarity, Leacock-Chodorow Similarity and Wu-Palmer Similarity [15]. They are “basic” since they only require the hypernym-hyponym relationship between two synsets [16], which is the same as the HIS Similarity. Hence, they are used as the comparisons to our model.

- **Shortest Path Similarity:** All the noun synsets share the same root hypernym *entity.n.01*. But there may be no common hypernym between two verb synsets. So, a fake root synset *root.v.01* is added to the verb synsets.

Then, for any two noun or verb synsets a and b , there is always a hypernym-hyponym path connecting them through a common hypernym of them. And there is a shortest path among all these paths, whose length is denoted as $l_{a,b}$. The Shortest Path Similarity between synsets a and b is then defined to be $\mathcal{H}_{a,b}^{S.P.} = 1/l_{a,b}$, which is between 0 and 1.

- **Leacock–Chodorow Similarity:** The *depth* of a noun or verb synset a , denoted as d_a , is defined to be the length of the shortest path from a to the root synset (*entity.n.01* for noun and *root.v.01* for verb). That is, $d_a = l_{a,root}$.

Then, for two synsets a and b , the Leacock-Chodorow (LCH) Similarity is defined as

$$\mathcal{H}_{a,b}^{LCH} = -\log \frac{l_{a,b}}{2 \cdot \max\{d_a, d_b\}}.$$

- **Wu–Palmer Similarity:** For two noun or verb synsets a and b , their *Least Common Subsumer (LCS)* is the common hypernym of a and b with the largest depth. We use $\hat{h}_{a,b}$ or simply \hat{h} to denote the LCS of synsets a and b . That is, $\hat{h} = \max_{h \in S_{a \cap b}} \{d_h\}$. Then, the Wu-Palmer (WP) Similarity between synsets a and b is defined as $\mathcal{H}_{a,b}^{WP} = 2\hat{h}/(d_a + d_b)$.

C. Sense spectrum

Suppose v_a and v_b are the embedding vectors of synsets a and b respectively. Then, we use the “overlapping” between v_a and v_b to represent the “commonness” between synsets a and b . The overlapping of two vectors is measured dimension-wise: Suppose v_{ai} and v_{bi} are the elements in the i ’th dimension of v_a and v_b respectively. We use $v_{ai} \cap v_{bi}$ to represent the overlapping between v_{ai} and v_{bi} . Then, if v_{ai} and v_{bi} have the same sign (i.e., both of them are positive or both are negative), $v_{ai} \cap v_{bi}$ will equal to the one of v_{ai} and v_{bi} with the smaller

absolute value. If v_{ai} and v_{bi} have different signs, $v_{ai} \cap v_{bi}$ will be zero. That is, mathematically:

$$v_{ai} \cap v_{bi} = \frac{\text{sgn}(v_{ai}) + \text{sgn}(v_{bi})}{2} \cdot \min\{|v_{ai}|, |v_{bi}|\}. \quad (2)$$

where $\text{sgn}(x)$ is the sign function:

$$\text{sgn}(x) = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ 1, & \text{if } x > 0. \end{cases}$$

Taking this operation to each dimension i , we can get the overlapping vector $v_a \cap v_b$ by $\{v_a \cap v_b\}_i = v_{ai} \cap v_{bi}$. We also denote $v_a \cap v_b$ as $v_{a \cap b}$, which can be regarded as the vector representation on the “commonness” between synsets a and b . After obtaining $v_{a \cap b}$, the vector representation on the “differences (uniqueness)” of synsets a and b is obvious: We use $v_{a/b} = v_a - v_{a \cap b}$ to represent the “uniqueness” of synset a , and use $v_{b/a} = v_b - v_{a \cap b}$ to represent the “uniqueness” of synset b .

We can see that each dimension in v_a and v_b operates independently to form $v_{a \cap b}$, $v_{a/b}$ and $v_{b/a}$. This makes our embedding vector looks like a “spectrum”, with its dimensions to be the measurements on specific senses. In fact, this is verified by experiments, which will be discussed in Section 5. As a result, we call our synset embedding the **Sense Spectrum**. For any two synsets a and b , we call $v_{a \cap b}$ the Commonness Spectrum, and we call $v_{a/b}$, $v_{b/a}$ the Uniqueness Spectra. Figure 2 provides a clear exhibition on how to obtain $v_{a \cap b}$, $v_{a/b}$ and $v_{b/a}$ based on the initial spectra v_a and v_b .

