# Distributional Inclusion Hypothesis and Quantifications: Probing for Hypernymy in Functional Distributional Semantics

**Chun Hei Lo**[1] **Wai Lam**[1] **Hong Cheng**[1] **Guy Emerson**[2]
[1]The Chinese University of Hong Kong [2]University of Cambridge
{chlo, wlam, hcheng}@se.cuhk.edu.hk gete2@cam.ac.uk

## Abstract

Functional Distributional Semantics (FDS) models the meaning of words by truth-conditional functions. This provides a natural representation for hypernymy but no guarantee that it can be learnt when FDS models are trained on a corpus. In this paper, we probe into FDS models and study the representations learnt, drawing connections between quantifications, the Distributional Inclusion Hypothesis (DIH), and the variational-autoencoding objective of FDS model training. Using synthetic data sets, we reveal that FDS models learn hypernymy on a restricted class of corpus that strictly follows the DIH. We further introduce a training objective that both enables hypernymy learning under the reverse of the DIH and improves hypernymy detection from real corpora.

## 1 Introduction

Functional Distributional Semantics (FDS; Emerson and Copestake, 2016; Emerson, 2018) suggests that the meaning of a word can be modelled as a truth-conditional function, whose parameters can be learnt using the distributional information in a corpus (Emerson, 2020a; Lo et al., 2023). Aligning with truth-conditional semantics, functional representations of words are logically more rigorous than vectors (e.g., Mikolov et al., 2013; Pennington et al., 2014; Levy and Goldberg, 2014; Czarnowska et al., 2019) and distributions (e.g., Vilnis and McCallum, 2015, Bražinskas et al., 2018) as concepts are separated from their referents (for a discussion, see: Emerson, 2020b, 2023). On top of its theoretical favour, Lo et al. (2023) also demonstrate FDS models in action and show that they are very competitive in the semantic tasks of semantic composition and verb disambiguation.

Hypernymy is also known as lexical entailment. It is formally defined as the subsumption of extensions between two word senses, which can be modelled with truth-conditional functions. Although FDS provides the basis for embedding hypernymy, it is not obvious whether hypernymy can be learnt by training an FDS model on a corpus, and if so, what kind of corpus it can successfully learn from.

To acquire hypernymy automatically from a corpus, one way is through the use of distributional information. In this class of methods, hypernymy is learnt in an unsupervised manner given certain hypotheses about the distributional properties of the corpus. One such hypothesis is the Distributional Inclusion Hypothesis (DIH; Weeds et al., 2004; Geffet and Dagan, 2005), which relates lexical entailment of words to the subsumption of the typical contexts they appear with in a corpus.

In §2, we first highlight that while existential quantifications support the DIH, universal quantifications reverse it. In §3, we discuss how FDS can embed hypernymy, formulate our hypothesis that FDS learns hypernymy under the DIH, and introduce a training objective that handles simple universal quantifications. In §4, using synthetic data sets, we show that FDS learns hypernymy under the DIH, and under the reverse of DIH when the new objective is used. In §5, we show that the new objective encodes word generality and improves hypernymy detection by FDS on real corpora.

## 2 Distributional Inclusion Hypothesis and Quantifications

The Distributional Inclusion Hypothesis (DIH) asserts that the typical characteristic features (contexts) of $r_h$ are expected to appear with $r_H$ if and only if $r_H$ is a hypernym of $r_h$. Although Geffet and Dagan (2005) report that the DIH is largely valid on a real corpus, it is not deemed fully correct in general as a hyponym can appear in exclusive contexts due to collocational (Rimell, 2014) and pragmatic reasons (Pannitto et al., 2018), and feature inclusion has been found to be selective (Roller et al., 2014). In this section, we describe how quantifications can also be pivotal to the hypothesis.



*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14625–14637
August 11-16, 2024 

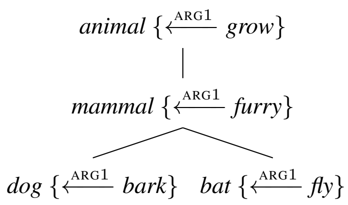



Figure 1: A taxonomic hierarchy of nouns. Next to each noun is the set of contexts that are applicable to the extension of it and those of its descendants (e.g., all dogs are furry, but not all animals.).

| Corpus 1 (DIH) | Corpus 2 (rDIH) |
|---|---|
| *a dog barks* | *every dog barks* |
| *a mammal barks* | *every dog is furry* |
| *an animal barks* | *every dog grows* |
| *a bat flies* | *every bat flies* |
| *a mammal flies* | *every bat is furry* |
| *an animal flies* | *every bat grows* |
| *a mammal is furry* | *every mammal is furry* |
| *an animal is furry* | *every mammal grows* |
| *an animal grows* | *every animal grows* |

Table 1: Corpora generated from the hierarchy in Fig. 1. Existential and universal quantifications result in two corpora that follow the DIH and rDIH respectively.

While Geffet and Dagan (2005) consider syntax-based context, we suggest that contexts based on semantic representation are more suitable since syntactic differences do not necessarily contribute to semantic ones (e.g., passivizations and inversions), and the subject of concern should be semantics. We use Dependency Minimal Recursion Semantics (DMRS; Copestake et al., 2005; Copestake, 2009) as the semantic representation, which is derived using the English Resource Grammar (ERG; Flickinger, 2000, 2011). Fig. 2 shows the predicate–argument structure of an example DMRS graph. If $r_i \xleftarrow{\text{ARG}[a]} r_j$ exists in the DMRS graph of a sentence in the corpus, we can say that $r_i$ appears in the context $\xleftarrow{\text{ARG}[a]} r_j$.

Consider a corpus as a partial description of a world. Distributional properties would depend on how the world is described. Here, we consider a corpus of simple sentences in the form '*[quantifier] [noun] [context word]*'. Take the taxonomic hierarchy in Fig. 1 as an example, where each noun has a set of applicable contexts. If we want to generate existentially quantified statements that are true, then: (1) a noun can appear in its hypernyms' contexts, e.g., '*a dog grows*', where $\xleftarrow{\text{ARG1}}$ *grow* is applicable to *animal*; and (2) a noun can appear in its hyponyms' contexts, e.g., '*an animal barks*', where $\xleftarrow{\text{ARG1}}$ *bark* is applicable to *dog*. If we only generate (2) and restrict (1) so that contexts that are broadly applicable are not used with more specific nouns, this creates a corpus that follows the DIH. Corpus 1 of Table 1 shows an example.[1]

In contrast, generating sentences with universal quantifications results in a corpus that follows the *reverse* of the DIH (rDIH), as in Corpus 2, where the set of contexts of *mammal* is a subset of that of *dog*. Consequently, methods that rely on the DIH as a cue for hypernymy would be undermined.

