Hyponym Detection Using Strict Partial Order Networks

Sarthak Dash, Md Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, Nicolas Rodolfo Faucellaglia
IBM Research AI, Yorktown Heights, NY, USA
{sdash, mchowdh, gliozzo}@us.ibm.com
{nandana.m, nicolas.faucellaglia}@ibm.com

Abstract
This paper introduces Strict Partial Order Networks (SPON), a novel neural network architecture designed to enforce asymmetry and transitive properties as soft constraints. We apply it to induce hypernymy relations by training with is-a pairs. We also present an augmented variant of SPON that can generalize type information learned for in-vocabulary terms to previously unseen ones. An extensive evaluation over eleven benchmarks across different tasks shows that SPON consistently either outperforms or attains the state of the art on all but one of these benchmarks.

1 Introduction
The ability to generalize the meaning of domain-specific terms is essential for many NLP applications. However, building taxonomies by hand for a new domain is time-consuming. This drives the requirement to develop automatic systems that are able to identify hypernymy relationships (i.e. is-a relations) from text.

Hypernymy relation is reflexive and transitive but not symmetric (George et al. 1990; Hearst 1992). For example, if Wittgenstein \( \prec \) philosopher and philosopher \( \prec \) person, where \( \prec \) means is-a, it follows that Wittgenstein \( \prec \) person (transitivity). In addition, it also follows that both philosopher \( \not\prec \) Wittgenstein and person \( \not\prec \) philosopher (asymmetry). Absence of self-loops within taxonomies (e.g. WordNet (George et al. 1990)) emphasizes that reflexivity (e.g. person \( \prec \) person) does not add any new information.

In order theory, a partial order is a binary relation that is transitive, reflexive and anti-symmetric. A strict partial order is a binary relation that is transitive, irreflexive and asymmetric. Strict partial orders correspond more directly to directed acyclic graphs (DAGs). In fact, hypernymy relation hierarchy in WordNet is a DAG (Suchanek, Kasneci, and Weikum 2008). Therefore, we hypothesize that the Hypernymy relations within a taxonomy can be better represented via strict partial order relations.

In this paper we introduce Strict Partial Order Networks (SPON), a neural network architecture comprising of non-negative activations and residual connections designed to enforce strict partial order as a soft constraint. We present an implementation of SPON designed to learn is-a relations. The input of SPON is a list of is-a pairs, provided either by applying Hearst-like patterns over a text corpus or via a list of manually validated pairs.

In order to identify hypernyms for out-of-vocabulary (OOV) terms, i.e. terms that are not seen by SPON during the training phase, we present an augmented variant of SPON that can generalize type information learned for the in-vocabulary terms to previously unseen ones. The augmented model does so by using normalized distributional similarity values as weights within a probabilistic model, the details of which are described in Section 5.

The main contributions of this paper are the following:
- We introduce the idea of Strict Partial Order Network (SPON), highlighting differences and similarities with previous approaches aimed at the same task.
- A theoretical analysis shows SPON enforces asymmetry and transitivity requirement as soft constraints.
- An augmented variant of SPON to predict hypernyms for OOV is proposed.
- Compared to previous approaches, we demonstrate that our system achieves and/or improves the state of the art (SOTA) consistently across a large variety of hypernymy tasks and datasets (multi-lingual and domain-specific), including supervised and unsupervised settings.

The rest of the paper is structured as follows. Section 2 describes related work. SPON is introduced in Section 3, and theoretical analysis is provided in Section 4. In Section 5 we show how SPON can be augmented for OOV terms in the test dataset. Section 6 and 7 describe the evaluation setup and results. Section 8 concludes the paper and highlights perspectives for future work.

2 Related Work
Since the pioneering work of Hearst (1992), lexico-syntactic pattern-based approaches (e.g., “NP_y is a NP_x”) remains influential in subsequent academic and commercial applications. Some work tried to learn such patterns automatically (Snow, Jurafsky, and Ng 2005; Schwartz, Goldberg, and Dagan 2016) instead of using a predefined list of patterns.

