Learning Contextualized Box Embeddings with Prototypical Networks

Kohei Oda **Kiyoaki Shirai** **Natthawut Kertkeidkachorn**
Japan Advanced Institute of Science and Technology
{s2420017,kshirai,natt}@jaist.ac.jp

Abstract

This paper proposes ProtoBox, a novel method to learn contextualized box embeddings. Unlike an ordinary word embedding, which represents a word as a single vector, a box embedding represents the meaning of a word as a box in a high-dimensional space: that is suitable for representing semantic relations between words. In addition, our method aims to obtain a "contextualized" box embedding, which is an abstract representation of a word in a specific context. ProtoBox is based on Prototypical Networks, which is a robust method for classification problems, especially focusing on learning the hypernym–hyponym relation between senses. ProtoBox is evaluated on three tasks: Word Sense Disambiguation (WSD), New Sense Classification (NSC), and Hypernym Identification (HI). Experimental results show that ProtoBox outperforms baselines for the HI task and is comparable for the WSD and NSC tasks.[1]

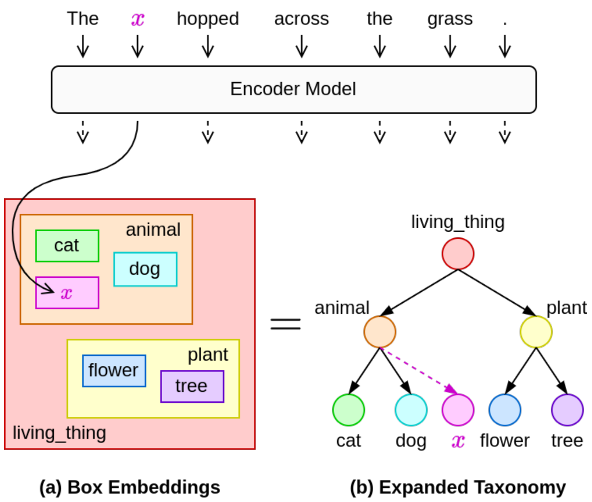

Figure 1: Example of contextualized box embedding and its application to New Sense Classification and Hypernym Identification.

1 Introduction

Word embedding is an abstract representation of a word, usually as a vector in a high dimensional space. Nowadays, word embeddings are widely used in models based on deep learning. Word embedding can represent the meaning not only of a word itself (Mikolov et al., 2013) but also of a word in a context. For example, BERT (Devlin et al., 2019) is often used to obtain vector representations of words in a given sentence. In this paper, we call such word embeddings "contextualized word embeddings." In addition, box embedding (Dasgupta et al., 2020) is a kind of word embedding, which represents a word not with a point but with an area in vector space. While an ordinary word embedding is primary used to measure the similarity between two words, box embeddings can be used to capture other semantic relations between words such as that between a hypernym and a hyponym. However, past studies of box embeddings did not well consider the context of the word, that is, the box embedding was not contextualized.

Contextualized word embeddings can be regarded as "sense embeddings," since a word may have two or more senses and convey one of those possible senses in a specific context. Word Sense Disambiguation (WSD) (Navigli, 2009) is a task that aims to identify the meaning of a word in a context. Most previous work on WSD has focused only on the senses in a predefined inventory and has ignored new (not predefined) senses. However, senses of words change day by day and new senses are constantly created (Yu and Xu, 2023). It is preferable that a WSD system can handle a new sense by classifying a word even if it is being used in a new sense, and, if possible, explaining the meaning of the new sense.

In this paper, we propose ProtoBox, a method to produce a contextualized box embedding of a word in a given context. In general, box embeddings can represent hypernym–hyponym relations between

[1] Our code is available at: https://github.com/iehok/ProtoBox.

Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024), pages 1–12
August 15, 2024 ©2024 Association for Computational Linguistics

words, as illustrated in Figure 1 (a). If a box of word w_a subsumes the box of another word w_b, w_a can be regarded as a hypernym of w_b. Such relations between words can be represented as taxonomy (Figure 1 (b)). Our ProtoBox can produce contextualized box embeddings. For example, a box embedding of x in the sentence "The x hopped across the grass." can be obtained as shown in Figure 1 (a). Contextualized box embeddings enables us to judge that x has a new sense when the box embedding of x does not overlap any other box embeddings of fine-grained senses such as "cat" and "dog". In addition, "animal" can be identified as a hypernym of x, since the box embedding of x is subsumed by that of "animal". Identification of a hypernym can provide a rough explanation of a new sense, i.e., x is a kind of an animal. Furthermore, ProtoBox can expand the existing taxonomy by adding a new node x to the structure as shown in Figure 1 (b).

We evaluate ProtoBox with three tasks: WSD, New Sense Classification (NSC), and Hypernym Identification (HI). Three datasets of different domains, one is large and two are small, are used to thoroughly evaluate our proposed method. Experimental results show that ProtoBox is better than or comparable to the baselines for WSD and NSC, and always outperforms the baselines for HI.

The contributions of this paper are summarized as follows:

- We propose ProtoBox, a new method to learn contextualized box embeddings based on Prototypical Networks (Snell et al., 2017).
- We propose a method to construct an mini-batch to learn hypernym–hyponym relations between senses in the contextualized box embeddings.
- We empirically evaluate the effectiveness of ProtoBox for three down-streaming tasks: WSD, NSC, and HI.

2 Related Work

2.1 WSD

Many recent WSD methods use glosses (sense definitions) and lexical relations (e.g., hypernym–hyponym relations) to improve their performance (Huang et al., 2019; Kumar et al., 2019; Bevilacqua et al., 2020; Bevilacqua and Navigli, 2020; Blevins and Zettlemoyer, 2020; Scarlini et al., 2020; Barba et al., 2021). However, the accuracy of WSD for infrequent senses tended to be lower than that for the whole of the test data (Maru et al., 2022).

To address this problem, Chen et al. (2021) proposed MetricWSD, a method to learn contextualized embeddings using Prototypical Networks (Snell et al., 2017). Prototypical Networks is a meta learning method that works better on imbalanced data. MetricWSD achieved a state-of-the-art WSD performance without additional lexical information such as glosses or lexical relations.