In §5, we use these processes to generate corpora which strictly align with the DIH or rDIH. Corpora with more complex sentence structures would require a richer world model than can be encoded in a taxonomic hierarchy like Fig. 1. For instance, with a restricted relative clause, '*every dog that is trained is gentle*' does not entail '*every Chihuahua is gentle*' even if *Chihuahua* is a hyponym of *dog*, as the universal quantifier applies only to trained dogs. We also disregard negations because they can co-occur nearly freely, effectively making a context word in the negated scope uninformative. For example, '*a dog does not* ___' is much less selective than '*a dog* ___'.

## 3 Functional Distributional Semantics

In this section, we introduce Functional Distributional Semantics (FDS), discuss hypernymy representation in FDS and explain how FDS can be adapted to handle quantifications. We follow Lo et al. (2023)'s FDS implementation which is briefly described here.

### 3.1 Model-Theoretic Semantics

FDS is motivated by model-theoretic semantics, which sees meaning in terms of an extensional model structure that consists of a set of *entities*, and a set of *predicates*, each of which is true or false of the entities. In parallel, FDS represents an entity by a *pixie* which is taken to be a high-dimensional feature vector, and represents a predicate by a truth-conditional *semantic function* which takes pixie(s) as input and returns the probability of truth.

### 3.2 Probabilistic Graphical Models

The framework is formalized in terms of a family of probabilistic graphical models. Each of them

[1] Without the restriction on (1), exhaustively generating true assertions generates a corpus where the DIH does not hold between nouns in a unary chain (e.g., *animal* and *mammal* in Fig. 1), which would appear in the same set of contexts.

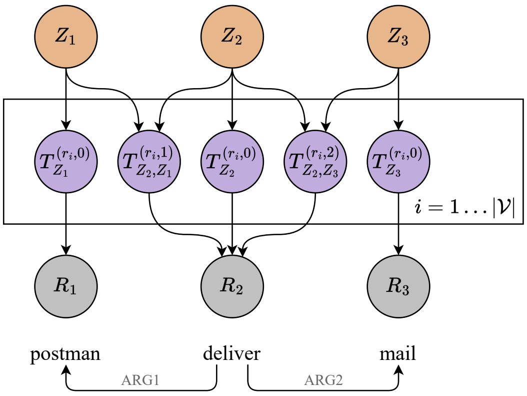



Figure 2: Probabilistic graphical model of FDS for generating words in an SVO triple '*postman deliver mail*'. Only $R_1 = postman$, $R_2 = deliver$, and $R_3 = mail$ are observed.

describes the generative process of predicates in the semantic graph of a sentence. Fig. 2 illustrates the process of generating the words given the argument structure $R_1 \xleftarrow{\text{ARG1}} R_2 \xrightarrow{\text{ARG2}} R_3$. First, a pixie $Z_j \in \mathbb{R}^d$ is generated for each node in the graph, together representing the entities described by the sentence. Then, for each pixie $Z_j$, a truth value $T^{(r_i,0)}_{Z_j}$ is generated for each predicate $r_i$ in the vocabulary $\mathcal{V}$; and for each pair of nodes connected as $R_j \xrightarrow{\text{ARG}a} R_k$ whose corresponding pixies are $Z_j$ and $Z_k$, a truth value $T^{(r_i,a)}_{Z_j,Z_k}$ is generated for each predicate $r_i$ in the vocabulary. Finally, a single predicate $R_j$ is generated for each pixie $Z_j$ conditioned on the truth values.

### 3.3 Semantic Functions

As mentioned in §3.1, instead of treating a predicate as an indicator function, FDS models the probability that it is true of the pixie(s) with unary (in (1)) and binary semantic functions (in (2)). This allows the model to account for vagueness.

$$P\left(T^{(r_i,0)}_{Z_j}{=}\top \,\middle|\, z_j\right) = t^{(r_i,0)}(z_j) \tag{1}$$

$$P\left(T^{(r_i,a)}_{Z_j,Z_k}{=}\top \,\middle|\, z_j, z_k\right) = t^{(r_i,a)}(z_j, z_k) \tag{2}$$

The functions are implemented as linear classifiers as in (3) and (4), where $S$ denotes the sigmoid function, and the trainable parameters of the semantic functions are the weights, $v^{(r_i,0)}$, $v_1^{(r_i,a)}$ and $v_2^{(r_i,a)}$, and the biases, $b^{(r_i,0)}$ and $b^{(r_i,a)}$.

$$t^{(r_i,0)}(z_j) = S\left({v^{(r_i,0)}}^\top z_j + b^{(r_i,0)}\right) \tag{3}$$

$$t^{(r_i,a)}(z_j, z_k) = S\left({v_1^{(r_i,a)}}^\top z_j + {v_2^{(r_i,a)}}^\top z_k + b^{(r_i,a)}\right) \tag{4}$$

### 3.4 Representing Hypernymy

In truth-conditional semantics, for a set of entities $D$, $r_H$ is a hypernym of $r_h$ if and only if

$$\forall x \in D\colon r_h(x) \implies r_H(x). \tag{5}$$

Although FDS provides truth-conditional interpretations of words, it is not straightforward to define hypernymy in FDS where predicates are probabilistic and work over high-dimensional pixies. One way is to translate (5) to a probabilistic counterpart for a score on hypernymy, $P\left(T^{(r_H,0)}_Z{=}\top \,\middle|\, T^{(r_h,0)}_Z{=}\top\right)$. However, FDS only directly models $P\left(T^{(r_H,0)}_Z{=}\top \,\middle|\, z\right)$ and $P\left(T^{(r_h,0)}_Z{=}\top \,\middle|\, z\right)$. The proposed conditional probability is in principle underspecified without further assuming a density $p(z)$ and the conditional independence of $T^{(r_H,0)}_Z$ and $T^{(r_h,0)}_Z$ given $Z$, and to obtain it is also computationally prohibitive as it requires integration over the high dimensional pixie space.