Fu et al. (2014) argued that hypernym-hyponym pairs preserve linguistic regularities such as $v$(shrimp) – $v$(prawn) ≈ $v$(fish) – $v$(goldfish), where $v$(w) is the embedding of the word w. In other words, they claimed that a hyponym word can be projected to its hypernym word learning a transition matrix $\Phi$. Tan, Gupta, and van Genabith (2015) proposed a deep neural network based approach to learn is-a vectors that can replace $\Phi$.

Recently, Roller, Kiela, and Nickel (2018) showed that exploitation of matrix factorization (MF) on a Hearst-like pattern-based system’s output vastly improved their results (for different hypernymy tasks; in multiple datasets) with comparison to that of both distributional and non-MF pattern-based approaches.

Another thread of related work involves the use of graph embedding techniques for representing a hierarchical structure. Order-embeddings (Vendrov et al. 2016) encode text and images with embeddings, preserving a partial order (i.e. $x \preceq y$, where x is a specific concept and y is a more general concept) over individual embedding dimensions using the Reversed Product Order on $\mathbb{R}^N$. In contrast, our proposed neural network based model encodes a strict partial order through a composition of non-linearities and residual connections. This allows our model to be as expressive as possible, all the while maintaining strict partial order.

Li, Vilnis, and McCallum (2017) extended the work of Vendrov et al. (2016) by augmenting distributional co-occurrences with order embeddings. In addition, hyperbolic embeddings model tree structures using non-euclidean geometries, and can be viewed as a continuous generalization of the same (Nickel and Kiela 2017). Other recent works have induced hierarchies using box-lattice structures (Vilnis et al. 2018) and Gaussian Word Embeddings (Athiwaratkun and Wilson 2018).

Regarding the recent SOTA, for unsupervised setting where manually annotated (i.e. gold standard) training data is not provided, Le et al. (2019) proposed a new method combining hyperbolic embeddings and Hearst-like patterns, and obtained significantly better results on several benchmark datasets.

For supervised setting, during the SemEval-2018 hypernymy shared task (Camacho-Collados et al. 2018), the CRIM system (Bennet-Colborne and Barriere 2018) obtained best results on English datasets (General English, Medical and Music). This system combines supervised projection learning with a Hearst-like pattern-based system’s output. In the same shared task, for Italian, the best system, 300-sparsans, was a logistic regression model based on sparse coding and a formal concept hierarchy obtained from word embeddings (Berend, Makrai, and Földiák 2018); whereas for Spanish, the best system, NLP_H2, was based on the nearest neighbors algorithm (Qiu et al. 2018).

In Sections 6 and 7 we compare our approach with all of the above mentioned recent SOTA in both unsupervised and supervised settings, respectively.

3 Strict Partial Order Networks

The goal of SPON is to estimate the probability for a distinct pair of elements $x, y \in \mathcal{E}$ to be related by a strict partial order $x \prec y$. A specific instance of this problem is the hypernym detection problem, where $\mathcal{E}$ is a vocabulary of terms and $\prec$ is the is-a relation. In this section, we present a SPON implementation, while a theoretical analysis of how the proposed architecture satisfies transitive and asymmetric properties is described in the next section.

An implementation of a SPON is illustrated in Figure 1. Each term $x \in \mathcal{E}$ is represented via a vector $\vec{x} \in \mathbb{R}^d$. In the first step, we perform an element-wise multiplication with a weight vector $w_1$ and then add to a bias vector $b_1$. The next step consists of a standard ReLU layer, that applies the transformation $ReLU(v) = max(0, v)$. Let us denote these transformations by a smooth function $g$,

$$g(\vec{x}) = ReLU(w_1 \otimes \vec{x} + b_1)$$

where $\otimes$ denote element-wise multiplication.

The final step, as depicted in Figure 1, consists of a residual connection, i.e.

$$f(\vec{x}) = \vec{x} + g(\vec{x})$$

Figure 1: Simple SPON architecture.

