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

Chun Hei Lo
The Chinese University of Hong Kong
chlo@se.cuhk.edu.hk

Guy Emerson
University of Cambridge
gete2@cam.ac.ukk

Abstract

Functional Distributional Semantics (FDS) attempts to model the truth-conditional meaning of words by learning from the distributional information in a corpus. Since hypernymy is formally defined by set subsumptions of extensions, we hypothesize that hypernymy can naturally be embedded in FDS. In this paper, we aim to address (1) how hypernymy can be represented in FDS, (2) if FDS learns hypernymy by training on a corpus, and (3) how it learns, if it does. We demonstrated that FDS learns hypernymy on synthetic data sets that follow the Distributional Inclusion Hypothesis. Then, we propose a new training objective that effectively handles the reverse of the hypothesis. Finally, we provide further results on real data sets and show that the new training objective of FDS improves the performance on hypernymy detection.

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 linguistically and 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) (for a discussion, see: Emerson, 2020b, 2023). On top of its theoretical favour, Lo et al. (2023) also demonstrated FDS models in action and showed that they are very competitive in the semantic tasks of semantic composition and verb disambiguation.

Hypernymy, formally defined as the subsumption of extensions between two word senses, can be modelled with truth-conditional functions. Although FDS provides the tools for hypernymy, it is not obvious whether hypernymy can be learnt by merely training an FDS model on a corpus, and if yes, how it is learnt.

To acquire hypernymy automatically from a corpus, one of the ways is through the use of distributional information in a corpus. In this class of methods, hypernymy is learnt in an unsupervised manner given certain hypotheses about the distributional properties of the corpus. One of such hypotheses is the Distributional Inclusion Hypothesis (DIH) (Weeds et al., 2004; Geffet and Dagan, 2005), which states that the meaning of a word r_1 entails another word r_2 if and only if all the typical contexts (features) of r_1 occur also with r_2. In this work, we show that FDS models learn hypernymy on a restricted class of corpus which follows DIH.

In this paper, we first give a brief introduction to FDS in §2 and its connection to truth-conditional semantics. Then, we describe how hypernymy can be represented in FDS in §3. Then, we discuss how FDS can learn hypernymy under the Distributional Inclusion Hypothesis and the reverse of it in §4. Finally, we present experimental results on applying FDS models on both synthetic data sets and real data sets in §5 and §6 respectively.

2 Functional Distributional Semantics

Model-theoretic semantics sees meaning in terms of an extensional model structure, which consists of a set of atomic *entities*, and a set of *predicates*, each of which is true or false of the entities. In parallel, Functional Distributional Semantics (FDS) represents an entity by a *pixie*, and predicate by a truth-conditional *semantic function* that takes pixie(s) as input and returns the probability of truth. In this paper, we follow the implementation of FDS by Lo et al. (2023), which was more scalable and performant over previous models. We briefly describe it here.

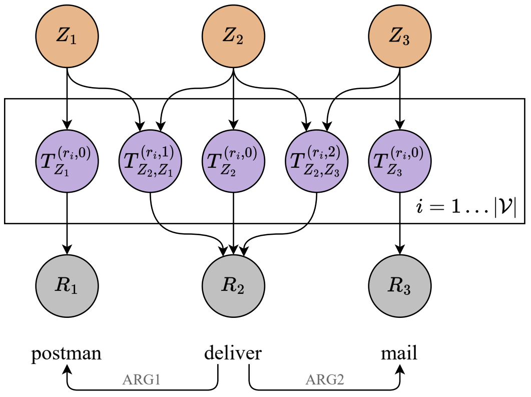


Figure 1: Probabilistic graphical model of FDS for generating the words in '*postman deliver mail*'. Only $R_1 =$ *postman*, $R_2 =$ *deliver*, $R_3 =$ *mail* are observed.

2.1 Probabilistic Graphical Models

The framework is formalized in terms of a family of probabilistic graphical models. Each of them describes the generative process of predicates in the semantic graph of a sentence. Fig. 1 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.

2.2 Semantic Functions

Instead of treating a predicate as an indicator function, FDS models the probability that it is true of the pixie(s) with unary and binary *semantic functions*, as in (1) and (2) respectively. Allowing continuous values also accounts for vagueness, a crucial aspect of meaning.

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

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

The functions are implemented as linear classifiers in (3) and (4), where S denotes the sigmoid function and $z_{i,j}$ denotes the concatenation of z_i and z_j.