Another way is to interpret the probability model from a fuzzy set perspective and use fuzzy set containment (Zadeh, 1965):

$$\forall z\colon t^{(r_H,0)}(z) > t^{(r_h,0)}(z). \tag{6}$$

Note that if we consider all $z \in \mathbb{R}^d$, (6) can only be true when $v^{(r_h,0)} = v^{(r_H,0)}$ and $b^{(r_H,0)} > b^{(r_h,0)}$. This poses a very strict condition on the semantic function parameters which is in practice impossible to be obtained from model training. For a more viable representation of hypernymy, we restrict the pixie space and only consider the valid space to be a unit hypersphere or hypercube. As a consequence, the original training objective that considers pixies within the whole $\mathbb{R}^d$ space has to be amended accordingly, which will be described in §3.5. With (3) and (4), $r_H$ is considered the hypernym of $r_h$ if and only if $s(r_h, r_H) > 0$, as defined by

$$s(r_h, r_H) = b^{(r_H,0)} - b^{(r_h,0)} - \left\| v^{(r_H,0)} - v^{(r_h,0)} \right\|_p, \tag{7}$$

where $p \in \{1, 2\}$ (derivation in Appendix A). Cheng et al. (2023) also use this score for hypernymy. Note that the transitivity of (5) is paralleled (derivation in Appendix B):

$$\begin{aligned} s(r_1, r_2) > 0 \wedge s(r_2, r_3) > 0 \\ \implies s(r_1, r_3) > 0. \end{aligned} \tag{8}$$

Having a hypernymy representation built into a distributional model allows generalization out of missing information, which can be difficult for hypernymy models based on strict subsumption of contexts. To illustrate, consider the nouns *dog* and *mammal* in Fig. 1 and Corpus 1 in Table 1. If *a dog is furry* does not exist in Corpus 1, the DIH does not hold between *dog* and *mammal*, and it would be challenging for models based strictly on the DIH to recover their hypernymy relations. However, if we know that *fox* has largely overlapping contexts with *dog* in a corpus (e.g., $\{\xleftarrow{\text{ARG1}}$ *bark*, $\xleftarrow{\text{ARG1}}$ *omnivorous*, $\dots\}$), and that *mammal* is known to be a hypernym of *fox* (e.g., from strict context subsumption), we may infer that *mammal* is also a hypernym of *dog*. This generalization can also apply to hyponymy, e.g., *machine* and *system* sharing $\{\xleftarrow{\text{ARG1}}$ *complex*$\}$ and sharing *computer* as their hyponym. In §4.4, we will present experimental results on the distributional generalization behaviour of FDS on hypernymy learning.

### 3.5 Original Training Objective

FDS models are trained using the variational-autoencoding method on simplified DMRS graphs where quantifiers and scopal information are removed from the graphs before training, leaving us with just the predicate–argument structure. The approximate posterior distribution of pixies $q_\phi$ is taken to be $n$ spherical Gaussian distributions, each with mean $\mu_{Z_i}$ and covariance $\sigma^2_{Z_i} I$. Given an observed DMRS graph $G$ with $n$ pixies $Z_1 \dots Z_n$, we maximize (9), reformulated from the $\beta$-VAE (Higgins et al., 2017).

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{n} \mathcal{C}_i + \sum_{r_i \xrightarrow{\text{ARG}[a]} r_j \text{in } G} \mathcal{C}_{i,j,a} \\ - \frac{d}{2} \sum_{i=1}^{n} \beta_1 \mu^2_{Z_i} + \beta_2 \left(\sigma^2_{Z_i} - \ln \sigma^2_{Z_i}\right) \end{aligned} \tag{9}$$

The first two terms, further defined by (10) and (11) respectively, aim to maximize the truthness of observed predicates and the falsehood of $K$ negatively sampled ones $r'_k$ over the inferred pixie distribution $q_\phi$. The last term in (9) is the regularization term for the approximate posterior. Owing to the decision of restricting the valid pixie space described in §3.4, $\beta_1$ has to be set higher in our models than Lo et al. (2023)'s (details described in Appendix D).

$$\begin{aligned} \mathcal{C}_i = \ln \mathbb{E}_{q_\phi} \left[ t^{(r_i,0)}(z_i) \right] \\ + \sum_{k=1}^{K} \ln \mathbb{E}_{q_\phi} \left[ 1 - t^{(r'_k,0)}(z_i) \right] \end{aligned} \tag{10}$$

$$\begin{aligned} \mathcal{C}_{i,j,a} = \ln \mathbb{E}_{q_\phi} \left[ t^{(r_i,a)}(z_i, z_j) \right] \\ + \sum_{k=1}^{K} \ln \mathbb{E}_{q_\phi} \left[ 1 - t^{(r'_k,a)}(z_i, z_j) \right] \end{aligned} \tag{11}$$

Both the local predicate–argument structure of each predicate and global topical information in the graph are used for variational inference. For instance, the approximate posterior distribution of the pixie $Z_1$ of *postman* in Fig. 2 is inferred from the direct argument information, $\xleftarrow{\text{ARG1}}$ *deliver*, and the indirect topical predicate, $\not\leftarrow$ *mail*.

**Our Hypothesis.** We hypothesize that if the training corpus strictly follows the DIH, hypernymy can be learnt by FDS models. The intuition behind our hypothesis is elaborated in Appendix C.

### 3.6 Proposed Objective for Universal Quantifications

FDS assumes that each observed predicate refers to only one point in the pixie space and offers no tools for dealing with regions. We propose a method to allow optimizations of semantic functions with respect to a region in the pixie space, thus enabling FDS to handle simple sentences with universal quantifications. Essentially, we add the following **∀-objective** to the original objective in (9) to give

$$\mathcal{L}_\forall = \sum_{r_j \xleftarrow{\text{ARG}[a]} r_i \text{ in } G} s_a(r_i, r_j) + \mathcal{U}_{i,j,a}, \tag{12}$$

where $r_j$ is a predicate whose referent is universally quantified, and

$$s_a(r_i, r_j) = b^{(r_i,a)} - b^{(r_j,0)} - \left\| v_2^{(r_i,a)} - v^{(r_j,0)} \right\|_p, \tag{13}$$

$$\mathcal{U}_{i,j,a} = \sum_{k=1}^{K} \min\left(0, -s_0(r_i, r'_k)\right) + \sum_{k=1}^{K} \min\left(0, -s_{a''_k}(r''_k, r_j)\right). \tag{14}$$

Note that (13) is modified based on (7), previously defined for classifying hypernymy.

To explain (12), consider the sentence '*every dog barks*' as an example. The first term inside the summation in (12) enforces that extension of $r_j$ is a subset of that of prototypical argument $a$ of $r_i$, i.e., the set of dogs should be contained in the set of agents that barks. The second term, described in (14), incorporates $K$ randomly generated negative samples. $r'_k$ is a noun, which is a negative sample for $r_j$. $r''_k$ is a verb or adjective and $a''_k$ is an argument role, together form a negative sample for $r_i$ and $a$. Then, (14) requires that it is false to universally quantify the referents of the noun $r'_k$ in $r'_k \xleftarrow{\text{ARG}[a]} r_i$ and $r_j$ in $r_j \xleftarrow{\text{ARG}[a''_k]} r''_k$. For the example, both of the following sentences are considered false: '*every dog is owned*' and '*every cat barks*', where $r'_k = cat$, $r''_k = own$ and $a''_k = 2$.