Generationary (Bevilacqua et al., 2020) is another approach for WSD. First, a definition of a sense for a given word is generated by BART (Lewis et al., 2020). Then, the similarity score between the generated definition and each definition of the target word in WordNet (Miller, 1995) is calculated by Sentence-BERT (Reimers and Gurevych, 2019) and the most similar sense chosen to be the predicted sense. Generationary aims not only to improve the performance at WSD but also explain a new sense. This paper also tries to explain the meaning of a new sense using trained contextualized box embeddings. Instead of generating a definition of a new sense, a hypernym of a new sense is identified as a coarse meaning of it.

2.2 Taxonomy Expansion

Taxonomy Expansion is the task to infer a hypernym of a new concept (Bordea et al., 2016). It has been actively studied. Recent methods improved the performance by using graph neural networks (Shen et al., 2020) and learning the shortest path between a target concept and the root concept (Yu et al., 2020). Some methods (Aly et al., 2019; Ma et al., 2021) used Hyperbolic space (Nickel and Kiela, 2017) learn hypernym–hyponym relations. This paper also presents a method of Taxonomy Expansion, but a hypernym of a new concept is guessed by contextualized box embeddings.

2.3 Contextualized Box Embeddings

There have been a few studies that have applied contextualized box embeddings to some tasks. The Entity Typing task is a multi-label classification problem to predict appropriate types such as "event" and "person", for a target in a context (Choi et al., 2018). Onoe et al. (2021) represented target entities by contextualized box embeddings, and also represented types of entity by dedicated box embeddings. The model was trained so that the contextualized box embedding of the target entity was enclosed by the box embeddings of its type of entity.

Jiang et al. (2023) proposed a method for Taxonomy Expansion by learning the box embeddings of concepts. The box embeddings of entities in the existing taxonomy were derived from their definition sentences. The model, which converts a sentence to a contextualized box embedding was trained by hypernym–hyponym pairs in the taxonomy so that the box embedding of a hypernym enclosed that of a hyponym. Although the above studies presented methods to learn contextualized box embeddings, we adapt another approach. Specifically, our framework follows that of Prototypical Networks, which can work well for imbalanced training data. We expect that our method can learn appropriate box embeddings for infrequent senses.

3 Proposed Method

This section describes the details of ProtoBox, our proposed method to train contextualized box embeddings. We first explain box embeddings, as background, in subsection 3.1, then explain ProtoBox in the succeeding subsections.

3.1 Box Embeddings

Single vectors represent items as points, while box embeddings represent items as boxes. Box embeddings can naturally represent asymmetric relations like hypernym–hyponym relations by the overlap of two boxes. In this work, a box embedding $\mathbf{b}$ is constructed from two vectors $\mathbf{c}$, the center of the box, and $\mathbf{o}$, an offset from $\mathbf{c}$. $\mathbf{c}$ is the center of a box and $\mathbf{o}$ is the offset from $\mathbf{c}$. Note that the dimensions of $\mathbf{c}$ and $\mathbf{o}$ are equal. The area of the ith dimension of the box embedding is defined as the range $[c_i - o_i, c_i + o_i]$.

Given two boxes $\mathbf{b}_i$ and $\mathbf{b}_j$, the probability that $\mathbf{b}_i$ encloses $\mathbf{b}_j$ can be defined as

$$P(\mathbf{b}_j|\mathbf{b}_i) = \frac{\text{Vol}(\mathbf{b}_i \cap \mathbf{b}_j)}{\text{Vol}(\mathbf{b}_i)}, \tag{1}$$

where $\mathbf{b}_i \cap \mathbf{b}_j$ is the are of the overlap of $\mathbf{b}_i$ and $\mathbf{b}_j$, and Vol($\mathbf{b}$) is the function that calculates the volume of $\mathbf{b}$.

The hard definition of the probability in Equation (1) often leads a serious problem for training box embeddings. When two boxes have no overlap, $P(\mathbf{b}_j|\mathbf{b}_i)$ is zero, causing the training to halt due to the vanishing of the gradient. Therefore, in general, a soft definition is often used. Following previous work (Onoe et al., 2021; Jiang et al., 2023), we use Gumbel Box (Dasgupta et al., 2020), one of the box embeddings that calculates the above probability with a soft definition. Specifically, the probability that $\mathbf{b}_i$ encloses $\mathbf{b}_j$ is calculated with the Gumbel distribution.

3.2 MetricWSD

Since ProtoBox is an extension of MetricWSD (Chen et al., 2021), we first briefly introduce the latter. The left side of Figure 2 shows an overview of MetricWSD. It is a model for WSD, based on Prototypical Networks. The training data is a collection of sentences including a target word (e.g., 'dog') labeled with its gold sense (e.g., dog.1). It is divided into two sets: a support set and a query set. Each sentence in the support set is converted to a contextualized word embedding (or sense embedding) by a model f_θ. In MetricWSD, BERT (Devlin et al., 2019) is used as f_θ. The prototype vector of each sense (e.g., dog.1, dog.2) is defined as the average of the contextualized word embeddings of that sense. Next, the sentences in the query set are converted to contextualized word embeddings by the same model f_θ, and then the loss is calculated by the distances between the query vector and the prototype vectors. Finally, the parameters of the model, θ, are updated so that the loss becomes minimized. At the inference, a test sentence is converted to an embedding by f_θ, and then the similarities between it and the prototype sense vectors are calculated, and the most similar sense is chosen.

3.3 Learning Contextualized Box Embeddings

The right side in Figure 2 shows an overview of ProtoBox. In our method, MetricWSD is modified in three ways. First, instead of a single vector, a sentence is converted to a box embedding by the model. Following previous work (Onoe et al., 2021), we add a Fully Connected Layer (FCL) after BERT. For a sentence x where the zth word is the target word, its contextualized box embedding is obtained as follows:

$$\mathbf{b} = f_\theta(x, z) = \text{FCL}(\text{BERT}(x)[z]). \tag{2}$$

The input of FCL is BERT(x)[z], the contextualized word embedding of the zth word when x is entered to BERT. The output of FCL forms the box embedding $\mathbf{b}$, which is equally divided into two vectors $\mathbf{c}$ and $\mathbf{o}$ by $\mathbf{b} = [\mathbf{c}, \mathbf{o}]$.