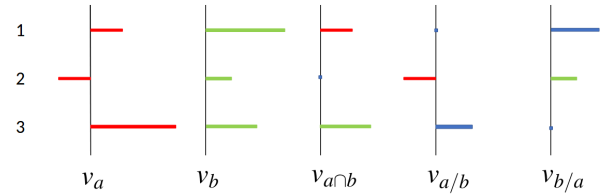


Fig. 2. (To be viewed in color) The commonness spectrum $v_{a \cap b}$ as well as the uniqueness spectra $v_{a/b}$ and $v_{b/a}$, based on the initial sense spectra v_a and v_b .

In Figure 2, we suppose a spectrum vector is dimension-three. We show v_a in red and v_b in green. According to our overlapping method, the dimension 1 of $v_{a \cap b}$ is the same as that of v_a , which is also in red. Similarly, dimension 3 of $v_{a \cap b}$ is the same as that of v_b , which is in green. But v_a and v_b are not overlapped in dimension 2, making that of $v_{a \cap b}$ to be zero. Hence, dimension 2 of $v_{a/b}$ and $v_{b/a}$ shall remain the same as in v_a and v_b respectively, since no cancellation is made from $v_{a \cap b}$. Finally, dimension 1 of $v_{a/b}$ and dimension 3 of $v_{b/a}$ are cancelled to zero. But dimension 3 of $v_{a/b}$ and dimension 1 of $v_{b/a}$ are partially cancelled, which is in blue.

It is then only straightforward to figure out that the three spectra $v_{a \cap b}$, $v_{a/b}$ and $v_{b/a}$ coincide with the initial HIS scalars γ , α and β : The commonness spectrum $v_{a \cap b}$ coincides with the scalar γ , while the two uniqueness spectra $v_{a/b}$ and $v_{b/a}$

coincide with scalars α and β , respectively. Hence, our training algorithm is as simple as:

$$\|v_{a \cap b}\|_1 = \gamma, \quad \|v_{a/b}\|_1 = \alpha, \quad \|v_{b/a}\|_1 = \beta, \quad (3)$$

where $\|\cdot\|_1$ is the L_1 norm of a vector [17].

We will show in the next section that after training, the hypernym-hyponym relationship between two noun or verb synsets a and b is preserved in their corresponding spectra v_a and v_b .

IV. EVALUATION

In this section, we show by experimental results that the Hypernym Intersection Similarity outperforms the three basic similarity measurements in WordNet. And we will show that Sense Spectra indeed capture the structures of the hypernym-hyponym relationship in WordNet.

A. The performance of HIS

To estimate the performance of HIS Similarity, we use the dataset SimLex-999 [18], which contains 666 noun pairs, 222 verb pairs and 111 adjective pairs. Each pair of words in SimLex-999 is scored from 0 to 10: The higher the score is, the more similar the two words in that pair should be. All the scores are given manually by native English speakers. Table 1 provides a brief exhibition on the noun and verb pairs in SimLex-999.

TABLE I
THE NOUN AND VERB PAIRS AS WELL AS THEIR CORRESPONDING SCORES IN SIMLEX-999

Noun pairs		Score	Verb pairs		Score
book	text	6.35	listen	hear	8.17
night	day	1.88	go	come	2.42
belief	flower	0.40	spend	save	0.55
...

However, there may be more than one synset related to a word [19]. For example, there are 11 noun synsets related to the word “book”, including *book.n.01* (a written work or composition that has been published), *book.n.02* (physical objects consisting of a number of pages bound together), *bible.n.01* (the sacred writings of the Christian religions), etc. So, we need to first choose the correct synsets for each word pair in SimLex-999.