## 4 Experiments on Synthetic Data Sets

Testing our hypothesis in §3.5 and the effectiveness of the new objective for universal quantifications in §3.6 requires corpora that strictly follow the DIH or rDIH, which is impractical for real corpora. Therefore, we create a collection of synthetic data sets and perform experiments on them.

### 4.1 Synthetic Data Sets under the (r)DIH

Each of the synthetic data sets consists of a taxonomic hierarchy of nouns and a corpus, created using the following procedure:

1. **Create a taxonomic hierarchy.** Define a set of nouns, the hypernymy relations of them, and the contexts applicable to its extension and those of its hyponyms (as in Fig. 1).
2. **Choose a hypothesis.** The DIH or rDIH.
3. **Create a corpus.** Create sentences in the form '*[quantifier] [noun] [context word]*' following the chosen hypothesis and the defined hierarchy (as in Table 1).

#### 4.1.1 Topology of Hierarchy

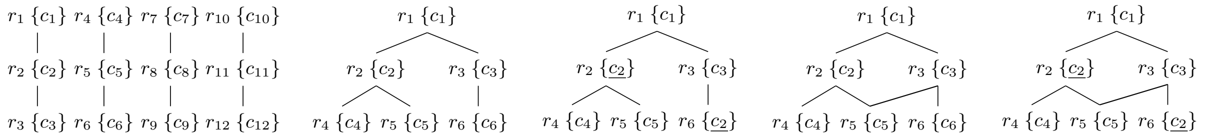
Different topologies of hierarchy lead to different distributional usage of words, thus possibly varying representations learnt for hypernymy. For example, a noun can have multiple hypernyms (e.g., *dog* is the hyponym of both *pet* and *mammal*), or share overlapping contexts with another noun far in the hierarchy (e.g., both *bat* and *airplane* $\xleftarrow{\text{ARG1}}$ *fly*).

To test the robustness of FDS models for learning hypernymy, we experiment with a range of topologies. Fig. 3 exemplifies the five classes of topologies used. We expect that directed acyclic graphs ($H_{\text{DAG}}$ and $H'_{\text{DAG}}$) be harder topologies than trees ($H_{\text{tree}}$ and $H'_{\text{tree}}$), and topologies with overlapping contexts ($H'_{\text{tree}}$ and $H'_{\text{DAG}}$) be harder than those without ($H_{\text{tree}}$ and $H_{\text{DAG}}$). In addition, we test $H_{\text{chains}}$ with pixie dimensionality $d = 2$. A 2-D pixie space allows lossless visualization of the semantic functions. To test hypernymy learning at scale on an actual hierarchy, we make use of WordNet (Miller, 1995; Fellbaum, 1998) and test our models on the WordNet's hierarchy ($H_{\text{WN}}$).

Every node in the hierarchy consists of a noun and a semantic context. The topology of the $H_{\text{chains}}$ used in the experiment is exactly as depicted in Fig. 3. $H_{\text{WN}}$ is created out of the synset *entity.n.01* in WordNet, which is the root, and its hyponymic synsets. This results in 74,374 nodes with 663,492 hypernymy pairs. We randomly sample 663,492 pairs from the remaining pairs as negative instances for evaluation. For the remaining hierarchies, each of them consists of 153 nodes with a height of 5. For $H_{\text{tree}}$, the first level is a root node, and a node at the $h^{\text{th}}$ level has $(h + 1)$ direct children. $H_{\text{tree}'}$ is created from $H_{\text{tree}}$ by choosing 5 pairs of nodes and making each pair share a context set. $H_{\text{DAG}}$ and $H'_{\text{DAG}}$ are created from $H_{\text{tree}}$ and $H_{\text{tree}'}$ respectively by choosing 5 pairs of nodes, where the nodes of each pair are at different levels, and make the higher level node the direct parent of the lower level one.

### 4.2 FDS Models Training

We experiment with two variations of FDS training: FDS is trained using the original objective in (9) whereas FDS$_\forall$ incorporates the $\forall$-objective following §3.6. The hypernymy score of each model, given by (7), is averaged over two runs of different random seed. We empirically find that setting




(a) Four chains ($H_{\text{chains}}$) (b) Tree ($H_{\text{tree}}$) (c) Tree with overlapping contexts (underlined; $H'_{\text{tree}}$) (d) DAG ($H_{\text{DAG}}$) (e) DAG with overlapping contexts (underlined; $H'_{\text{DAG}}$)

Figure 3: Examples of the topologies of the synthetic taxonomic hierarchies.

$p = 1$ in (13) and $p = 2$ in (7) almost always give the best performances, and we only report the results in this setup. Other than the newly introduced training objective, training of the models largely follows that of Lo et al. (2023). No hyperparameter search is conducted due to the large number of experiments (details described in Appendix D).

### 4.3 Evaluation on Hypernymy Detection

We test if a model trained on the corpus learns to identify hypernymy defined in the hierarchy that generates the corpus. Concretely, a model is asked to give a score of hypernymy between every pair of nouns using (7). Performance is then measured by the area under the receiver operating characteristic curves (AUC). Unlike average precision, AUC values do not reflect changes in the distribution of classes, which is favourable since we are comparing models' performances across varying class distributions generated from different topologies.

We include two distributional methods for hypernymy detection based on the DIH in the experiments, namely WeedsPrec (Weeds et al., 2004) and invCL (Lenci and Benotto, 2012):

$$\text{WeedsPrec}(r_1, r_2) = \frac{\sum_i u_i^{(r_1)} \mathbb{1}_{u_i^{(r_2)} > 0}}{\sum_i u_i^{(r_1)}}$$

$$\text{invCL}(r_1, r_2) = \sqrt{\text{CL}(r_1, r_2)(1 - \text{CL}(r_2, r_1))},$$

$$\text{where } \text{CL}(r_1, r_2) = \frac{\sum_i \min\left(u_i^{(r_1)}, u_i^{(r_2)}\right)}{\sum_i u_i^{(r_1)}}.$$

They measure the context inclusion of $r_1$ by $r_2$ and invCL measures also the non-inclusion of $r_2$ by $r_1$. Their distributional space is constructed by first counting co-occurrences of adjacent predicates in the preprocessed DMRS graphs, then transforming the resulting matrix using positive pointwise mutual information. Each row vector $u^{(r_i)}$ in the transformed matrix represents a predicate $r_i$.