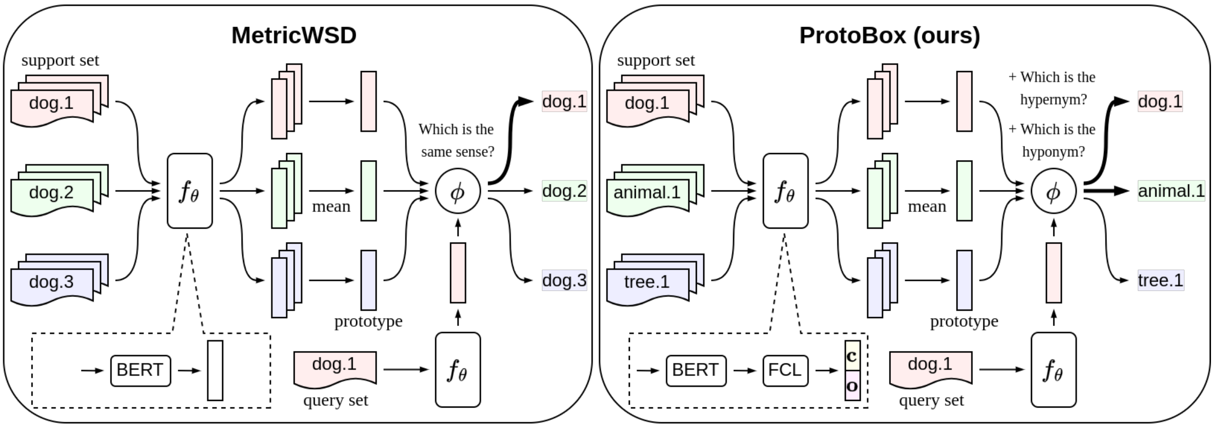

Figure 2: Overview of MetricWSD and ProtoBox (ours).

Second, the episodes are constructed differently. In Prototypical Networks, a mini-batch used to train a model is called an "episode." On the one hand, an episode is a set of instances with different senses of the same target word in MetricWSD. On the other hand, an episode is a set of instances with different senses of multiple target words (e.g. dog.1, animal.1, and tree.1) in ProtoBox. The details of the construction of an episode will be explained in subsection 3.4.

Third, the loss is calculated between a query representation (contextualized box embedding) $\mathbf{b}^q$ and each prototype representation (sense box embedding) $\mathbf{b}^p$. Given a query set $\mathcal{E}_Q = \{x_1, x_2, ..., x_{N_Q}\}$, box embedding of each sample x_i is computed by

$$\mathbf{b}_i^q = f_\theta(x_i, z). \tag{3}$$

Let us suppose $\mathcal{C} = \{s_1, s_2, ..., s_{N_C}\}$ is the set of the senses (of different words) in the entire support set and $\mathcal{E}_{P_j} = \{x_{j1}, x_{j2}, ..., x_{jN_P}\}$ is the support set of the jth sense s_j. The prototype representation of s_j, $\mathbf{b}_j^p$, is defined as the mean of the box embeddings of the samples in the support set:

$$\mathbf{b}_j^p = \frac{1}{N_P} \sum_{i=1}^{N_P} f_\theta(x_{ji}, z). \tag{4}$$

In the above two equations, z stands for the position of the target word in the sentence.

Following Onoe et al. (2021), we use the binary cross-entropy loss between the prototype sense s_j in the support set and the sample x_i in the query set:

$$\begin{aligned} l(\mathbf{b}_j^p, \mathbf{b}_i^q) = &- \delta{\cdot}\log P(\mathbf{b}_j^p|\mathbf{b}_i^q) \\ &- (1-\delta){\cdot}\log\,(1 - P(\mathbf{b}_j^p|\mathbf{b}_i^q)). \end{aligned} \tag{5}$$

Here, δ is 1 if the prototype sense s_j is equal to the sense of x_i or s_j is a hypernym of x_i, otherwise 0. Finally, the total loss $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \frac{1}{N_Q N_C} \sum_{i=1}^{N_Q} \sum_{j=1}^{N_C} \frac{1}{2} (l(\mathbf{b}_j^p, \mathbf{b}_i^q) + l(\mathbf{b}_i^q, \mathbf{b}_j^p)).$$

Intuitively, the model f_θ is trained so that contextualized box embeddings of the same sense overlap each other and a contextualized box embedding of a hypernym encloses that of a hyponym.

3.4 Episode Construction

The training data of ProtoBox is a collection of "sense instances." A sense instance is an example sentence including a certain sense of a target word. To train the model, the training data is divided into episodes. Note that each episode is a pair of support and query sets, $(\mathcal{E}_S, \mathcal{E}_Q)$. The following sets are made: (1) $\mathcal{W}$, a set of small number of randomly chosen target words, (2) $\mathcal{P}_\mathcal{S}$, a set of senses of the target words in $\mathcal{W}$, and (3) $\mathcal{P}_\mathcal{H}$, a set of direct hypernym senses of the senses in $\mathcal{P}_\mathcal{S}$. Then, several senses in $\mathcal{P}_\mathcal{S}$ and $\mathcal{P}_\mathcal{H}$ are chosen as the prototype senses, thus the support set is formed by sense instances of those prototype senses. The query set is made up of the sense instances of the senses in $\mathcal{P}_\mathcal{S}$ that are mutually exclusive with the support set. We limit the number of the target words in each episode to N_W, the maximum number of the prototype senses to N_C, the maximum number of sentences for each prototype sense to N_P (the maximum number of sentences in the entire support set is $N_C \times N_P$), and the maximum number of the sentences in the query set to N_Q.