For a word pair (w_1, w_2) , suppose there are M synsets $\{s_1, \dots, s_M\}$ related to w_1 , and N synsets $\{s'_1, \dots, s'_N\}$ related to w_2 . Then, there are $M \times N$ possible combinations of synsets for the word pair (w_1, w_2) : $\{(s_1, s'_1), \dots, (s_1, s'_N), \dots, (s_M, s'_1), \dots, (s_M, s'_N)\}$. We compute the HIS Similarity $\mathcal{H}_{s_m, s'_n}^{HIS}$ by formula (1) on each synset combination (s_m, s'_n) , and then choose the combination (\hat{s}_m, \hat{s}'_n) with the maximal similarity score $\mathcal{H}_{\hat{s}_m, \hat{s}'_n}^{HIS}$. After that, (\hat{s}_m, \hat{s}'_n) is regarded as the correct synset choice for the word pair (w_1, w_2) , and $\mathcal{H}_{\hat{s}_m, \hat{s}'_n}^{HIS}$ is regarded as the similarity score of (w_1, w_2) under the HIS Similarity, denoted as $\mathcal{H}_{w_1, w_2}^{HIS}$.

Finally, suppose $\{(w_1^k, w_2^k)\}_{k=1}^K$ is a specific set of word pairs in SimLex-999 (say, all the noun pairs). For each word pair (w_1^k, w_2^k) , suppose \mathcal{S}_k is the manually given similarity score in SimLex-999, and $\mathcal{H}_k := \mathcal{H}_{w_1^k, w_2^k}^{HIS}$ is the similarity score under HIS Similarity. Then, we compute the Spearman’s correlation [20] between $\{\mathcal{S}_k\}_{k=1}^K$ and $\{\mathcal{H}_k\}_{k=1}^K$ as:

$$\rho = \frac{\sum_{k=1}^K (\mathcal{S}_k - \overline{\mathcal{S}})(\mathcal{H}_k - \overline{\mathcal{H}})}{\sqrt{\sum_{k=1}^K (\mathcal{S}_k - \overline{\mathcal{S}})^2 \sum_{k=1}^K (\mathcal{H}_k - \overline{\mathcal{H}})^2}}, \quad (4)$$

where $\overline{\mathcal{S}}$ and $\overline{\mathcal{H}}$ are the averages of $\{\mathcal{S}_k\}_{k=1}^K$ and $\{\mathcal{H}_k\}_{k=1}^K$ respectively. This Spearman’s correlation ρ is then the estimation on the performance of the HIS Similarity. A higher Spearman’s correlation here means that the language model can handle the semantic meanings of words more like humans do [21].

In order to obtain comparisons, we apply the same process onto the Shortest Path Similarity, Leacock-Chodorow (LCH) Similarity and Wu-Palmer (WP) Similarity. That is, we replace the similarity score $\mathcal{H}_{s_m, s'_n}^{HIS}$ with $\mathcal{H}_{s_m, s'_n}^{s.p.}$, $\mathcal{H}_{s_m, s'_n}^{LCH}$ and $\mathcal{H}_{s_m, s'_n}^{WP}$ as described in Section 3.2 to get the corresponding Spearman’s correlation $\rho_{s.p.}$, ρ_{LCH} and ρ_{WU} , respectively. Moreover, we work on three different sets of word pairs in SimLex-999: only noun pairs, only verb pairs, or combining both noun and verb ones. Results are shown in Table 2.

TABLE II
THE SPEARMAN’S CORRELATIONS OBTAINED BY PERFORMING EACH SIMILARITY MEASUREMENT ON DIFFERENT SETS OF WORD PAIRS IN SIMLEX-999.

Model \ Group	Noun pairs	Verb pairs	Both
HIS	61.12	48.38	55.98
Shortest Path	58.38	39.20	51.96
LCH	58.38	39.20	54.92
WP	55.00	37.84	48.82

We can see that the HIS Similarity achieves the highest Spearman’s correlation on all the three sets of word pairs. To be specific, on the verb pairs, the HIS Similarity outperforms the other three similarity measurements by 10 percent, which is a significant improvement.

Therefore, we claim that the HIS Similarity captures the hypernym-hyponym relationship in WordNet better than the three basic similarities do. Hence, it is meaningful to use the initial HIS scalars α , β , γ as labels to train our sense spectra, whose performance is given in the following subsection.