Table 2 and Table 3 show the results of FDS models when trained on the DIH and rDIH corpora respectively (a visualization of results on $H_{\text{chains}}$ is provided in Appendix E). As expected, FDS, WeedsPrec and invCL are shown to work on the DIH corpora, and only $\text{FDS}_\forall$ works on the rDIH corpora. Reversing the FDS models on respective corpora yields substantially worse performances. In particular, $\text{FDS}_\forall$ attains AUCs of about 0.2 on the DIH corpora means hypernymy predictions are even mostly reversed, which in turn reflects the effectiveness of the universal objective when $\text{FDS}_\forall$ interprets the subsumption of contexts reversely based on the rDIH. Moreover, hierarchies with overlapping contexts and multiple direct hypernyms are not harder for FDS than those without. Scaling up to the huge WordNet hierarchy $H_{\text{WN}}$ results in a slight drop in AUC for FDS on the DIH corpus, and markedly worse performances for $\text{FDS}_\forall$. While our preset hyperparameters work nicely on all other settings, it is possible that $\text{FDS}_\forall$ requires a different set of hyperparameters to perform optimally on the rDIH corpora generated from huge hierarchies.

| Model | $H_{\text{chains}}$ | $H_{\text{tree}}$ | $H'_{\text{tree}}$ | $H_{\text{DAG}}$ | $H'_{\text{DAG}}$ | $H_{\text{WN}}$ |
|---|---|---|---|---|---|---|
| FDS | .990 | .994 | .995 | .995 | .995 | .940 |
| $\text{FDS}_\forall$ | .925 | .206 | .210 | .214 | .221 | .788 |
| WeedsPrec | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| invCL | 1.000 | 1.000 | 1.000 | 1.000 | .999 | 1.000 |

Table 2: AUC of models trained on the DIH corpora.

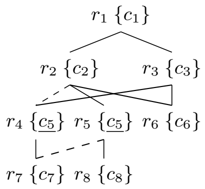
| Model | $H_{\text{chains}}$ | $H_{\text{tree}}$ | $H'_{\text{tree}}$ | $H_{\text{DAG}}$ | $H'_{\text{DAG}}$ | $H_{\text{WN}}$ |
|---|---|---|---|---|---|---|
| FDS | .876 | .842 | .793 | .752 | .688 | .444 |
| $\text{FDS}_\forall$ | .988 | .983 | .978 | .981 | .977 | .675 |
| WeedsPrec | .900 | .675 | .619 | .613 | .556 | .809 |
| invCL | .900 | .355 | .280 | .236 | .276 | .564 |

Table 3: AUC of models trained on the rDIH corpora.

We further produce new corpora by combining the DIH and rDIH corpus of each topology. The resulting corpora still follow the DIH. In this setup, instead of applying the same FDS training objective across the whole corpus, the $\forall$-objective can be added only when there is a universal quantifier

| Model | $H_{\text{chains}}$ | $H_{\text{tree}}$ | $H_{\text{DAG}}$ | $H_{\text{WN}}$ |
|---|---|---|---|---|
| FDS | .947 | .991 | .990 | .946 |
| $\text{FDS}_\forall$ | .956 | .358 | .378 | .183 |
| $\text{FDS}_{(\forall)}$ | .999 | .986 | .984 | .992 |
| WeedsPrec | .950 | .996 | .993 | .993 |
| invCL | .958 | .891 | .864 | .868 |

Table 4: AUC of models trained on combined corpora.




(a) An example taxonomy with missing relations. The siblings $r_4$ and $r_5$ have the same context $c_5$. Dashed lines show the relations to be removed from training.

| Hyp. | Noun | Contexts |
|---|---|---|
| DIH | $r_4$ | $\{c_5, c_7\}$ |
| | $r_5$ | $\{c_5, c_8\}$ |
| rDIH | $r_4$ | $\{c_1, c_3, c_5\}$ |
| | $r_5$ | $\{c_1, c_2, c_5\}$ |

(b) Contexts that $r_4$ and $r_5$ in a would appear with in the new (r)DIH corpora.

Figure 4: Illustration of the setup for testing distributional generalization.

(e.g., *every*). We name this FDS model $\text{FDS}_{(\forall)}$. Table 4 shows the results. The DIH methods (FDS, WeedsPrec, and invCL) perform well as expected, whereas invCL performs worse since it also measures non-inclusion which is undermined by the rDIH half. There are several interesting insights into FDS models. First, FDS still works on the corpora with an rDIH half because it is still valid to say '*a dog grows*', as mentioned in §2. Second, $\text{FDS}_{(\forall)}$ is as good as FDS across topologies and even better on $H_{\text{chains}}$ and $H_{\text{WN}}$. This implies that the $\forall$-objective indeed captures simple universal quantifications and can be used compatibly with the original training method on a corpus with varying quantifications. Third, on $H_{\text{WN}}$, $\text{FDS}_{(\forall)}$ performs better than FDS on the DIH corpora and much better than $\text{FDS}_\forall$ on the rDIH corpora. This reflects that the $\forall$-objective is more effective when mixed with the original mode of training.

### 4.4 Evaluation on Distributional Generalization

We also test if the distributional generalization power mentioned in §3.4 exists in FDS. We construct a new corpus from a hierarchy with removed hypernymy information. Fig. 4 illustrates the idea with an example hierarchy of nouns and the contexts that would appear with the nouns in the new corpora obtained. If upward (downward) distributional generalization exists in a model, based on that $r_4$ and $r_5$ share $c_5$ as their contexts, it should identify the hypernyms (hyponyms) of $r_5$ ($r_4$) as the *candidate hypernyms (hyponyms)* of $r_4$ ($r_5$) after training on the new corpus. That is, we expect $\forall r_j \in \{r_5, r_6, r_7, r_8\}$: $s(r_4, r_2) > s(r_4, r_j)$ if upward generalization exists, and $\forall r_j \in \{r_1, r_2, r_3, r_4, r_6\}$: $s(r_7, r_5) > s(r_j, r_5)$ if downward exists in FDS.

In our experiments, we sample five nouns from the $H'_{\text{DAG}}$ hierarchy. Then, for each of these nouns $\tilde{r}$, we equate the contexts set of $\tilde{r}$ to that of one of its siblings and remove the hypernymy (hyponymy) information of their common parent (daughter) from $\tilde{r}$ when creating the new corpora.

| Model | Hypothesis | Upward | Downward |
|---|---|---|---|
| FDS | DIH | .922 | .742 |
| $\text{FDS}_\forall$ | rDIH | .976 | .998 |

Table 5: Mean AUC for distributional generalizations.