Algorithm 1 shows how the episodes are constructed. First, N_W words are randomly chosen

Algorithm 1 Construction of Episodes

1: $\mathcal{D}^{\text{train}}$: the training data
2: $\mathcal{V}$: all words in the training data
3: $\mathcal{E} \leftarrow \emptyset$
4: **while** $\mathcal{V} \neq \emptyset$ **do**
5: $\quad \mathcal{W} \leftarrow \textsc{Random}(\mathcal{V}, N_W)$
6: $\quad \mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{W}$
7: $\quad \mathcal{P}_{\mathcal{S}} \leftarrow \bigcup_{w \in \mathcal{W}} \textsc{SenseOfWord}(w)$
8: $\quad \mathcal{P}_{\mathcal{H}} \leftarrow \bigcup_{s \in \mathcal{S}} \textsc{DirectHypernyms}(s)$
9: $\quad$ **if** $|\mathcal{P}_{\mathcal{S}}| + |\mathcal{P}_{\mathcal{H}}| > N_C$ **then**
10: $\quad\quad$ /* ensure $|\mathcal{P}_{\mathcal{S}}| + |\mathcal{P}_{\mathcal{H}}| = N_C$ */
11: $\quad\quad \mathcal{P}_{\mathcal{H}} \leftarrow \textsc{Random}(\mathcal{P}_{\mathcal{H}}, N_C - |\mathcal{P}_{\mathcal{S}}|)$
12: $\quad \mathcal{E}_S \leftarrow \emptyset; \mathcal{E}_Q \leftarrow \emptyset$
13: $\quad$ /* $\mathcal{D}_s^{\text{train}}$: sense instances of s in $\mathcal{D}^{\text{train}}$ */
14: $\quad$ **for** $s \in \mathcal{P}_{\mathcal{S}}$ **do**
15: $\quad\quad \tilde{\mathcal{E}}_S \leftarrow \textsc{Random}(\mathcal{D}_s^{\text{train}}, N_P)$
16: $\quad\quad \mathcal{E}_S \leftarrow \mathcal{E}_S \cup \tilde{\mathcal{E}}_S$
17: $\quad\quad \mathcal{E}_Q \leftarrow \mathcal{E}_Q \cup (\mathcal{D}_s^{\text{train}} \setminus \tilde{\mathcal{E}}_S)$
18: $\quad$ **for** $s \in \mathcal{P}_{\mathcal{H}}$ **do**
19: $\quad\quad \mathcal{E}_S \leftarrow \mathcal{E}_S \cup \textsc{Random}(\mathcal{D}_s^{\text{train}}, N_P)$
20: $\quad \mathcal{E}_Q \leftarrow \textsc{Random}(\mathcal{E}_Q, N_Q)$
21: $\quad \mathcal{E} \leftarrow \mathcal{E} \cup \{(\mathcal{E}_S, \mathcal{E}_Q)\}$
22: **return** $\mathcal{E}$

(line 5). Second, all senses of the randomly chosen target words are kept as $\mathcal{P}_{\mathcal{S}}$ (line 7), and all direct hypernym of those senses are kept as $\mathcal{P}_{\mathcal{H}}$ (line 8). The senses in $\mathcal{P}_{\mathcal{S}}$ and $\mathcal{P}_{\mathcal{H}}$ are used as the prototype senses. More precisely, all the senses in $\mathcal{P}_{\mathcal{S}}$ are kept as prototype senses, while the rest are randomly chosen from $\mathcal{P}_{\mathcal{H}}$ so that the total number of prototype senses becomes N_C (lines 9–11). Then, the randomly chosen N_P instances for each prototype sense are kept as the support set $\mathcal{E}_S$ (lines 15–16, 19), while the N_Q instances of the senses in $\mathcal{P}_{\mathcal{S}}$, which were not selected in the support set, are chosen as the query set $\mathcal{E}_Q$ (lines 17, 20). Note that the function $\textsc{Random}(S, n)$ randomly chooses n samples at most from the set S; all samples are chosen when $|S| < n$. The above procedure is repeated until all words in the training data have been used to make episodes.

Since instances of hypernym senses as well as all the senses of a target word are included in the support set, ProtoBox can consider not only the sense discrimination, as does MetricWSD, but also the hypernym–hyponym relations in the training of the model that produces contextualized box embeddings.

4 Applications of ProtoBox

This section describes how ProtoBox is applied to three tasks: Word Sense Disambiguation, New Sense Classification, and Hypernym Identification.

4.1 Word Sense Disambiguation

Task Definition The goal of Word Sense Disambiguation (WSD) is to select the most appropriate sense of the target word w in a given context x from a predefined inventory $\mathcal{S}_w$ of senses.

Method First, we get the contextualized box embedding $\mathbf{b}^q$ of w in x. Second, we create the sense embedding $\mathbf{b}_i^p$ for each sense s_i in $\mathcal{S}_w$ from the training data. Finally, we calculate the similarity score between $\mathbf{b}_i^p$ and $\mathbf{b}^q$ using Equation (6), which measures by how much two box embeddings overlap, then the most similar sense is chosen to be the predicted sense.

$$\text{sim}(\mathbf{b}_i^p, \mathbf{b}^q) = 2 \times \frac{P(\mathbf{b}_i^p|\mathbf{b}^q)P(\mathbf{b}^q|\mathbf{b}_i^p)}{P(\mathbf{b}_i^p|\mathbf{b}^q) + P(\mathbf{b}^q|\mathbf{b}_i^p)} \quad (6)$$

4.2 New Sense Classification

Task Definition The goal of New Sense Classification (NSC) is to classify the target word w in a given context x, whether it has a new sense or not. In this study, new senses are defined as senses that do not appear in the training data.

Method First, we get $\mathbf{b}^q$ and $\mathbf{b}_i^p$ in the same way that WSD does. For all senses s_i in $\mathcal{S}_w$, if $\text{sim}(\mathbf{b}_i^p, \mathbf{b}^q)$ is smaller than a threshold α_{s_i}, w in x is predicted to be a new sense, otherwise not.