B. The performance of sense spectra

Again, we note that it is meaningless to compare a noun synset with a verb one. So, the noun and verb spectra are generated and trained independently: There are 82,115 noun synsets, whose spectra are generated as v_1, \dots, v_{82115} ; And there are 13,767 verb synsets, whose spectra are v'_1, \dots, v'_{13767} . We always set the dimension of a spectrum to be $D = 200$ for both noun and verb synsets.

When training the sense spectra, we use TensorFlow in Python [22]. We shall first introduce our methods of implementations. Then, we shall introduce our specific strategy on how to build a training batch. Finally, meaningful testing results will be given.

1) *Implementation issues:* When computing the dimension-wise overlapping $v_{ai} \cap v_{bi}$, we realize that it is difficult to perform formula (2) directly in TensorFlow. This is because errors cannot path through the sign function $\text{sgn}(x)$ by back propagation [23]. Besides, there is no necessary to generate $v_{a \cap b}$ by each dimension in practice. So, we use a formula evolved with the rectifier function [24] $\text{ReLU}(x) = \max\{0, x\}$ to compute $v_{a \cap b}$ directly:

$$v_{a \cap b} = \min(\text{ReLU}(v_a), \text{ReLU}(v_b)) \\ - \min(\text{ReLU}(-v_a), \text{ReLU}(-v_b))$$

To be specific, suppose the dimension of a spectrum vector is D . We first apply *tf.concat* to concatenate the rectified vectors $\text{ReLU}(v_a)$ and $\text{ReLU}(v_b)$ along each dimension, which returns a $D \times 2$ tensor (matrix) c_+ . After that, we apply *tf.reduce_min* to get the minimum value on each dimension of c_+ , which returns a D dimensional vector t_+ . Similarly, we can get c_- and t_- with respect to $\text{ReLU}(-v_a)$ and $\text{ReLU}(-v_b)$. Then, we have that $v_{a \cap b} = t_+ - t_-$.

The formulas to obtain $v_{a/b}$ and $v_{b/a}$ in practice are much more straightforward:

$$v_{a/b} = \text{ReLU}(\text{ReLU}(v_a) - \text{ReLU}(v_b)) \\ - \text{ReLU}(\text{ReLU}(-v_a) - \text{ReLU}(-v_b)), \\ v_{b/a} = \text{ReLU}(\text{ReLU}(v_b) - \text{ReLU}(v_a)) \\ - \text{ReLU}(\text{ReLU}(-v_b) - \text{ReLU}(-v_a)).$$

After that, we compute the L_1 norm of a D dimensional vector v as $\|v\|_1 = \sum_{d=1}^D |v_d| = \text{tf.reduce_sum}(\text{tf.abs}(v))$.

Finally, suppose $\|v_{a/b}\|_1 = \hat{\alpha}$, $\|v_{b/a}\|_1 = \hat{\beta}$ and $\|v_{a \cap b}\|_1 = \hat{\gamma}$. Applying the initial HIS scalars α, β, γ as labels, we complete the training by minimizing the error $|\alpha - \hat{\alpha}| + |\beta - \hat{\beta}| + |\gamma - \hat{\gamma}|$ via AdamOptimizer [25].

2) *Batch formation strategies:* Each batch in our model consists of a synset pair (a, b) , which is formed in three different ways:

- The direct hypernym pair: After choosing a synset a randomly, we pick its direct hypernym h_a to form a pair (a, h_a) . If there are more than one direct hypernyms for the synset a , we shall choose one of them randomly.

- The semantic sense related pair: As we mentioned in the introduction, each synset is related to a specific semantic sense. There are 67,176 noun semantic senses and 7,440 verb semantic senses that have more than one related synsets. In the training, we shall randomly pick one semantic sense and randomly choose two of its related synsets to form a semantic sense related pair. That is, suppose we get the semantic sense S and its related synsets $\{s_1, \dots, s_k\}$. Then, we randomly pick two synsets \hat{s}_1, \hat{s}_2 from $\{s_1, \dots, s_k\}$ to form the pair.

- Random pair: We choose two synsets a and b randomly to form the pair.

We have T pairs built in each of these three ways. So, our total batch size is $3T$. We always set $T = 100$ in our training. And again, we note that noun and verb synset pairs are formed independently.