For each $\tilde{r}$, we compute the hypernymy score of between $\tilde{r}$ and each of the candidate hypernyms, and between $\tilde{r}$ and a random noun. We measure the performance with mean AUC, averaged over the five chosen $\tilde{r}$. Table 5 shows that both upward and downward distributional generalizations exist when the corpus follows either the DIH or rDIH, and to a larger extent on the rDIH corpus.

### 4.5 Summary

The experimental results confirm that: (1) the original FDS models learn hypernymy under the DIH, (2) the proposed $\forall$-objective captures universal quantifications and enables hypernymy learning under the rDIH, and (3) FDS models can generalize about nouns with incomplete contexts in a corpus using distributional information.

## 5 Experiments on Real Data Sets

Seeing how FDS performs on restricted synthetic data sets is helpful for understanding models' behaviour but it does not immediately tell us more about hypernymy learning from open classes of sentences. Therefore, we perform further experiments using a real corpus and data sets for hypernymy.

### 5.1 FDS Models Training

**Training Data.** FDS models are trained on Wikiwoods (Flickinger et al., 2010; Solberg, 2012), which provide linguistic analyses of 55m sentences (900m tokens) in English Wikipedia. Each of the sentences was parsed by the PET parser (Callmeier, 2001; Toutanova et al., 2005) using the 1212 version of the ERG, and the parses are ranked by a ranking model trained on WeScience (Ytrestøl et al., 2009). We extract the DMRS graphs from Wikiwoods using Pydelphin[2] (Copestake et al., 2016). After preprocessing, there are 36m sentences with 254m tokens.

**Model Configurations.** Although quantifications are annotated in Wikiwoods, neither of the proposed training objectives is entirely applicable in general. For example, even for a sentence of modest complexity like '*every excited dog barks*', it requires a universal quantification over the intersection of the set of dogs and excited entities. However, set intersection is not modelled by FDS. In our experiments, we apply either FDS or $\text{FDS}_\forall$ described in §4.2 to every training instance. We also test an additional model $\text{FDS}_{\forall/2}$ where the ∀-objective is scaled by 0.5. Each model is trained for 1 epoch and the results of each model are averaged over two random seeds as discussed in §4.2.

### 5.2 Evaluation Method

We test the trained models on four English hypernymy data sets for nouns, namely Kotlerman2010 (Kotlerman et al., 2010), LEDS (Baroni et al., 2012), WBLESS (Weeds et al., 2014), and EVALution (Santus et al., 2015). Each of them consists of a set of word pairs, each with a label indicating whether the second word is a hypernym of the first word. We removed the out-of-vocabulary instances from all data sets, and non-nouns from EVALution during the evaluation. Table 6 reports the statistics of the test sets data. We report the AUC as in §4.

| **Test Set** | **# Positive** | **# Negative** |
|---|---|---|
| Kotlerman2010 | 880 [831] | 2058 [1919] |
| LEDS | 1385 [1344] | 1385 [1342] |
| WBLESS | 834 [830] | 834 [813] |
| Evalution | 1592 [1352] | 4561 [3241] |

Table 6: Class distributions of test sets. In brackets are the numbers after removal of OOV instances and non-nouns.

In addition, we use WBLESS for further performance analysis, which provides categorizations of the negative instances. Each of the negative instances is either a hyponymy pair, co-hyponymy pair, meronymy pair, or pair of random nouns.

[2] https://github.com/delph-in/pydelphin

### 5.3 Baselines

Following Roller et al. (2018), we implement five distributional methods and train them on Wikiwoods using the distributional space described in §4.3. Apart from the two DIH measures in §4.3, we use SLQS (Santus et al., 2014), a word generality measure that rests on another hypothesis that general words mostly appear in uninformative contexts:

$$\text{SLQS}(r_1, r_2) = 1 - \frac{E_{r_1}}{E_{r_2}},$$
$$\text{where } E_{r_i} = \text{median}_{j=1}^{N}[H(c_j)].$$

For each word $r_i$, the median of the entropies of $N$ most associated contexts (as measured by local mutual information) is computed, where $H(c_j)$ denotes the Shannon entropy of the associated context $c_j$. Then, SLQS compares the generality of two words by their medians. $N$ is chosen to be 50 following Santus et al. (2014). We also include cosine similarity (Cosine) of $u^{(r_1)}$ and $u^{(r_2)}$, and SLQS–Cos, which multiplies the SLQS measure by Cosine since the SLQS measure only considers generality but not similarity.

### 5.4 Results

| **Model** | Kotlerman2010 | LEDS | WBLESS | Evalution |
|---|---|---|---|---|
| Cosine | .701 | .782 | .620 | .526 |
| WeedsPrec | .674 | .897 | .709 | .650 |
| invCL | .679 | .905 | .707 | .620 |
| SLQS | .491 | .480 | .568 | .532 |
| SLQS–Cos | .489 | .477 | .557 | .532 |
| FDS | .473 | .650 | .508 | .459 |
| $\text{FDS}_{\forall/2}$ | .558 | .759 | .660 | .583 |
| $\text{FDS}_\forall$ | .550 | .735 | .655 | .554 |

Table 7: AUC on the test sets.

Table 7 shows the results on the four test sets. The DIH baselines are competitive and nearly outperform all models across the test sets. $\text{FDS}_\forall$ and $\text{FDS}_{\forall/2}$ both outperform FDS considerably across the test sets. This reflects that including the proposed ∀-objective in training is useful for extracting hypernymy information in a corpus. Compared to the 2.7-billion-token corpus used by Santus et al. (2014) in training SLQS, we suggest that the Wikiwoods corpus is too small for SLQS to obtain meaningful contexts of the median entropy: setting $N$ to be small results in frequent contexts that are not representative of the nouns, whilst setting it large would require a disproportionate number of contexts for the infrequent words.

| Model | Hyponymy | Co-hyponymy | Meronymy | Random |
|---|---|---|---|---|
| Cosine | .511 | .369 | .683 | .924 |
| WeedsPrec | .754 | .615 | .631 | .843 |
| invCL | .745 | .568 | .652 | .872 |
| SLQS | .606 | .551 | .590 | .524 |
| SLQS–Cos | .581 | .525 | .574 | .547 |
| FDS | .596 | .288 | .561 | .587 |
| $\text{FDS}_{\forall/2}$ | .783 | .612 | .549 | .704 |
| $\text{FDS}_{\forall}$ | .783 | .625 | .527 | .691 |

Table 8: AUC on the sub-categories of WBLESS.