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

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

2.3 Model Training

FDS models are trained on graphs of Dependency Minimal Recursion Semantics (DMRS; Copestake et al., 2005; Copestake, 2009), which are derived using the English Resource Grammar (ERG; Flickinger, 2000, 2011). Quantifiers and scopal information are removed from the graphs before training, leaving us with just the predicate–argument structure expressed by a sentence.

Given an observed DMRS graph G with n pixies $Z_1 \ldots Z_n$, model parameters are optimized in an unsupervised manner to maximize (5), which is reformulated from the β-VAE (Higgins et al., 2017). The approximate posterior distribution q_ϕ is taken to be n spherical Gaussian distributions, each with mean μ_{Z_i} and covariance $\sigma^2_{Z_i} I$. Such distributions are inferred from both the local predicate–argument structure of each predicate and global topical information in the graph. For instance, the approximate posterior distribution of the pixie Z_1 of *postman* in Fig. 1 is inferred from the direct argument information, $\xleftarrow{\text{ARG1}}$ *deliver*, and the indirect topical predicate, *mail*. This inference method plays an important role in hypernymy learning as we will discuss in §4.2.

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{n} \mathcal{C}_i + \sum_{r_i \xrightarrow{\text{ARG}[a]} r_j \text{in } G} \mathcal{C}_{i,j,a} \\ &- \frac{d}{2} \sum_{i=1}^{n} \beta_1 \mu^2_{Z_i} + \beta_2 \left(\sigma^2_{Z_i} - \ln \sigma^2_{Z_i}\right)\end{aligned} \tag{5}$$

The first two terms in (5), detailed in (6) and (7), aim to maximize the truthness of observed predicates and the falsehood of the negatively sampled ones r' over the inferred pixie distribution q_ϕ, whereas the last term in (5) is the regularization term on the means and variances of the inferred pixies distributions.

$$\mathcal{C}_i = \ln \mathbb{E}_{q_\phi}\left[t^{(r_i,0)}(z_i)\right] + \sum_{r'\in N(i)} \ln \mathbb{E}_{q_\phi}\left[1 - t^{(r',0)}(z_i)\right] \tag{6}$$

$$\mathcal{C}_{i,j,a} = \ln \mathbb{E}_{q_\phi}\left[t^{(r_i,a)}(z_i, z_j)\right] + \sum_{r'\in N(i)} \ln \mathbb{E}_{q_\phi}\left[1 - t^{(r',a)}(z_i, z_j)\right] \tag{7}$$

3 Representing Hypernymy in FDS

In truth-conditional semantics, with respect to a set of entities D, r_H is a hypernym of r_h if and only if $\phi(r_h, r_H)$ is true in (8).

$$\phi(r_h, r_H) = \forall x{\in}D\colon r_h(x) \implies r_H(x) \tag{8}$$

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 (8) to probabilistic counterpart for a score on hypernymy $P\left(T_Z^{(r_H,0)}{=}\top \,\middle|\, T_Z^{(r_h,0)}{=}\top\right)$. However, this conditional probability is unavailable since only $P\left(T_Z^{(r_H,0)}{=}\top \,\middle|\, z\right)$ and $P\left(T_Z^{(r_h,0)}{=}\top \,\middle|\, z\right)$ are modelled by FDS.

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

$$\forall z \in \mathbb{R}^d\colon t^{(r_H,0)}(z) > t^{(r_h,0)}(z) \tag{9}$$

In this setup, (9) can only be true when $v^{(r_h,0)} = kv^{(r_H,0)}$ where $k \neq 0$, which is impossible to be obtained in practice from model training. Therefore, we restrict the pixie space and only consider pixies in a unit d-sphere or d-cube to be meaningful. With (3) and (4), r_H is considered the hypernym of r_h if and only if $s(r_h, r_H) > 0$ in (10), where $p \in \{1, 2\}$ (derivations in Appendix A).

$$s(r_h, r_H) = b^{(r_H,0)} - b^{(r_h,0)} - \left\|v^{(r_H,0)} - v^{(r_h,0)}\right\|_p \tag{10}$$

Note that the transitivity of (8) is paralleled:

$$\forall r_1, r_2, r_3\colon s(r_1, r_2) > 0 \wedge s(r_2, r_3) > 0 \implies s(r_1, r_3) > 0 \tag{11}$$

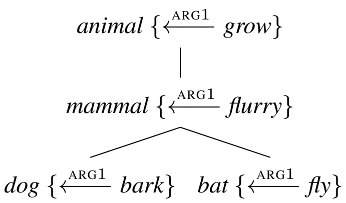

animal $\{\xleftarrow{\text{ARG1}}$ *grow*$\}$
mammal $\{\xleftarrow{\text{ARG1}}$ *flurry*$\}$
dog $\{\xleftarrow{\text{ARG1}}$ *bark*$\}$
bat $\{\xleftarrow{\text{ARG1}}$ *fly*$\}$

Figure 2: A taxonomic hierarchy of nouns. Next to each noun is the set of contexts that appear with it and its descendants.