3) *Testing results:* After training, we look up the three closest spectra for each spectrum under the HIS Similarity. That is, for a spectrum vector v_a related to the synset a , we compute $\|v_{a/b}\|_1 = \hat{\alpha}$, $\|v_{b/a}\|_1 = \hat{\beta}$ and $\|v_{a \cap b}\|_1 = \hat{\gamma}$ with respect to every else noun (or verb) spectrum v_b . Then, we apply formula (1) on each scalar set $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})_b$ to find the three synsets b_1, b_2 and b_3 that provide the maximal values $\mathcal{K}_{a,b}^{HIS}$. Again, we note that this procedure is performed on the noun and the verb spectra independently. Some results are shown in Table 3.

Table 3 contains the synsets related to both commonly used words and specific terminologies. To be specific, synsets *cynoglossum.n.01*, *genus_martynia.n.01* and *physostigma.n.01* represents three different genera of plants, to which *myosotis.n.01* belongs; The synset *stub.n.03* is defined as “a torn part of a ticket returned to the holder as a receipt”; And the synset *break_up.v.19* means “laugh unrestrainedly”. By these examples, we can see that sense spectra with similar meanings (corresponding to their synsets) are clustered together under the HIS Similarity, which is similar to the performance of word2vec [6]. This result shows that sense spectra to some extent preserve the hypernym-hyponym relationship in WordNet after training.

However, one may ask: To what extent, or how precise, can sense spectra preserve the hypernym-hyponym relationship? In order to answer this question, we pick a pair of synsets and their corresponding spectra (a, b, v_a, v_b) , and then compare the initial HIS scalars $(\alpha, \beta, \gamma)_{(a,b)}$ with the spectrum-based HIS scalars $(\hat{\alpha} = \|v_{a/b}\|_1, \hat{\beta} = \|v_{b/a}\|_1, \hat{\gamma} = \|v_{a \cap b}\|_1)$ by

$$R_{(a,b)} = \frac{|\alpha - \hat{\alpha}| + |\beta - \hat{\beta}| + |\gamma - \hat{\gamma}|}{\alpha + \beta + \gamma}. \quad (5)$$

The numerator of formula (5) represents the error made by the sense spectra, while the denominator represents the magnitude of the initial HIS scalars. Hence, the smaller $R_{(a,b)}$ is, the less important the error is comparing to the initial HIS scalars, and hence the more precise sense spectra v_a, v_b can capture the hypernym-hyponym relationship between their corresponding synsets a, b .

We perform formula (1) on SimLex-999 dataset. That is, we obtain the synset pair (\hat{s}_m, \hat{s}'_n) for each noun and verb pair (w_1, w_2) in SimLex-999, and then compute $R_{(\hat{s}_m, \hat{s}'_n)}$ on each $(\hat{s}_m, \hat{s}'_n, v_{\hat{s}_m}, v_{\hat{s}'_n})$. The distribution of the pairs $(\hat{s}_m, \hat{s}'_n, v_{\hat{s}_m}, v_{\hat{s}'_n})$ based on the values $R_{(\hat{s}_m, \hat{s}'_n)}$ is observed by the histograms in Figure 3 and the statistical results in Table 4.

TABLE III
THE THREE CLOSEST SPECTRA FOR EACH SENSE SPECTRUM UNDER THE HIS SIMILARITY.