We encode the loss layer to capture the distance-to-satisfaction $\psi$ for a given candidate hyponym-hypernym pair $(x, y)$, defined as follows:

$$\psi(x, y) = \sum_{i=1}^{d} max(0, \epsilon + f(\vec{x})_i - \vec{y}_i)$$

where the sum is taken over all the components of the participating dimensions, and $\epsilon$ is a scalar hyper-parameter.
The network is trained by feeding positive and negative samples generated that satisfy expressions given by equations 1 and 2, satisfy a zero loss for a given hyponym-hypernym pair \(x, y\). Formally, for a given candidate hyponym term \(x\), let
\[
\mathcal{H}_x = \{e \in \mathcal{E} \mid (x, e, s) \in \mathcal{T}\}
\]
denote all the candidate hypernym terms of \(x\), and let
\[
\mathcal{H}'_x = \{e \in \mathcal{E} \mid (x, e, s) \notin \mathcal{T}\}
\]
denote negative hypernym samples for \(x\). Negative hypernym terms are sampled at random from \(\mathcal{E}\), and as many negative samples are generated that satisfy \(|\mathcal{H}_x| + |\mathcal{H}'_x| = k\), a constant (hyper-parameter for the model).

The probability of \((x, y)\) being a true hyponym-hypernym pair is then calculated using an approach analogous to Boltzmann distribution as follows,
\[
p(x, y) = \frac{e^{-\psi(x, y)}}{\sum_{z \in \mathcal{H}_x \cup \mathcal{H}'_x} e^{-\psi(x, z)}}
\]
Equation 6 is used for training, while during scoring, the probability that a pair \((x, y)\) exhibits hypernymy relationship is given by,
\[
p(x, y) = \frac{e^{-\psi(x, y)}}{\sum_{z \in \mathcal{H}} e^{-\psi(x, z)}}
\]

whereas, the most likely hypernym term \(y^*\) for a given hyponym term \(x\) is given by,
\[
y^* = \arg \max_{y \in \mathcal{H}} \frac{e^{-\psi(x, y)}}{\sum_{z \in \mathcal{H}} e^{-\psi(x, z)}}
\]
Here, \(\mathcal{H}\) denotes the list of all hypernym terms observed in the training set \(\mathcal{T}\).

Finally, we define the loss function \(J\) using a weighted negative log-likelihood criterion (w-NLL) defined as follows,
\[
J = - \sum_{(x, y, s) \in \mathcal{T}} s \log p(x, y)
\]
where \(s\) represents the relative importance of the loss associated with pair \((x, y)\) in \(\mathcal{T}\).

### 4 Theoretical Analysis

**Hypernymy** relations within a taxonomy satisfy two properties: asymmetry and transitivity. The asymmetry property states that given two distinct terms \(x, y \in \mathcal{E}\), if \(x \prec y\) then \(y \not\prec x\). The transitive property states that given three distinct terms \(x, y, z \in \mathcal{E}\), if \(x \prec y\) and \(y \prec z\) then \(x \prec z\).

In this section we analytically demonstrate that the neural network architecture depicted in Fig. 1, whose forward pass expressions are given by equations 1 and 2, satisfy asymmetry and transitive properties.

As described by equation 3, our proposed model assigns a zero loss for a given hyponym-hypernym pair \((x, y)\) if the learned model satisfies \(f(x) < y\) element-wise. This formulation of the loss layer puts forth the following constraint that defines our model,

\[
x \prec y \iff f(x)_i < y_i, \forall i
\]

In other words, the relation \(\prec\) is satisfied if and only if \(f(x) < y\), component-wise. In the rest of this section, we show that under the assumption of \([10]\), our proposed model for hypernymy relation satisfies asymmetry and transitivity.

**Theorem 4.1.** Expression 10 satisfies asymmetry.

**Proof.** Let \(x \prec y\). Then, it follows expression 10 that \(f(x) < y\) component-wise. We need to show that \(y \not\prec x\). Using the definition of equation 10, it is enough to show \(f(y) \geq x\) component-wise.

Now, using expression 2, we have \(f(y) = y + g(f(y))\). From the definition of function \(g\), it is clear that \(g(x) \geq 0\) component-wise. Thus, applying this inequality to the previous expression, we have \(f(y) \geq y\) component-wise. On similar lines, we can also show that

\[
f(x) \geq x \forall x \in \mathcal{E}
\]

component wise.