Then α_{s_i} is determined for each sense using the training and development data. Let $\mathcal{D}_{s_i}^{\text{dev}}$ be a set of sense instances of s_i in the development data. The threshold is set to be

$$\alpha_{s_i} = \frac{1}{|\mathcal{D}_{s_i}^{\text{dev}}|} \sum_{j=1}^{|\mathcal{D}_{s_i}^{\text{dev}}|} \text{sim}(\mathbf{b}_i^p, \mathbf{b}_j^q), \quad (7)$$

where $\mathbf{b}_i^p$ is the box embedding of the prototype sense s_i and $\mathbf{b}_j^q$ is the box embedding of the jth instance in $\mathcal{D}_{s_i}^{\text{dev}}$. That is, α_{s_i} is determined as the average similarity between the sense instance of s_i in the development data and the prototype sense s_i in the training data. When there is no sense instance of s_i in the development data, the threshold is set to the average of α_{s_i} for all senses.

4.3 Hypernym Identification

Task Definition Hypernym Identification (HI) is the task of predicting a hypernym of a new sense. Specifically, for a given new sense of a target word w in a context x, we choose and rank the top ten senses that are most likely to be a hypernym of it.

Method First, we get the contextualized box embedding $\mathbf{b}^q$ of w in x and the box embeddings of the prototype senses $\mathbf{b}_i^p$ as in the WSD task. Then, the set $\mathcal{H}$ of candidates of hypernym senses is created:

$$\mathcal{H} = \{s_i \mid P(\mathbf{b}_i^p|\mathbf{b}^q) > \beta\}, \quad (8)$$

where β is a pre-defined threshold. Next, we choose the sense where the difference of the volume of $\mathbf{b}_i^p$ and $\mathbf{b}^q$ is the smallest as the best hypernym sense u.

$$u = \arg\min_{s_i \in \mathcal{H}} |\mathrm{Vol}(\mathbf{b}_i^p) - \mathrm{Vol}(\mathbf{b}^q)| \quad (9)$$

The motivation to consider the difference of the volumes is that when the volume of the box embedding is large, the sense may be an abstract concept and not likely to be a direct hypernym of an input new sense. Finally, all other senses are ranked by their similarity with u (using Equation (6)) and the top nine senses are chosen to make the final ranked list of the hypernyms.

5 Experiments

5.1 Dataset

Following the WSD framework proposed by Raganato et al. (2017), we use SemCor 3.0 (Miller et al., 1994) as the training data, SemEval-2007 (Pradhan et al., 2007) as the development data, and Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-2013 (Navigli et al., 2013), SemEval-2015 (Moro and Navigli, 2015) as the test data. All datasets are corpora annotated with sense labels defined by WordNet (Miller, 1995).

In this work, the only the senses of nouns in the datasets are used. In WordNet, senses of nouns connected by hypernym–hyponym relations form a Directed Acyclic Graph of which the root is the synset "entity.n.01". We create three datasets: $\mathcal{D}_{\text{living\_thing}}$, $\mathcal{D}_{\text{artifact}}$, and $\mathcal{D}_{\text{entity}}$. These datasets consist of instances of hyponyms of "living_thing.n.01", "artifact.n.01", and "entity.n.01" in WordNet, respectively. Here, $\mathcal{D}_{\text{entity}}$ is a large dataset that includes all nouns, while $\mathcal{D}_{\text{living\_thing}}$ and $\mathcal{D}_{\text{artifact}}$ are smaller ones including a restricted number of nouns.

Training data The statistics of the training data $\mathcal{D}^{\text{train}}$ are presented in Table 1. The sizes of $\mathcal{D}^{\text{train}}_{\text{living\_thing}}$ and $\mathcal{D}^{\text{train}}_{\text{artifact}}$ are almost the same, while $\mathcal{D}^{\text{train}}_{\text{entity}}$ is much larger than they are.

	$\mathcal{D}^{\text{train}}_{\text{living\_thing}}$	$\mathcal{D}^{\text{train}}_{\text{artifact}}$	$\mathcal{D}^{\text{train}}_{\text{entity}}$
#senses	1,713	1,939	12,760
#words	1,809	1,994	11,029
#instances	15,838	8,708	84,962

Table 1: The statistics of the training data.

Development and test data The development and the test data for the WSD task are constructed from instances including a target word that has multiple senses and its gold sense appears in the training data. Statistics are shown in Table 2. It is found that a considerable number of test instances have infrequent senses.

	$\mathcal{D}_{\text{living\_thing}}$		$\mathcal{D}_{\text{artifact}}$		$\mathcal{D}_{\text{entity}}$	
	ALL	≤ 10	ALL	≤ 10	ALL	≤ 10
dev	13	5	9	5	125	54
test	190	66	78	40	2,514	992

Table 2: The number of instances in development and test data for WSD task. "ALL" means all instances. "≤ 10" means instances of a sense that appears in the training data less than or equal to 10 times.

The development and the test data for the NSC task are constructed from instances including a target word that appear in the training data. The instances are labeled as "new sense" if its gold sense does not appear in the training data, otherwise as "not new sense". The statistics are shown in Table 3.

	$\mathcal{D}_{\text{living\_thing}}$		$\mathcal{D}_{\text{artifact}}$		$\mathcal{D}_{\text{entity}}$	
	new	not	new	not	new	not
dev	0	23	0	18	7	144
test	20	500	12	221	295	3,379

Table 3: The number of instances in development and test data for NSC task. "new" and "not" mean new sense and not new sense, respectively.

The development and the test data for the HI task are constructed from instances whose gold senses do not appear in the training data. The statistics are shown in Table 4. The gold hypernym is determined by WordNet.

	$\mathcal{D}_{\text{living\_thing}}$	$\mathcal{D}_{\text{artifact}}$	$\mathcal{D}_{\text{entity}}$
dev	2	2	11
test	107	32	658

Table 4: The number of instances in development and test data for HI task.