Synset a	Synset b_1	$\mathcal{H}_{a,b_1}^{HIS}$	Synset b_2	$\mathcal{H}_{a,b_2}^{HIS}$	Synset b_3	$\mathcal{H}_{a,b_3}^{HIS}$
<i>trade.n.01</i>	<i>fair_trade.n.02</i>	8.04	<i>fair_trade.n.01</i>	8.03	<i>free_trade.n.01</i>	7.59
<i>finance.n.01</i>	<i>flotation.n.02</i>	8.98	<i>banking.n.02</i>	8.90	<i>high_finance.n.01</i>	8.88
<i>war.n.01</i>	<i>jihad.n.01</i>	7.06	<i>hot_war.n.01</i>	7.02	<i>world_war.n.01</i>	7.01
<i>vent.n.01</i>	<i>smoke_hole.n.01</i>	6.80	<i>bunghole.n.02</i>	6.32	<i>air_hole.n.02</i>	6.29
<i>receipt.n.02</i>	<i>bill_of_lading.n.01</i>	4.99	<i>pawn_ticket.n.01</i>	4.87	<i>stub.n.03</i>	4.55
<i>myosotis.n.01</i>	<i>cynoglossum.n.01</i>	6.22	<i>genus_martynia.n.01</i>	6.21	<i>physostigma.n.01</i>	6.20
<i>cough.v.01</i>	<i>hack.v.08</i>	1.91	<i>clear_the_throat.v.01</i>	1.87	<i>expectorate.v.02</i>	0.93
<i>laugh.v.01</i>	<i>snicker.v.01</i>	0.98	<i>break_up.v.19</i>	0.97	<i>cackle.v.03</i>	0.97
<i>coach.v.01</i>	<i>condition.v.01</i>	4.00	<i>mentor.v.01</i>	3.93	<i>reinforce.v.02</i>	3.92
<i>propose.v.01</i>	<i>submit.v.02</i>	2.99	<i>recommend.v.01</i>	2.93	<i>advance.v.02</i>	2.93

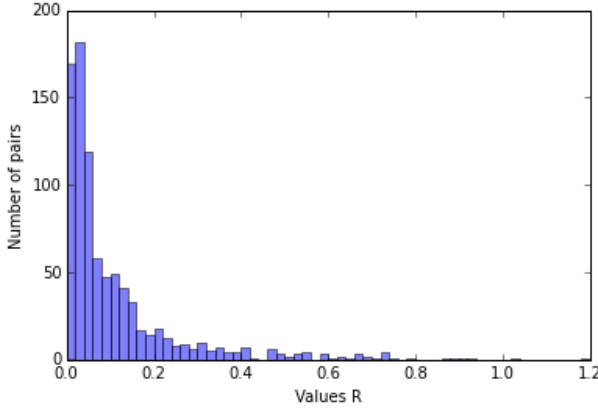


Fig. 3. The histogram distribution of the pairs $(\hat{s}_m, \hat{s}'_n, v_{\hat{s}_m}, v_{\hat{s}'_n})$ based on the Values $R_{(\hat{s}_m, \hat{s}'_n)}$.

TABLE IV
THE STATISTICAL DISTRIBUTION OF THE PAIRS $(\hat{s}_m, \hat{s}'_n, v_{\hat{s}_m}, v_{\hat{s}'_n})$ BASED ON THE VALUES $R_{(\hat{s}_m, \hat{s}'_n)}$.

Range of $R_{(\hat{s}_m, \hat{s}'_n)}$	< 0.05	< 0.1	< 0.2	< 0.3
Percentage of Pairs	47.81	66.24	83.99	90.09

We can see from Figure 3 and Table 4 that more than 90% of the pairs $(\hat{s}_m, \hat{s}'_n, v_{\hat{s}_m}, v_{\hat{s}'_n})$ have a value $R_{(\hat{s}_m, \hat{s}'_n)}$ significantly less than one. That is, in most SimLex-999 pairs $(\hat{s}_m, \hat{s}'_n, v_{\hat{s}_m}, v_{\hat{s}'_n})$, the sense spectra $v_{\hat{s}_m}, v_{\hat{s}'_n}$ capture the hypernym-hyponym relationship between the synsets \hat{s}_m, \hat{s}'_n precisely. Taking the authority and complexity of the SimLex-999 dataset, we claim that in general, our sense spectra capture the hypernym-hyponym relationships among their corresponding synsets precisely. To the best of our knowledge, this is the first time that low dimensional embeddings can preserve the semantic relationships among synsets in WordNet.

Finally, in Figure 4, we plot the spectra for the synsets in Table 3 that are related to *trade.n.01* (row 1) and *cough.v.01* (row 7). Different from vertical spectra in Figure 2, we plot horizontal spectra here to save spaces.

We can see from Figure 4 that verb spectra are in general sparser than noun spectra. This is because the hypernym-hyponym relationships among verb synsets are more concise

comparing to those among noun synsets. Hence, fewer dimensions in a spectrum are enough to preserve the semantic information. In addition, spectra with similar meanings (corresponding to their synsets) also have similar distributions across dimensions. That is, in most dimensions, spectra with similar meanings tend to have the same sign and the same magnitude. These phenomena are in fact meaningful, which will be discussed in the next section.