We now have \(f(y) \geq y > f(x) \geq x\) component wise. The middle inequality holds, since we assume \(x \prec y\); in other words, \(f(x) < y\) holds component wise. Thus expression 10 satisfies asymmetry.

**Theorem 4.2.** Expression 10 satisfies transitivity.

**Proof.** Let \(x \prec y\) and \(y \prec z\). Then, it follows from expression 10 that \(f(x) < y\) and \(f(y) < z\), component wise. We need to show that \(x \prec z\) or, alternatively, that \(f(x) < z\).

Generalizing equation 11, we have, \(\forall e \in \mathcal{E}, f(e) \geq \bar{e}\) component wise. Using this observation, we have \(f(x) < y \leq f(y) < z\) component wise. Note that the middle inequality holds from the aforementioned observation. This proves that Expression 10 satisfies transitivity.

### 5 Generalizing SPON to OOV

The proposed SPON model is able to learn embedding for terms appearing in the training data (extracted either using Hearst-like patterns or provided via a manually labelled training set). However, for tasks wherein one needs to induce hypernymy relationships automatically from a text corpus, Hearst-like patterns usually are not exhaustive.

Yet, there is often a practical requirement in most applications to assign OOV to their most likely correct type(s). Designing a system that fulfills this requirement is highly significant since it allows the creation of hypernymy relationships from a given text corpus, avoiding the problem of sparsity that often characterizes most knowledge bases. The basic idea is to use an augmented SPON approach that leverages distributional similarity metrics between words in the same corpus. This is formally described as follows.

For a given domain, let \(\mathcal{T}_{\text{train}}\) and \(\mathcal{O}_{\text{train}}\) denote the in-vocabulary and OOV input trial hyponym terms; and let \(\mathcal{T}_{\text{test}}\) and \(\mathcal{O}_{\text{test}}\) denote the in-vocabulary and OOV input
test hyponym terms respectively. Let $E^{train}$ denote all the terms observed in the list of training hyponym-hypernym pairs, and let $H^{train}$ denote the list of known hyponyms obtained from the list of training pairs. The hyponym terms from $T^{trial}$ and $T^{test}$ are handled by our proposed SPON model, i.e. top-ranked hyponyms for each hyponym term are generated via our model.

The rest of this section deals with how to generate top ranked hyponyms for each hyponym term within $O^{trial}$ and $O^{test}$ respectively. Let $Y_x$ be the random variable denoting the hyponym assignment for an OOV term $x \in O^{test}$ (Similar approach holds for OOV terms from $O^{trial}$). The probability of the random variable $Y_x$ taking on the value $c \in H^{train}$ is then given by,

$$P(Y_x = c | x) = \sum_{h \in E^{train}} P(Y_x = c, h | x)$$

$$= \sum_{h \in E^{train}} P(Y_x = c | h, x).P(h | x) \quad (12)$$

The first equality in the above expression is a direct consequence of Marginalisation property in probability, whereas the second equality merely represents the marginal probability in terms of conditional probability. We now make a conditional independence assumption, i.e. $Y_x \perp\!\perp x | h$, or in other words, ignoring the subscript for brevity we have, $P(Y_x = c | h, x) = P(Y = c | h)$. Using this assumption, we can rewrite Equation 12 as,

$$P(Y_x = c | x) = \sum_{h \in E^{train}} P(Y = c | h).P(h | x)$$

$$\approx \sum_{h \in S^p_{\text{f}}} P(Y = c | h).f(h | x) \quad (13)$$

where $f$ is a scoring function that provides a score between $[0, 1]$, and $S^p_{\text{f}}$ contains $p$-terms from $E^{train}$ that provide top-k largest values for the scoring function $f$. In practice, we first normalize the values of $f(h | x)$ where $h \in S^p_{\text{f}}$ using a softmax operation, before computing the weighted sum, as per Equation 13. Also, note that $p$ is a hyper-parameter in this model.

Looking back at Equation 13, we notice that the first part of the summation, i.e. $P(Y = c | h)$, can be obtained directly from our proposed SPON model, since $h \in E^{train}$. In addition, we model the function $f(h | x)$ as cosine-similarity between the vectors for the term $h$ and $x$, wherein the vectors are trained via a standard Word2Vec model pre-built on the corresponding tokenized corpus for the given benchmark dataset.