5.2 Settings

Baselines We prepare two baselines: vanilla BERT (BERT-NN) and MetricWSD (Chen et al., 2021). These models output a contextualized embedding (single vector) $\mathbf{r}$ for a given sense instance. In BERT-NN, the embedding of a prototype sense are obtained by the average of the vectors of sense instances derived from the pre-trained BERT. The similarity between two vectors $\mathbf{r}_i$ and $\mathbf{r}_j$ is defined as the dot product $\text{sim}(\mathbf{r}_i, \mathbf{r}_j) = \mathbf{r}_i \cdot \mathbf{r}_j$.

The baselines perform WSD and NSC in the same way as our method, except that the similarity between two instances is measured by two single vectors. In HI, the baseline chooses the ten most similar senses to make up a ranked list of hypernym senses.

Parameters For all models in BERT-NN, MetricWSD, and ProtoBox, we use bert-base-uncased as the BERT model. We set the number of dimensions of the output layer of FCL to 256 (i.e. the size of $\mathbf{c}$ and $\mathbf{o}$ is 128), N_W is 32, N_C is 128, N_P is 5, and N_Q is 64. As for the hyperparameters for the fine-tuning of BERT, the learning rate is set to 1e-5. The number of epochs is optimized, that is, it is varied from 1 to 200 and the best value is chosen using the development data.

5.3 Results and Analysis

Word Sense Disambiguation Table 5 shows the accuracy on the WSD task. As can be seen from the column "ALL", our ProtoBox outperformed the two baselines for $\mathcal{D}_{\text{living\_thing}}$, but was comparable for $\mathcal{D}_{\text{artifact}}$ and $\mathcal{D}_{\text{entity}}$. We guess that the poor performance on $\mathcal{D}_{\text{entity}}$ was caused by the scale, that is, our method failed to obtain appropriate contextualized box embeddings when it was applied to many sense instances. The reason why ProtoBox was worse than MetricWSD on $\mathcal{D}_{\text{artifact}}$ may not be a scale issue, but the semantic domain of the target noun, since the sizes of $\mathcal{D}_{\text{living\_thing}}$ and $\mathcal{D}_{\text{artifact}}$ were almost the same.

A similar tendency for the disambiguation of infrequent senses can be seen in the column "≤ 10". Surprisingly, BERT-NN achieved the best accuracy on $\mathcal{D}_{\text{entity}}$, although the pretrained BERT model was just applied without fine-tuning. MetricWSD and ProtoBox still suffered from the data sparseness when they were applied to the large dataset.

New Sense Classification The results on the New Sense Classification task are shown in Table 6. Comparing the F1-score, ProtoBox was comparable to MetricWSD on $\mathcal{D}_{\text{living\_thing}}$ and $\mathcal{D}_{\text{entity}}$, and significantly worse on $\mathcal{D}_{\text{artifact}}$. The poor performance for NSC for the senses of artifacts was coincident with the results of the WSD task, where ProtoBox was worse than MetricWSD on $\mathcal{D}_{\text{artifact}}$. The F1-score of BERT-NN on $\mathcal{D}_{\text{living\_thing}}$ was notable as it was better than that of MetricWSD and ProtoBox. ProtoBox is designed to learn hypernym–hyponym relations between senses, but such knowledge may not be indispensable for New Sense Classification. This might be the reason why ProtoBox could not outperform MetricWSD.

Hypernym Identification Following previous work on Taxonomy Expansion (Shen et al., 2020; Yu et al., 2020; Jiang et al., 2023), we evaluate the baselines and ProtoBox in term of three metrics: accuracy (ACC), Mean Reciprocal Rank (MRR), Wu-Palmer similarity (W&P) (Wu and Palmer, 1994). Accuracy measures the agreement ratio between the gold hypernym and the highest ranked hypernym, while Wu–Palmer similarity measures how closely these two hypernyms are located in WordNet. The parameter β described in subsection 4.3 is set to 0.5, 0.7, or 0.9.[2]

The results on the HI task are shown in Table 7. ProtoBox outperformed the baselines in all three evaluation metrics on the three datasets. In particular, the difference between ProtoBox and the baselines was significant on $\mathcal{D}_{\text{living\_thing}}$. The many gold hypernyms in the test data of $\mathcal{D}_{\text{living\_thing}}$ were "person.n.01", which were correctly predicted by ProtoBox. On the other hand, on $\mathcal{D}_{\text{artifact}}$ and $\mathcal{D}_{\text{entity}}$, the differences in terms of ACC and MRR between ProtoBox and the baselines were small. However, a significant difference of W&P was confirmed, indicating that ProtoBox could predict hypernyms closer to the correct ones. Finally, the performance of ProtoBox was sensitive to the parameter β, especially in terms of ACC and MRR. Investigating how to optimize β would be the important future work.

[2] β was not optimized due to the insufficiency of the development data.

Model	$\mathcal{D}_{\text{living\_thing}}$		$\mathcal{D}_{\text{artifact}}$		$\mathcal{D}_{\text{entity}}$	
	ALL	≤ 10	ALL	≤ 10	ALL	≤ 10
BERT-NN	.816	.727	.744	.775	.579	**.602**
MetricWSD	.821	.773	**.872**	**.925**	**.711**	.588
ProtoBox (ours)	**.884**	**.788**	.859	.875	.707	.584

Table 5: Accuracy of WSD task. "ALL" indicates the results for all senses, and "$\leq$ 10" for infrequent senses.

Model	$\mathcal{D}_{\text{living\_thing}}$				$\mathcal{D}_{\text{artifact}}$				$\mathcal{D}_{\text{entity}}$			
	A	P	R	F	A	P	R	F	A	P	R	F
BERT-NN	**.744**	**.099**	**.700**	**.174**	**.682**	.069	.417	.119	**.656**	.119	.515	.194
MetricWSD	.712	.055	.400	.096	.674	**.119**	**.833**	**.208**	.633	**.138**	**.678**	**.229**
ProtoBox (ours)	.704	.053	.400	.094	.618	.086	.667	.152	.628	.132	.651	.219

Table 6: Results of New Sense Classification task. A, P, R, and F mean accuracy, precision, recall, and F1 score, respectively.