Corpus 1: DIH	**Corpus 2: rDIH**
a dog barks	*every dog barks*
a mammal barks	*every dog is flurry*
an animal barks	*every dog grows*
a bat flies	*every bat flies*
a mammal flies	*every bat is flurry*
an animal flies	*every bat grows*
a mammal is flurry	*every mammal is flurry*
an animal is flurry	*every mammal grows*
an animal grows	*every animal grows*

Table 1: Corpora generated from the hierarchy in Fig. 2. Existential and universal quantifications result in two corpora that follow DIH and rDIH respectively.

4 Learning under (Reverse of) Distributional Inclusion Hypothesis

Given the power of representing hypernymy in FDS, we explore in this section whether hypernymy can be learnt by FDS models from just text, and if yes, how.

4.1 Quantifications and Distributional Inclusion Hypothesis

In this section, we revisit the Distributional Inclusion Hypothesis (DIH) and explain how quantifications support or undermine the hypothesis.

DIH asserts that the typical characteristic features of r_h are expected to appear with r_H if and only if r_H is a hypernym of r_h. Behind the hypothesis is the assumption that all true assertions appear in the corpora, i.e., the narrow-scope negation readings of sentences such as '*a / every dog does not fly*' are implied. Geffet and Dagan (2005) considers syntax-based context. We suggest that semantics-based ones be more suitable since syntactic difference does not necessarily contribute to semantic ones, e.g., passivization.

Consider the simple hierarchy in Fig. 2. Table 1 shows the sentences that are true with respect to the hierarchy. It can be seen that DIH applies to Corpus 1. For example, the set of contexts of dog $(\{\xleftarrow{\text{ARG1}}$ *bark*$\})$ is a subset of those of mammal

($\{\xleftarrow{\text{ARG1}}$ *bark*, $\xleftarrow{\text{ARG1}}$ *fly*, $\xleftarrow{\text{ARG1}}$ *flurry*$\}$). However, substituting existential with universal quantifications results in the reverse of DIH (rDIH) in Corpus 2, where the set of contexts of *mammal* then becomes a subset of that of *dog*.

With this simple example, we explain how methods that rely on DIH as a cue for hypernymy can be undermined. We do not discuss further more complex sentence structures because then the entailment conditions of sentences will not be trivial, and such a corpus will not strictly align with DIH or rDIH. For instance, with a restricted relative clause, *every dog that is trained is gentle* does not entail *every Chihuahua is gentle*. Therefore, *Chihuahua* may not appear in the context of $\{\xleftarrow{\text{ARG1}}$ *gentle*$\}$ even if *Chihuahua* is a hyponym of *dog*.

4.2 Our Hypothesis: FDS Learns Hypernymy under DIH

We hypothesize that the way that FDS models are trained allows hypernymy to be learnt from a corpus that follows DIH. Below is the intuition behind it.

FDS models are trained following the variational autoencoding method described in §2.3. Essentially, the approximate posterior distributions of pixies are first inferred from the observed graph. Then, 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.

Second, the contexts of nouns are also contexts of their hypernyms in DIH. 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 higher than those of their hyponyms over the pixie distributions inferred from the same contexts, aligning with (9).

4.3 Universal Quantification in FDS for rDIH

FDS assumes that each observed predicate refers to only one point but not a region in the pixie space. This corresponds to the interpretation that all nouns are uniquely existentially quantified (∃!), so only a corpus that follows DIH can be handled by FDS.

To this end, we propose a method to enable FDS to be trained on simple sentences with universal quantification. Concretely, we want to optimize semantic functions with respect to not a point but a region in the pixie space. We add the following **universal quantification objective** to the original objective proposed by Lo et al. (2023):

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

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

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

$$\begin{aligned} \mathcal{U}_{i,j,a} = {} & \sum_{r'} \min\left(0, -s_0(r_i, r')\right) \\ & + \sum_{(r'',a'')} \min\left(0, -s_{a''}(r'', r_j)\right) \end{aligned} \tag{14}$$

Note that (13) is modified based on (10), previously defined for classifying hypernymy.