Table 8 shows the results on the WBLESS sub-categories. It is shown that $\text{FDS}_{\forall}$ is stronger than the DIH baselines in distinguishing between hyponymy and hypernymy pairs, and between co-hyponymy and hypernymy pairs, while weaker for meronymy or random pairs. $\text{FDS}_{\forall}$ and $\text{FDS}_{\forall/2}$ outperform FDS in three out of the four sub-categories, with much higher distinguishing power for co-hyponymy and hyponymy. These imply that the $\forall$-objective makes FDS more sensitive to the relative generality than the similarity of word pairs.

## 6 Conclusion

We have discussed how Functional Distributional Semantics (FDS) can provide a truth-conditional representation for hypernymy and demonstrate that it is learnable from the distributional information in a corpus. On synthetic data sets, we confirm that FDS learns hypernymy under the Distributional Inclusion Hypothesis (DIH), and under the reverse of the DIH if the proposed objective for universal quantifications is applied. On real data sets, the proposed objective substantially improves FDS performance on hypernymy detection. We hope that this work provides insights into FDS models and hypernymy learning from corpora in general.

## Limitations

The proposed representation of hypernymy in FDS compares the semantic functions of DMRS predicate pairs. Following previous implementations of Functional Distributional Semantics, a semantic function is a linear classifier. Consequently, each DMRS predicate is assumed to have only one sense. Modelling polysemy would require more expressive parametrizations of semantic functions, which can pose additional challenges to model training, and the hypernymy representation would possibly need to be revised. Such an approach is considered out of the scope of this work.

## Ethics Statement

We anticipate no ethical issues directly stemming from our experiments.

## References


Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France. Association for Computational Linguistics.

Arthur Bražinskas, Serhii Havrylov, and Ivan Titov. 2018. Embedding words as distributions with a Bayesian skip-gram model. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1775–1789, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ulrich Callmeier. 2001. Efficient parsing with large-scale unification grammars. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany.

Xuyou Cheng, Michael Schlichtkrull, and Guy Emerson. 2023. Are embedded potatoes still vegetables? On the limitations of WordNet embeddings for lexical semantics. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8763–8775, Singapore. Association for Computational Linguistics.

Ann Copestake. 2009. **Invited Talk:** slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 1–9, Athens, Greece. Association for Computational Linguistics.

Ann Copestake, Guy Emerson, Michael Wayne Goodman, Matic Horvat, Alexander Kuhnle, and Ewa Muszyńska. 2016. Resources for building applications with dependency Minimal Recursion Semantics. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1240–1247, Portorož, Slovenia. European Language Resources Association (ELRA).

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Paula Czarnowska, Guy Emerson, and Ann Copestake. 2019. Words are vectors, dependencies are matrices: Learning word embeddings from dependency graphs. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 91–102, Gothenburg, Sweden. Association for Computational Linguistics.

Guy Emerson. 2018. *Functional Distributional Semantics: Learning Linguistically Informed Representations from a Precisely Annotated Corpus*. Ph.D. thesis, University of Cambridge.

Guy Emerson. 2020a. Autoencoding pixies: Amortised variational inference with graph convolutions for functional distributional semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3982–3995, Online. Association for Computational Linguistics.

Guy Emerson. 2020b. What are the goals of distributional semantics? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7436–7453, Online. Association for Computational Linguistics.

Guy Emerson. 2023. Probabilistic lexical semantics: From Gaussian embeddings to Bernoulli Fields. In Jean-Philippe Bernardy, Rasmus Blanck, Stergios Chatzikyriakidis, Shalom Lappin, and Aleksandre Maskharashvili, editors, *Probabilistic Approaches to Linguistic Theory*, pages 65–122. CSLI Publications.

Guy Emerson and Ann Copestake. 2016. Functional distributional semantics. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 40–52, Berlin, Germany. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database* [Website]. Massachusetts Institute of Technology (MIT) Press.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage, and Processing*, chapter 3, pages 31–50. Center for the Study of Language and Information (CSLI) Publications.

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. WikiWoods: Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Chun Hei Lo, Hong Cheng, Wai Lam, and Guy Emerson. 2023. Functional distributional semantics at scale. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 423–436, Toronto, Canada. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR), Workshop Track*.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Ludovica Pannitto, Lavinia Salicchi, and Alessandro Lenci. 2018. Refining the distributional inclusion hypothesis for unsupervised hypernym identification. *IJCoL. Italian Journal of Computational Linguistics*, 4(4-2):45–55.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for*

*Computational Linguistics*, pages 511–519, Gothenburg, Sweden. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden. Association for Computational Linguistics.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69, Beijing, China. Association for Computational Linguistics.

Lars Jørgen Solberg. 2012. A corpus builder for Wikipedia. Master's thesis, University of Oslo.

Kristina Toutanova, Christopher D. Manning, Dan Flickinger, and Stephan Oepen. 2005. Stochastic HPSG parse disambiguation using the Redwoods corpus. *Research on Language and Computation*, 3(1):83–105.

Luke Vilnis and Andrew McCallum. 2015. Word representations via Gaussian embedding. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1015–1021, Geneva, Switzerland. COLING.

Gisle Ytrestøl, Dan Flickinger, and Stephan Oepen. 2009. Extracting and annotating Wikipedia subdomains: Towards a new eScience community resource. In *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories*, pages 185–197.

Lotfi A. Zadeh. 1965. Fuzzy sets. *Information and Control*, 8(3):338–353.

## A Derivation of Hypernymy Conditions

Consider (6). $\forall z$:

$$t^{(r_H,0)}(z) > t^{(r_h,0)}(z)$$
$$S\left({v^{(r_H,0)}}^\top z + b^{(r_H,0)}\right) > S\left({v^{(r_h,0)}}^\top z + b^{(r_h,0)}\right).$$

$S$ is monotonic, so $\forall z$:

$${v^{(r_H,0)}}^\top z + b^{(r_H,0)} > {v^{(r_h,0)}}^\top z + b^{(r_h,0)}$$
$$b^{(r_H,0)} - b^{(r_h,0)} > \left(v^{(r_h,0)} - v^{(r_H,0)}\right)^\top z. \quad (15)$$

Consider $z$ within a unit hypercube, (15) is equivalent to

$$b^{(r_H,0)} - b^{(r_h,0)} > \max_{\|z\|_\infty \le 1} \left(v^{(r_h,0)} - v^{(r_H,0)}\right)^\top z$$

Note that

$$\arg\max_{z:\ \|z\|_\infty \le 1} \left(v^{(r_h,0)} - v^{(r_H,0)}\right)^\top z = \mathrm{sgn}(v_i^{(r_h,0)} - v_i^{(r_H,0)}),$$

where sgn is the sign function. Hence, we have

$$b^{(r_H,0)} - b^{(r_h,0)} > \left\|v^{(r_H,0)} - v^{(r_h,0)}\right\|_1.$$

If we consider $z$ within a unit hypersphere, (15) is equivalent to

$$b^{(r_H,0)} - b^{(r_h,0)} > \max_{\|z\|_2 \le 1} \left(v^{(r_h,0)} - v^{(r_H,0)}\right)^\top z.$$