Model	β	$\mathcal{D}_{\text{living\_thing}}$			$\mathcal{D}_{\text{artifact}}$			$\mathcal{D}_{\text{entity}}$		
		ACC	MRR	W&P	ACC	MRR	W&P	ACC	MRR	W&P
BERT-NN	–	.150	.259	.754	.094	.150	.567	.068	.113	.460
MetricWSD	–	.103	.219	.767	.062	.170	.505	.073	.124	.494
ProtoBox (ours)	0.5	.439	.502	.855	.125	**.179**	.628	.061	.084	.539
	0.7	.533	.570	.876	**.156**	.175	**.644**	.081	.113	.558
	0.9	**.579**	**.604**	**.877**	.062	.076	.621	**.100**	**.126**	**.565**

Table 7: Results of Hypernym Identification task.

5.4 Optimization of Number of Dimensions

We analyzed how the performance of WSD was influenced by the number of the dimensions of the box embeddings **c** and **o**. In this experiment, the number of dimensions of the box embeddings was set to $\{32, 64, 128, 192, 256\}$. Table 8 shows the accuracy of WSD on the development data of $\mathcal{D}_{\text{entity}}$. It was found that the best performance for both "ALL" and "$\leq$ 10" was obtained when the number of dimensions was set to 128. Therefore, as described in subsection 5.2, the number of dimensions was set to 256 (128 + 128). For NSC and HI tasks, we did not optimize this since the development data was small, but set it to be the same number as for the WSD task.

Dimension	ALL	≤ 10
32	.744	.593
64	.784	.630
128	**.792**	**.685**
192	.752	.630
256	.736	.574

Table 8: Accuracy of WSD task on the development data for different number of dimensions of **c** and **o**.

6 Conclusion

This paper proposed ProtoBox, an expansion of MetricWSD to learn contextualized box embeddings. The representations of words in a context were changed from single vectors in MetricWSD to box embeddings in our ProtoBox, since box embeddings are suitable to represent semantic relations between senses such as the hypernym–hyponym relation. Additionally, we proposed a method to construct episodes to train the model to produce the contextualized box embeddings. We evaluated ProtoBox on three tasks: Word Sense Disambiguation (WSD), New Sense Classification (NSC), and Hypernym Identification (HI). ProtoBox outperformed the baselines in terms of all evaluation metrics in the HI task. This was reasonable, since ProtoBox was designed to take the hypernym–hyponym relation into account when training the contextualized box embeddings. In addition, ProtoBox achieved a performance comparable with the baselines for the other sense related tasks, WSD and NSC.

In the future, the scalability of ProtoBox should be improved. As reported in subsection 5.3, the performance of ProtoBox was degraded when the number of sense instances was large. A more efficient and precise method to learn contextualized box embeddings should be investigated. In addition, the definition of the prototype representation (the box embedding of a sense) can be reconsidered. Currently, the prototype representation is an average of box embeddings of the elements in the support set. However, it can be a box that includes all the elements in the support set. It is worth to explore better ways to obtain the prototype representation.

Limitations

In the experiments, ProtoBox was only applied to nouns. Additional experiments are required to investigate how ProtoBox can work well for other parts of speech, such as verbs.

The parameter β in the HI task was not optimized due to the insufficiency of the development data. It is worth investigating how to find an appropriate threshold in the future.

We did not compare ProtoBox with other methods of contextualized box embeddings such as Onoe et al. (2021) and Jiang et al. (2023) in the experiments, since the target tasks were not completely the same as the three tasks in this paper. However, empirical comparison is necessary to clarify the contribution of our method.

Ethics Statement

Since ProtoBox was developed using the established datasets for WSD that have been widely used in the community and contain no private information, there is no concern for data and privacy.

References

Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. 2019. Every child should have parents: A taxonomy refinement algorithm based on hyperbolic term embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4811–4817, Florence, Italy. Association for Computational Linguistics.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego, California. Association for Computational Linguistics.

Howard Chen, Mengzhou Xia, and Danqi Chen. 2021. Non-parametric few-shot learning for word sense disambiguation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1774–1781, Online. Association for Computational Linguistics.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Shib Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Li, and Andrew McCallum. 2020. Improving local identifiability in probabilistic box embeddings. *Advances in Neural Information Processing Systems*, 33.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. 2023. A single vector is not enough: Taxonomy expansion via box embeddings. In *Proceedings of the ACM Web Conference 2023*, pages 2467–2476.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Mingyu Derek Ma, Muhao Chen, Te-Lin Wu, and Nanyun Peng. 2021. HyperExpan: Taxonomy expansion with hyperbolic representation learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4182–4194, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4724–4737, Dublin, Ireland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, 30.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. Association for Computational Linguistics.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. TaxoExpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*, pages 486–497.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Lei Yu and Yang Xu. 2023. Word sense extension. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3281–3294, Toronto, Canada. Association for Computational Linguistics.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. STEAM: Self-supervised taxonomy expansion with mini-paths. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1026–1035.

A Visualization of Box Embeddings

To verify whether ProtoBox could learn appropriate relations between senses, we visualize box embeddings of several prototype senses. Figure 3 represents box embeddings of animal.n.01 and dog.n.01 trained by ProtoBox from $\mathcal{D}^{\text{train}}_{\text{living\_thing}}$. The horizontal axis represents the dimensions of the boxes, while the vertical axis represents the intervals of each dimension $[c_i - o_i, c_i + o_i]$. It is found that the box of animal.n.01 almost encloses that of dog.n.01, indicating that animal.n.01 is a hypernym of dog.n.01. This is also supported by the fact that P(animal.n.01 | dog.n.01) = 0.999. Therefore, the model learned the hypernym–hyponym relation between animal.n.01 and dog.n.01. Next, let us consider cat.n.01 and dog.n.01, for which there is no hypernym–hyponym relation in WordNet. Looking at Figure 4 and the two probabilities P(cat.n.01 | dog.n.01) = 0.146 and P(dog.n.01 | cat.n.01) = 0.089, the two boxes seem to not overlap very much. Even though cat.n.01 and dog.n.01 are conceptually similar, ProtoBox can learn that there is no hypernym–hyponym relation between them.