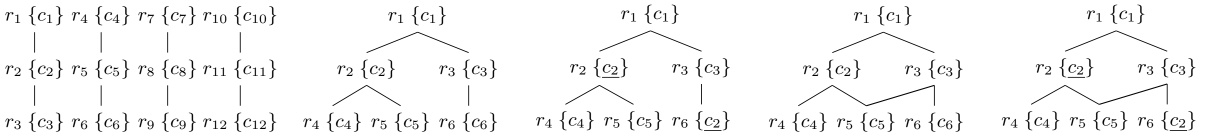
To explain (12), let's take the sentence *every dog barks* as an example. The first term inside the summation in (12) enforces that extensions of r_j is a subset of that of prototypical argument a of r_i, i.e., the set of dogs should be contained in the set of agents that barks. The second term, described in (14), incorporates negative samples. The negative samples r' for r_j are generated by randomly sampling K nouns, and (r'', a'') for r_i by randomly sampling K verbs or adjectives, each with an argument role. Then, (14) requires that it is false to universally quantify the referents of the noun r' in $r' \xleftarrow{\text{ARG}[a]} r_i$ and r_j in $r_j \xleftarrow{\text{ARG}[a'']} r''$. For the example, it means that the following two sentences are both considered false: *every dog is owned* and *every cat barks*, where $r' = $ *cat*, $r'' = $ *own* and $a'' = 2$.

5 Experiments on Synthetic Data Sets

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

5.1 Synthetic Data Sets under (r)DIH

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


r_1 {c_1} r_4 {c_4} r_7 {c_7} r_{10} {c_{10}}
r_2 {c_2} r_5 {c_5} r_8 {c_8} r_{11} {c_{11}}
r_3 {c_3} r_6 {c_6} r_9 {c_9} r_{12} {c_{12}}
r_1 {c_1}
r_2 {c_2} r_3 {c_3}
r_4 {c_4} r_5 {c_5} r_6 {c_6}
r_1 {c_1}
r_2 {$\underline{c_2}$} r_3 {c_3}
r_4 {c_4} r_5 {c_5} r_6 {$\underline{c_2}$}
r_1 {c_1}
r_2 {c_2} r_3 {c_3}
r_4 {c_4} r_5 {c_5} r_6 {c_6}
r_1 {c_1}
r_2 {$\underline{c_2}$} r_3 {c_3}
r_4 {c_4} r_5 {c_5} r_6 {$\underline{c_2}$}

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

Figure 3: Example of topologies of synthetic taxonomic hierarchies.

1. **Create the taxonomic hierarchy.** Define a set of nouns, the hypernymy relations between them, and a finite verb or predicative adjective with the argument role for each noun as the contexts of them and their hyponyms.
2. **Choose a hypothesis.** DIH or rDIH.
3. **Create a corpus.** Create sentences in the form '[quantifier] [noun] [context]' following the chosen hypothesis and the defined hierarchy.

5.1.1 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 five classes of topologies used. We expect that directed acyclic graphs (H_{DAG} and H'_{DAG}) be harder topologies than trees (H_{tree} and H'_{tree}), and topologies with overlapping contexts (H_{tree} and H'_{DAG}) be harder than those without (H_{tree} and H_{DAG}). In addition, we test H_{chains} with pixie dimensionality $d = 2$. A 2-D pixie space provides adequate expressive power for embedding 4 chains of hypernymy. It also allows lossless visualization of the semantic functions. To test hypernymy learning at scale on an actual hierarchy, we also test our models on a subgraph of WordNet's hierarchy (H_{WN}).

Every node in the hierarchy consists of a noun and a semantic context. The topology of the H_{chains} used in the experiment is exactly as depicted in Fig. 4. H_{WN} is created out of the synset *animal.n.01* in WordNet, which is the root, and its hyponymic synsets. To keep the size of the hierarchy reasonable for pair-wise hypernymy scoring, we keep only the synsets whose shortest distance to the root is less than 6. This results in 982 nodes. For the remaining hierarchies, each of them consists of 153 nodes with a height of 5. For H_{tree}, the first level is a root node, and a node at the h^{th} level has $(h + 1)$ direct children. $H_{\text{tree}'}$ is created from H_{tree} by choosing 5 pairs of nodes and making each of them to share a context set. H_{DAG} and H'_{DAG} are created from H_{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.