Reproducibility: Our code can be accessed via github. The link will be provided upon acceptance.

V. DISCUSSIONS

In this section, we shall further discuss the meaning behind our experimental results, based on which we shall describe about the potential applications of the sense spectra.

A. Building hierarchical language model based on sense spectra

As we mentioned in the previous section, sense spectra with similar meanings tend to have the same sign and magnitude in most dimensions. However, since the number of dimensions in a spectrum is far less than the number of (noun or verb) synsets, it is impossible for each dimension to preserve semantic information independently. As a result, we can conclude that specific combinations of dimensions in a sense spectrum work together to preserve specific semantic information. That is, there exists structures related to semantic senses among the dimensions in sense spectra, indicating the name Sense Spectrum is fair and genuine.

Then, combining with text training corpus [26], it is possible to build hierarchical language model based on the structures in sense spectra. For example, we may first group together the dimensions that co-occur frequently across noun or verb spectra. These dimension groups should be highly related to the dimension combinations preserving specific semantic information. Then, for each word in the training corpus, we may find all the related synsets and put the corresponding spectra into a list. After that, we will have a list of the possible spectra for each word in the training corpus. Finally, based on this, we may discover the “groups of dimension groups.” That is, we further group together the dimension groups that co-occur frequently in the training corpus. In this way, a hierarchical language model with explicit upper layer units can be built.

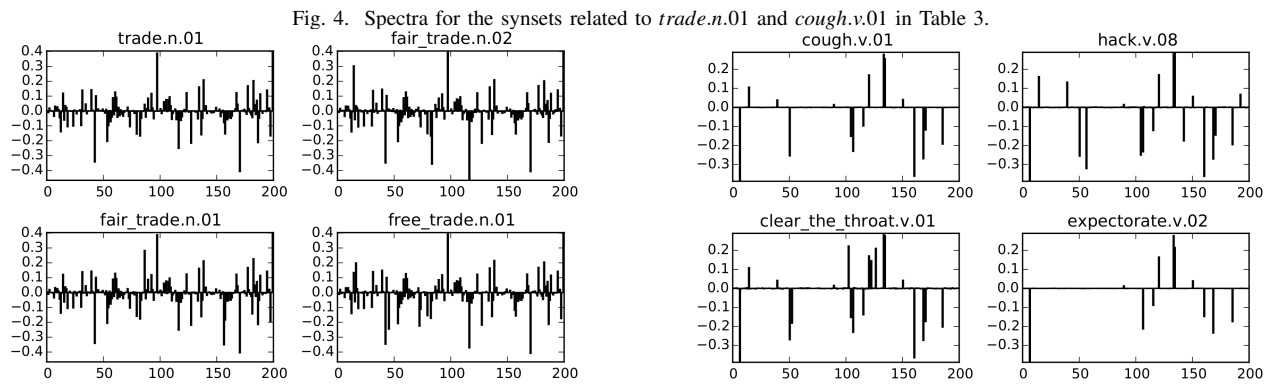


Fig. 4. Spectra for the synsets related to *trade.n.01* and *cough.v.01* in Table 3.

B. Combining sense spectra with word embeddings

Now that sense spectra are low dimensional and dense, we can directly concatenate them to the pre-trained word embeddings. Similar to the above discussion, for each word w in the training corpus, we may find its related synsets. Then, we pick out the corresponding spectra of these synsets and perform the average summation over the spectra to get a summation vector. Finally, we concatenate the summation vector to the pre-trained embedding vector of the word w .

In this way, the embedding vectors now contain not only the contextual information obtained from the corpus-based training, but also the semantic relationship information obtained from the knowledge-based training [27]. We believe that such word embeddings are promising for tasks such as Word Sense Disambiguation (WSD) [28] and Outlier Detection [29], where information about semantic relationships are highly demanded. This is being investigated.

VI. CONCLUSION

We provide sense spectra, which are the first dense and low-dimension embeddings for the noun and verb synsets in WordNet that preserve the hypernym-hyponym relationships among the synsets. We train sense spectra by HIS Similarity, which is a similarity measurement describing the “commonness” and “uniqueness” between two noun or verb synsets in WordNet.