Summarizing, given a query OOV term within the trial or test fold of any dataset, our proposed model follows the aforementioned strategy to generate a list of hyponym terms that have been ranked using the formula in Equation 13.

It should be clearly pointed out that our aforementioned proposed OOV strategy is not a stand-alone strategy, rather its performance is inherently dependent on SPON.

### 6 Unsupervised Benchmarks and Evaluation

SPON is intrinsically supervised because it requires example is-a pairs for training. However, it can also be applied to unsupervised hyponymy task, provided that example is-a pairs are generated by an external unsupervised process such as Hearst-like patterns.

#### Benchmarks

In the unsupervised setting, no gold training data is provided and the system is supposed to assess the validity of test data, provided as a set of pairs of words. A small validation dataset is also provided which is used for tuning hyper-parameters.

We evaluated our approach on two tasks. The first one is hyponym detection where the goal is to classify whether a given pair of terms are in a hyponymy relation. The second task is direction prediction, i.e. to identify which term in a given pair is the hyponym. We use the same datasets, same settings, same evaluation script and same evaluation metrics as Roller, Kiela, and Nickel (2018). Table 1 shows the dataset statistics for unsupervised benchmarks, wherein the split into validation/test folds is already given.

For detection, Average Precision is reported on 5 datasets, namely BLESS (Baroni and Lenci 2011), LEDS (Baroni et al. 2012), EVAL (Santus et al. 2015), WBLESS (Weeds et al. 2014) and SHWARTZ (Shwartz, Goldberg, and Dagan 2016). While for direction, Average Accuracy is reported on 3 datasets, which are BIBLESS (Kiel et al. 2015), BLESS and WBLESS. We refer the readers to Roller, Kiela, and Nickel (2018) for details about these datasets.

We adopted the approach of Roller, Kiela, and Nickel (2018) where a list $L$ of hyponym-hypernym pairs $(x, y)$ is extracted using a Hearst-like pattern-based system. This system consists of 20 Hearst-like patterns applied to a concatenation of Wikipedia and Gigaword corpora, to generate $L$.

Each pair within $L$ is associated with a count $c$ (how often $(x, y)$ has been extracted). Positive Mutual Information

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Valid</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLESS</td>
<td>1,453</td>
<td>13,089</td>
</tr>
<tr>
<td>EVAL</td>
<td>736</td>
<td>12,714</td>
</tr>
<tr>
<td>LEDS</td>
<td>275</td>
<td>2,495</td>
</tr>
<tr>
<td>SHWARTZ</td>
<td>5,236</td>
<td>47,321</td>
</tr>
<tr>
<td>WBLESS</td>
<td>167</td>
<td>1,501</td>
</tr>
</tbody>
</table>

Table 1: Statistics for benchmark datasets used in unsupervised hyponym detection and direction prediction tasks. The columns represent the number of hyponym-hypernym pairs within the validation and test folds respectively.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Italian/Spanish</th>
<th>Music/Medical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>1500</td>
<td>1000</td>
<td>500</td>
</tr>
<tr>
<td>Trial</td>
<td>50</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>Test</td>
<td>1500</td>
<td>1000</td>
<td>500</td>
</tr>
</tbody>
</table>

Table 2: Number of hyponyms in different datasets within SemEval 2018 hyponym discovery task.
Table 3: Results on the unsupervised hypernym detection and direction prediction tasks. The first three rows of results are from Roller, Kiela, and Nickel (2018). The HyperbolicCones results were reported by Le et al. (2019). The improvements in LEDS and BLESS benchmark are statistically significant with two-tailed p values being 0.019 and \( \leq 0.001 \) respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLESS</th>
<th>EVAL</th>
<th>LEDS</th>
<th>SHWARTZ</th>
<th>WBLESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count based ( p(x,y) )</td>
<td>.49</td>
<td>.38</td>
<td>.71</td>
<td>.29</td>
<td>.74</td>
</tr>
<tr>
<td>ppmi(( x, y ))</td>
<td>.45</td>
<td>.36</td>
<td>.70</td>
<td>.28</td>
<td>.72</td>
</tr>
<tr>
<td>SVD ppmi(( x, y ))</td>
<td>.76</td>
<td>.48</td>
<td>.84</td>
<td>.44</td>
<td>.96</td>
</tr>
<tr>
<td>HyperbolicCones</td>
<td>.81</td>
<td>.50</td>
<td>.89</td>
<td>.50</td>
<td>.98</td>
</tr>
<tr>
<td>Proposed SPON</td>
<td>.81</td>
<td>.50</td>
<td>.91</td>
<td>.50</td>
<td>.98</td>
</tr>
</tbody>
</table>