5.2 FDS Models Training

We experiment with two variations of FDS training: FDS is trained using the original objective in (5) whereas FDS$_\forall$ incorporates the universal quantification objective following §4.3. Each model is trained on every synthetic corpus. We empirically find that setting $p = 1$ in (13) and $p = 2$ in (10) almost always give the best performances, and we only report the results in this setup. Other than the newly introduced training objective, models training largely follows that of Lo et al. (2023). However, some minor yet crucial modifications have to be made. Further details about model training are described in §B.1.

5.3 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 (10). 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.

Table 2 and Table 3 show the results of FDS models on different topologies when trained on a DIH and rDIH corpus respectively. FDS is shown to

Model	H_{chains}	H_{tree}	H'_{tree}	H_{DAG}	H'_{DAG}	H_{WN}
FDS	.992	.982	.984	.986	.986	.963
FDS$_\forall$	.800	.211	.219	.219	.228	.100

Table 2: AUC of models trained on synthetic DIH corpora generated from different topologies of hierarchy.

Model	H_{chains}	H_{tree}	H'_{tree}	H_{DAG}	H'_{DAG}	H_{WN}
FDS	.900	.805	.758	.718	.642	.853
FDS$_\forall$	.994	.981	.978	.981	.976	.935

Table 3: AUC of models trained on synthetic rDIH corpora generated from different topologies of hierarchy.

work on the DIH corpus, and FDS$_\forall$ on rDIH corpus. Reversing the models on respective corpora yields substantially worse performances. In particular, FDS$_\forall$ attains AUCs of about 0.2 on the DIH corpus means hypernymy predictions are even largely reversed, which in turn reflects the effectiveness of the universal objective when FDS$_\forall$ interprets the implication of context sets subsumption on hypernymy based on rDIH. Hierarchies with overlapping contexts and multiple direct hypernyms are not harder cases than those without. Both models are also shown to be robust when scaling up to the larger WordNet hierarchy, with only a slight drop in performances.

Fig. 4 visualizes the semantic functions trained on the corpora of H_{chains}. Training FDS on the DIH corpus and FDS$_\forall$ on the rDIH corpus results 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, e.g., $t^{(r_{12},0)}$ points to the opposite direction of $t^{(r_{10},0)}$, $t^{(r_{11},0)}$ for FDS$_\forall$ on DIH corpus.

5.4 Evaluation on Distributional Generalization

Apart from testing models directly on the mentioned topologies, we also test if distributional generalization power exists in FDS. Intuitively, when the information about *mammal* being a hypernym of *dog* is missing, consequently the breakdown of (r)DIH, knowing that both *dogs* and *foxes* share the same contexts (e.g., $\{\xleftarrow{\text{ARG1}}$ *bark*$\}$) is indicative that they share common hypernyms, e.g., *mammal*. This also applies to hyponymy, e.g., *machine* and *system* sharing $\{\xleftarrow{\text{ARG1}}$ *complex*$\}$ and sharing *computer* as their hyponym. Such generalization power is largely absent in models that rely purely on (r)DIH. This can be helpful (generalizing out of missing data) or harmful (generalizing too strongly) depending on particular use cases.

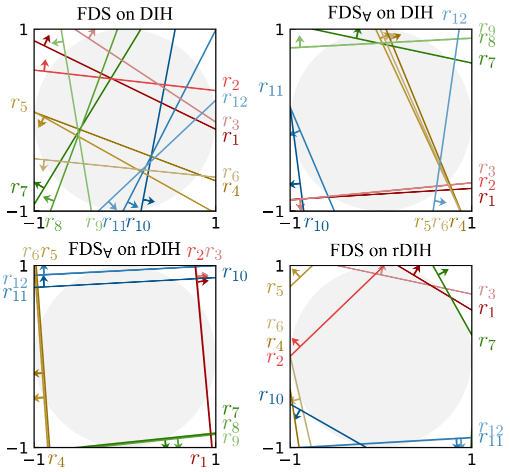
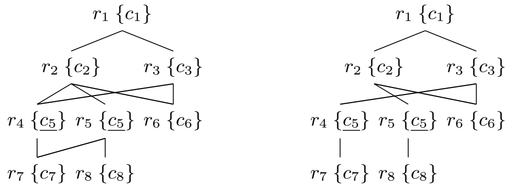


Figure 4: Visualization of semantic functions trained on H_{chains}. Each plot shows a pixie space in a unit 2-cube (2-sphere in grey). Each line plots $t^{(r_i,0)}(z) = 0$ and the arrow points to the pixie subspace where $t^{(r_i,0)}(z) > 0$.