Results show that the HIS Similarity outperforms the three basic similarity measurements in WordNet on the SimLex-999 noun and verb pairs, and sense spectra do preserve the hypernym-hyponym relationship among synsets precisely. Novel applications built on sense spectra are described and are being actively explored.

REFERENCES

- [1] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, and K. Miller, “Wordnet: An online lexical database,” *International Journal of Lexicography*, pp. 235–244, 1990.
- [2] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [3] J. Boyd-Graber, F. Fellbaum, D. Osherson, and R. Schapire, “Adding dense, weighted connections to wordnet,” *In proceedings of the third global WordNet meeting*, 2006.
- [4] I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. D. Saeger, F. Bond, and A. Sumida, “Hypernym discovery based on distributional similarity and hierarchical structures,” *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 929–937, 2009.
- [5] G. A. Miller and F. Hristea, “Wordnet nouns: Classes and instances,” *Computational Linguistics*, vol. 32, no. 1, pp. 1–3, 2006.
- [6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint, arXiv:1810.04805*, 2018.
- [8] G. Recski, E. Iklodi, K. Pajkossy, and A. Kornai, “Measuring semantic similarity of words using concept networks,” *In Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016.
- [9] S. Banerjee and T. Pedersen, “An adapted lesk algorithm for word sense disambiguation using wordnet,” *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, 2002.
- [10] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone,” *The 5th annual international conference on Systems documentation*, pp. 24–26, 1986.
- [11] S. Rothe and H. Schutze, “Autoextend: Extending word embeddings to embeddings for synsets and lexemes,” *In Proceedings of the 53rd ACL*, vol. volume 1, pp. 1793–1803, 2015.
- [12] C. Saedi, A. Branco, J. A. Rodrigues, and J. Silva, “Wordnet embeddings,” *In Proceedings of The Third Workshop on Representation Learning for NLP at Association for Computational Linguistics*, pp. 122–131, 2018.
- [13] R. Rojas, “The curse of dimensionality,” *arxiv.org/pdf/1507.01127*, 2015.
- [14] C. Gao and B. Xu, “The application of semantic field theory to english vocabulary learning,” *Theory and Practice in Language Studies*, vol. 3, no. 11, 2013.
- [15] T. Slimani, “Description and evaluation of semantic similarity measures approaches,” *Journal of Computer Applications*, vol. volume 80, no. 10, pp. pages 1–10, 2013.
- [16] D. S. Jones, *Elementary Information Theory*. Oxford, UK: Clarendon Press, 1979.
- [17] J. Cape, M. Tang, and C. E. Priebe, “The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics,” *arXiv preprint, arxiv.org/pdf/1507.01127*, 2017.
- [18] F. Hill, R. Reichart, and A. Korhonen, “Simlex-999: Evaluating semantic models with (genuine) similarity estimation,” *Computational Linguistics*, vol. volume 41, 2015.
- [19] R. Navigli, “Word sense disambiguation: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 2, p. 10, 2009.
- [20] J. H. McDonald, *Handbook of biological statistics*. Baltimore: Sparky House, 2009.
- [21] J. C. de Winter, S. D. Gosling, and J. Potter, “Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data,” *Psychological Methods*, vol. 21, no. 3, pp. 273–290, 2016.

- [22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, and Z. Chen, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint, arXiv:1603.04467*, 2016.
- [23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
- [24] A. F. Agarap, "Deep learning using rectified linear units (relu)," *Computing Research Repository (CoRR)*, 2018.
- [25] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *arXiv preprint, arXiv:1412.6980*, 2014.
- [26] V. Liu and J. R. Curran, "Web text corpus for natural language processing," *European Chapter of the Association for Computational Linguistics (ACL-EACL)*, 2006.
- [27] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt, "Representation learning for very short texts using weighted word embedding aggregation," *Pattern Recognition Letters*, pp. 150–156, 2016.
- [28] D. Yarowsky, "Hierarchical decision lists for word sense disambiguation," *Computers and the Humanities*, vol. 34, no. 1, pp. 179–186, Apr 2000.
- [29] C. C. Aggarwal, *Outlier Analysis. Second Edition*. New York: Springer, 2016.