Table 4: Ablation tests reporting Average Precision values on the unsupervised hypernym detection task, signifying the choice of layers utilized in our proposed SPON model. The first row represents SPON i.e. a \( \text{RELU} \) layer followed by a Residual connection. The second row removes the Residual connection, whereas the third row substitutes the non-negative activation layer \( \text{RELU} \) with \( \text{Tanh} \) that can take negative values.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLESS</th>
<th>EVAL</th>
<th>LEDS</th>
<th>WBLESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{RELU+Residual} )</td>
<td>.81</td>
<td>.50</td>
<td>.91</td>
<td>.98</td>
</tr>
<tr>
<td>( \text{RELU Only} )</td>
<td>.73</td>
<td>.49</td>
<td>.82</td>
<td>.96</td>
</tr>
<tr>
<td>( \text{Tanh+Residual} )</td>
<td>.79</td>
<td>.49</td>
<td>.90</td>
<td>.98</td>
</tr>
</tbody>
</table>

Table 5: Results on the unsupervised hypernym detection task for BLESS dataset. With 13,089 test instances, the improvement in Average Precision values obtained by SPON as compared against Smoothed Box model is statistically significant with two-tailed \( p \) value equals 0.00116.

In addition, we used \( L_1 \) regularization for model weights, and also used dropout with probability of 0.5. Adam optimizer was used with default settings. In addition, the term vectors in our model were initialized uniformly at random, and are constrained to have unit L2 norm during the entire training procedure. Furthermore, an early stopping criterion of 20 epochs was used.

**Evaluation**

We use the same evaluation script as provided by Roller, Kiela, and Nickel (2018) for evaluating our proposed model. Table 3 shows the results on the unsupervised tasks of hypernym detection and direction predictions, reporting average precision and average accuracy, respectively.

The first row titled Count based (in Table 3) depicts the performance of a Hearst-like Pattern system baseline, that uses a frequency based threshold to classify candidate hyponym-hypernym pairs as positive (i.e. exhibiting hypernymy) or negative (i.e. not exhibiting hypernymy). The ppmi approach in Table 3 builds upon the Count based approach.
Table 6: Results on SemEval 2018 General-purpose hypernym discovery task. CRIM, NLP_HZ, and 300-sparsans are the corresponding best systems on English, Spanish and Italian datasets (see Section 2).

Table 7: Results on SemEval 2018 Domain-specific hypernym discovery task. CRIM is the best system on the domain specific datasets.

Table 4 shows the results for each of these ablation experiments, when evaluated on the unsupervised hypernym detection task across four datasets chosen randomly. Removing the Residual layer and using RELU activation function only, violates the aforementioned component-wise inequality $f(\vec{x}) \geq \vec{x}$, and has the worst results out of the three. On the other hand, using Residual connections with Tanh activations may not violate the aforementioned inequality, since, it depends upon the sign of the activation outputs. This argument is supported by the results in Table 4, wherein using Tanh activations instead of RELU almost provides identical results, except for the BLESS dataset. Nevertheless, the results in Table 4 show that encouraging asymmetry and transitive properties for this task, in fact improves the results as opposed to not doing the same.

Furthermore, Table 5 illustrates the results on the unsupervised hypernym detection task for BLESS dataset, wherein we compare our proposed SPON model to other supervised SOTA approaches for hypernym prediction task, namely Order Embeddings (OE) approach as introduced by (Vendrov et al. 2016), and Smoothed Box model as introduced by (Li et al. 2019). We run the OE and Smoothed Box experiments using the codes provided with those papers.