Note that

$$\arg\max_{z:\ \|z\|_2 \le 1} \left(v^{(r_h,0)} - v^{(r_H,0)}\right)^\top z = \frac{v^{(r_h,0)} - v^{(r_H,0)}}{\left\|v^{(r_h,0)} - v^{(r_H,0)}\right\|_2}.$$

Hence, we have

$$b^{(r_H,0)} - b^{(r_h,0)} > \left\|v^{(r_H,0)} - v^{(r_h,0)}\right\|_2.$$

## B Derivation of Transitivity

$$
\begin{aligned}
&s(r_1, r_2) + s(r_2, r_3) \\
&= b^{(r_2,0)} - b^{(r_1,0)} - \left\| v^{(r_2,0)} - v^{(r_1,0)} \right\|_p \\
&\quad + b^{(r_3,0)} - b^{(r_2,0)} - \left\| v^{(r_3,0)} - v^{(r_2,0)} \right\|_p \\
&= b^{(r_3,0)} - b^{(r_1,0)} - \\
&\quad \left( \left\| v^{(r_2,0)} - v^{(r_1,0)} \right\|_p + \left\| v^{(r_3,0)} - v^{(r_2,0)} \right\|_p \right).
\end{aligned}
$$

By the Minkowski inequality, the last term is greater than $\left\| v^{(r_3,0)} - v^{(r_1,0)} \right\|_p$. Besides, when $s(r_1, r_2) > 0$ and $s(r_2, r_3) > 0$, $s(r_1, r_2) + s(r_2, r_3) > 0$. Hence,

$$
\begin{aligned}
b^{(r_3,0)} - b^{(r_1,0)} - \left\| v^{(r_3,0)} - v^{(r_1,0)} \right\|_p &> 0 \\
s(r_3, r_1) &> 0.
\end{aligned}
$$

## C Intuition behind Hypernymy Learning by FDS under the DIH

We hypothesize that the way that FDS models are trained allows hypernymy learning under the DIH. During training described in §3.5, the approximate posterior distributions of pixies are first inferred from the observed graph. After variational inference, the semantic functions of the observed predicates are optimized to be true of the inferred pixie distributions. This process is analogous to the following process under a model-theoretic approach: the entities described by a sentence are first identified, and then the truth conditions of predicates over the entities are updated as asserted by the sentence.

Under the DIH, the contexts of nouns are also contexts of their hypernyms. The local predicate–argument information of nouns, i.e. contexts, is thus repeated for their hypernyms for inference during training. Consequently, the semantic functions of hypernyms are trained to return values at least as high as those of their hyponyms over the pixie distributions inferred from the same contexts. The additional contexts appearing exclusively with the hypernyms will further increase the probability of truths of the hypernyms over the pixie space. By (6), hypernymy should thus be learnt under the DIH.

## D Training Details

### D.1 Hyperparameters and Tuning

For all the experiments, the hyperparameters of the FDS models largely follow that of $\text{FDSAS}_{\text{id}}$ in Lo et al. (2023) except that we set $\beta_1$ to 0.5 instead of 0. The consequence is that the inferred pixie distributions during VAE training will be centred closer to the origin. This is motivated by our decision in §3.4 that pixies are only meaningful within the unit hypersphere or hypercube.

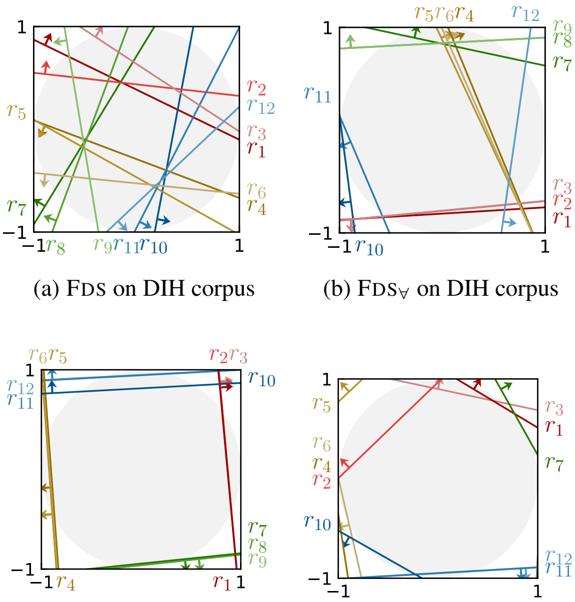
Here are the changes exclusive to the experiments on the synthetic data sets. We set $K$ to 1 and perform random negative sampling without weighing by unigram distribution, which trains models maximally using information from the data with minimal assumptions needed for the negative samples. We set the learning rate to 0.01. For experiments on $H_{\text{chains}}$, $d$ is set to 2. For $H_{\text{WN}}$, $d$ is set to 50. For the remaining topologies, $d$ is set to 10. The models are trained for 2 epochs for $H_{\text{WN}}$, and 5000 epochs for the rest.

### D.2 Computational Configurations

All models are implemented in PyTorch (Paszke et al., 2019) and trained with distributed data parallelism on three NVIDIA GeForce GTX 1080 Ti. Training a run of FDS or $\text{FDS}_\forall$ on Wikiwoods takes about 360 GPU hours.

## E Visualization of Semantic Functions

A visualization of results on $H_{\text{chains}}$ is provided in Fig. 5. As seen in Figs. 5a and 5c, training FDS on the DIH corpus and $\text{FDS}_\forall$ on the rDIH corpus both result in four nicely divided pixie subspaces, each for one of the four hypernymy chains, as shown in the plots on the left column. In contrast, applying the other models sometimes gives badly learnt semantic functions, as shown in Figs. 5b and 5d. For example, $t^{(r_{12},0)}$ points to the opposite direction of $t^{(r_{10},0)}$ and $t^{(r_{11},0)}$ in Fig. 5b.




(a) FDS on DIH corpus

(b) $\text{FDS}_\forall$ on DIH corpus

(c) $\text{FDS}_\forall$ on rDIH corpus

(d) FDS on rDIH corpus

Figure 5: Visualization of semantic functions of a run trained on $H_{\text{chains}}$. Each plot shows a pixie space in a unit square (unit circle in grey). Each line plots $t^{(r_i,0)}(z) = 0$ and the arrow points to the pixie subspace where $t^{(r_i,0)}(z) > 0$.