Figure 5 shows box embeddings of building.n.01 and house.n.01 trained by ProtoBox from $\mathcal{D}^{\text{train}}_{\text{artifact}}$, and Figure 6 shows box embeddings of hotel.n.01 and house.n.01. The box embeddings of those senses are also adequate. That is, the box of building.n.01 almost encloses that of house.n.01, and the boxes of hotel.n.01 and house.n.01 do not overlap very much.

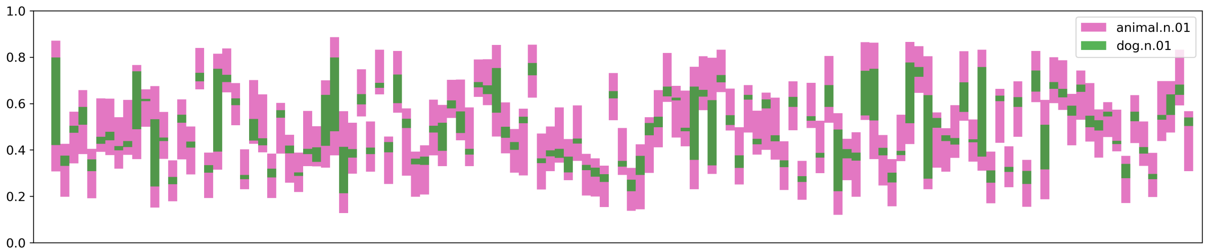

Figure 3: Box embeddings of animal.n.01 and dog.n.01 trained from $\mathcal{D}^{\text{train}}_{\text{living\_thing}}$. $P(\text{animal.n.01} \mid \text{dog.n.01}) = 0.999$, $P(\text{dog.n.01} \mid \text{animal.n.01}) = 3.12\text{e-}9$.

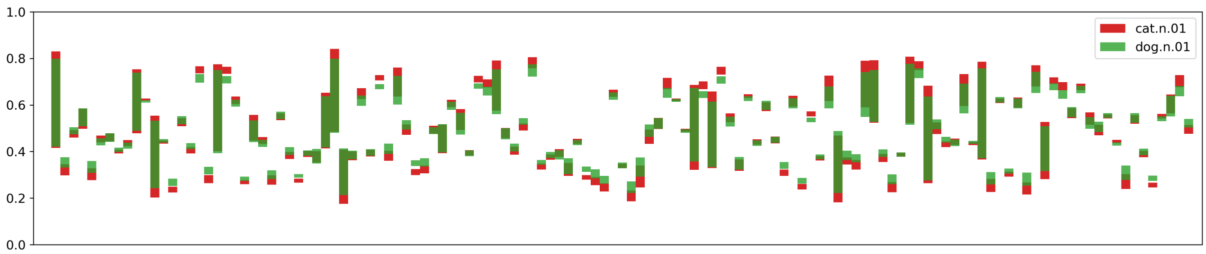

Figure 4: Box embeddings of cat.n.01 and dog.n.01 trained from $\mathcal{D}^{\text{train}}_{\text{living\_thing}}$. $P(\text{cat.n.01} \mid \text{dog.n.01}) = 0.146$, $P(\text{dog.n.01} \mid \text{cat.n.01}) = 0.089$.

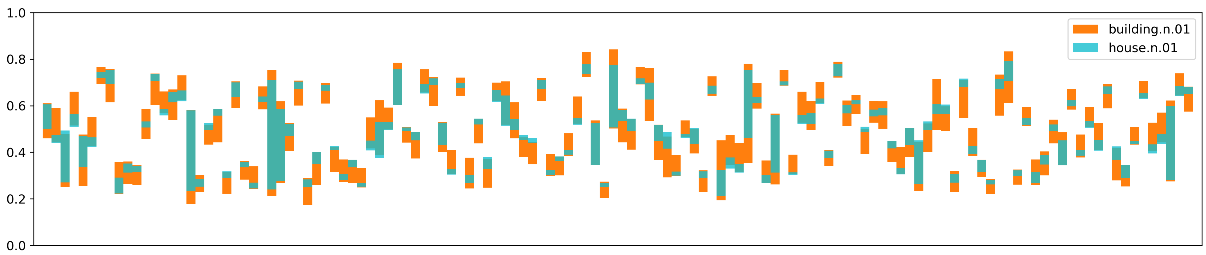

Figure 5: Box embeddings of building.n.01 and house.n.01 trained from $\mathcal{D}^{\text{train}}_{\text{artifact}}$. $P(\text{building.n.01} \mid \text{house.n.01}) = 0.766$, $P(\text{house.n.01} \mid \text{building.n.01}) = 2.97\text{e-}4$.

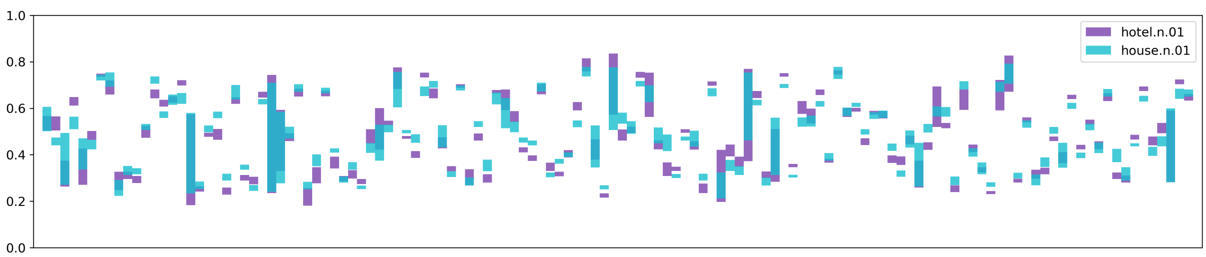

Figure 6: Box embeddings of hotel.n.01 and house.n.01 trained from $\mathcal{D}^{\text{train}}_{\text{artifact}}$. $P(\text{hotel.n.01} \mid \text{house.n.01}) = 0.011$, $P(\text{house.n.01} \mid \text{hotel.n.01}) = 0.013$.