In addition, we used the validation fold within BLESS dataset to empirically determine optimal hyper-parameter configurations, and settled on the following values: For OE, we used an embedding dimensions of 20, margin parameter of 5, generated one negative example for every positive instance using so-called contrastive approach. For Smoothed Box model, we used an embedding dimensions of 50 and generated five negatives per training instance. In either case, we observed that using the entire set of is-a pairs extracted by the Hearst-like patterns (without employing a frequency based cutoff) for training provided the best performance.

From Table 5, it is clear that SPON performs much better (by at least 1.6%) as compared to Smoothed Box model as well as Order Embedding model in an Unsupervised benchmark dataset.

### 7 Supervised Benchmarks and Evaluation

In the supervised setting, a system has access to a large corpus of text from where training, trial, and test is-a pairs are extracted, and labeled manually.

#### Benchmarks

We used the benchmark of SemEval 2018 Task on Hypernym Discovery. The task is defined as "given an input term,
The HP system uses a fixed set of Hearst-like patterns (e.g. “y such as x”, “y including than x1, x2”, etc) to extract pairs from the input corpus. Then, it filters out any of these pairs where either x (hyponym) or y (hypernym) is not seen in the corresponding vocabulary provided by the organizers of the shared task. It also discards any pair where y is not seen in the corresponding gold training data.

Following that, the HP system makes a directed graph by considering each pair as an edge and the corresponding terms inside pair as nodes. The weight of each edge is the count/frequency of how often (x, y) has been extracted by the Hearst-like patterns.

It also excludes any cycle inside the graph; e.g. if (x, y), (y, z) and (z, x) then all these edges (i.e. pairs) were discarded. Finally, it discards any edge that has a value lower than a frequency threshold, \(ft\). We set \(ft=10\) for English. For Medical and Music, we set \(ft=2\).

The is-a pairs obtained from HP system are then merged with the corresponding training gold pairs (i.e. treated them equally) to form a larger training set for English, Medical and Music datasets. As a result of this step, the number of total unique training pairs for English, Medical and Music increased to 17,903 (from 11,779 training gold pairs), 4,593 (from 3,256) and 6,282 (from 5,455) correspondingly.

The dataset statistics for the general-purpose and domain-specific hypernym discovery tasks are mentioned in Table 2. It is evident that a significant fraction of the terms in the Trial/Test fold is OOV (see Figure 2), therefore SPON is not able to make any assessment about them. Therefore, in order to handle OOV cases, we represent all terms in the dictionary provided by the SemEval organizers via Word2Vec vectors acquired from the given text corpus.

The \(d\) dimensions for SPON model was chosen from \(\{50, 100, 200\}\), whereas the parameter \(p\) (for handling OOV terms) was chosen from \(\{2,3,5,8,10\}\). Parameter \(k\) (from Equation 5) was chosen from \(\{100,200,500\}\). The regularization, dropout and initialization strategies are exactly similar to Section 7. An early stopping criterion of 50 epochs was used.

**Evaluation**

We use the scorer script provided as part of SemEval-2018 Task 9 for evaluating our proposed model. Table 6 shows the results on the three general purpose domains of English, Spanish, and Italian respectively. For brevity, we compare only with the SOTA, i.e., the best system in each task. Performances of all the systems that participated in SemEval 2018 Task on Hypernym Discovery can be found in (Camacho-Collados et al. 2018). Similarly, Table 7 shows the results on the two domain-specific tasks of music and medical domain corpora. SPON outperforms the SOTA systems in all tasks except for the medical domain in which it achieves comparable results. It is worthwhile to notice that SPON is fully domain-agnostic, i.e., it neither uses any domain-specific approaches, nor any domain-specific external knowledge. We provide an illustration of the output of our system in Table 8, showing a sample of randomly selected terms and their corresponding ranked predictions.

**8 Conclusion and Future Work**

In this paper, we introduced SPON, a novel neural network architecture that models hypernymy as a strict partial order relation. We presented a materialization of SPON, along with an augmented variant that assigns types to OOV terms. An extensive evaluation over several widely known academic benchmarks demonstrates that SPON largely improves (or attains) SOTA values across different tasks.

There are so many benchmark datasets for hypernymy prediction task with different evaluation settings (supervised and unsupervised). None of the recent approaches choose to report results on all of them which makes it difficult to decide whether any one of them performs consistently well over others. Our paper fills this void.

In the future, we plan to explore how to extend SPON in two directions. On the one hand, we plan to analyze how to use SPON for the taxonomy construction task (i.e., constructing a hierarchy of hypernyms instead of flat is-a pairs). On the other hand, we plan to generalize our work to relations other than is-a.

Table 8: Examples of ranked predictions (from left-to-right) made by our system on a set of eight randomly selected test queries from SemEval 2018 English dataset. The top four query terms are OOV, while the bottom ones are in-vocabulary. Hypernyms predicted by SPON that matches the gold annotations are highlighted in bold, while we use *italicize* for predictions that we judge to be correct but are missing in the gold standard expected hypernyms.

<table>
<thead>
<tr>
<th>Term</th>
<th>Predicted hypernyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>dicoumarol</td>
<td>drug, carbohydrate, acid, person, ...</td>
</tr>
<tr>
<td>Planck</td>
<td>person, particle, physics, elementary particle, ...</td>
</tr>
<tr>
<td>Belt Line</td>
<td>main road, infrastructure, expressway, ...</td>
</tr>
<tr>
<td>relief</td>
<td>service, assistance, resource, ...</td>
</tr>
<tr>
<td>honesty</td>
<td>virtue, ideal, moral philosophy, ...</td>
</tr>
<tr>
<td>shoe</td>
<td>footwear, shoe, footwear, overshoe, ...</td>
</tr>
<tr>
<td>ethanol</td>
<td>alcohol, fuel, person, fluid, resource, ...</td>
</tr>
<tr>
<td>ruby</td>
<td>language, precious stone, person, ...</td>
</tr>
</tbody>
</table>

*retrieve its hypernyms from a target corpus*. For each input hyponym in the test data, a ranked list of candidate hypernyms is expected. The benchmark consists of five different subtasks covering both general-purpose (multiple languages – English, Italian, and Spanish) and domain-specific (Music and Medicine domains) hypernymy discovery.

**Experimental Settings**

This subsection describes the technical solution we implemented for the SemEval tasks, more specifically, the strategies for training dataset augmentation, handling OOV terms, and the hyperparameter optimization.

For the corpora in English, we augmented the training data using automatically extracted pairs by an unsupervised Hearst-like Pattern (HP) based system, following an approach similar to that described by Bernier-Colborne and Barriere (2018), the best system in English, Medical and Music hypernymy subtasks in SemEval 2018. Henceforth, we refer to our pattern-based approach as the HP system.

The HP system uses a fixed set of Hearst-like patterns (e.g. “y such as x”, “y including than x1, x2”, etc) to extract pairs from the input corpus. Then, it filters out any of these pairs where either x (hyponym) or y (hypernym) is not seen in the corresponding vocabulary provided by the organizers of the shared task. It also discards any pair where y is not seen in the corresponding gold training data.

Following that, the HP system makes a directed graph by considering each pair as an edge and the corresponding terms inside pair as nodes. The weight of each edge is the count/frequency of how often (x, y) has been extracted by the Hearst-like patterns.

It also excludes any cycle inside the graph; e.g. if (x, y), (y, z) and (z, x) then all these edges (i.e. pairs) were discarded. Finally, it discards any edge that has a value lower than a frequency threshold, \(ft\). We set \(ft=10\) for English. For Medical and Music, we set \(ft=2\).

The is-a pairs obtained from HP system are then merged with the corresponding training gold pairs (i.e. treated them equally) to form a larger training set for English, Medical and Music datasets. As a result of this step, the number of total unique training pairs for English, Medical and Music increased to 17,903 (from 11,779 training gold pairs), 4,593 (from 3,256) and 6,282 (from 5,455) correspondingly.
References

Athiwaratkun, B., and Wilson, A. G. 2018. Hierarchical density order embeddings. In ICLR.