(a) A DAG hierarchy where the siblings r_4 and r_5 have the same context set $\{c_5\}$.

(b) A new hierarchy by (1) removing hypernymy information of r_4 of the common parent (r_2) and (2) hyponymy information of r_5 of the common daughter (r_7).

Figure 5: If upward distributional generalization exists, then $\forall r_j \in \{r_5, r_6, r_7, r_8\}$: $s(r_4, r_2) > s(r_4, r_j)$; If downward exists, then $\forall r_j \in \{r_1, r_2, r_3, r_4, r_6\}$: $s(r_7, r_5) > s(r_j, r_5)$.

To test for upward (downward) distributional generalization, we construct a new hierarchy for this test. First, we sample five nouns from the H'_{DAG} hierarchy. Then, for each of these nouns $\tilde{r}$, we change the contexts set of it to that of one of its siblings $\tilde{r}'$, and remove the hypernymy (hyponymy) information of $\tilde{r}$ and $\tilde{r}'$'s common parent (daughter) from $\tilde{r}$ to create a new hierarchy. If distributional upward (downward) generalization exists in FDS models, we should expect the hypernyms (hyponyms) of $\tilde{r}'$ to be identified also as those of $\tilde{r}$'s after training on the new hierarchy. For instance, if upward generalization happens, we should expect (15) to be true, where $h(r)$ is the set

Model	Hypothesis	Upward	Downward
FDS	DIH	.906	.690
$\text{FDS}_\forall$	rDIH	.971	.993

Table 4: Mean AUC for distributional generalization.

of hypernyms of r in the original hierarchy H'_{DAG} and s (described in (10)) is the hypernymy scoring function trained on the new hierarchy. Fig. 5 illustrates the idea with an example.

$$\begin{aligned} &\forall r_i \in h(\tilde{r}') \setminus h(\tilde{r}): \\ &\forall r_j \notin h(\tilde{r}) \cup h(\tilde{r}') \cup \{\tilde{r}\}: \\ &\qquad s(\tilde{r}, r_i) > s(\tilde{r}, r_j) \end{aligned} \tag{15}$$

As the distributions of classes vary for each $\tilde{r}$, we measure the performance with mean AUC, averaged over the five chosen nouns $\tilde{r}$. AUC equals 1 if and only if (15) is true. Table 4 shows that both upward and downward distributional generalizations exist when the corpus follows either DIH or rDIH, and to a larger extent on the rDIH corpus.

5.5 Summary

The experiments confirm that: (1) hypernymy is learnt by FDS models under DIH, (2) the new training objective for universal quantifications enables FDS models to also learn hypernymy under rDIH, and (3) our approach of hypernymy modelling allows generalization over missing information in a corpus.

6 Experiments on Real Data Sets

Seeing how FDS performs on synthetic data sets does not immediately tell us more about hypernymy learning on real data sets. Therefore, we perform further experiments to test if FDS models learn hypernymy on open classes of sentences.

6.1 FDS Models Training

Training Data. FDS models are trained on Wikiwoods[1] (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 quantifiers are annotated in Wikiwoods, it is not feasible to determine which of the two training objectives to use specifically for each instance. This is because quantifications interact heavily with other semantic components in a complex sentence. For example, *every dog that is excited barks* requires universal quantification over the intersection of the set of dogs and the set of entities that are excited, but set intersection is not modelled by FDS. In our experiments, we choose one of the models from FDS or $\text{FDS}_\forall$ described in §5.2, and apply the same objective to every training instance. We also test an additional model $\text{FDS}_{\forall/2}$ where the universal quantification objective is scaled by 0.5. We only train each of the models for 1 epoch.

6.2 Evaluation Method

We only consider hypernymy over nouns but not verbs or adjectives since FDS is trained on DMRS graphs, where only nominals are quantified and accepted by verbs and adjectives as arguments. We test the trained models on four hypernymy data sets for nouns, namely LEDS (Baroni et al., 2012), Kotlerman 2010 (Kotlerman et al., 2010) and WBLESS (Weeds et al., 2014), EVALution (Santus et al., 2015). We removed the out-of-vocabulary instances during the evaluation. We report the AUC as in §5.

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

6.3 Baselines

Following Roller et al. (2018), we implemented four distributional hypernymy detection models and trained them on Wikiwoods as baselines. First, we have WeedsPrec (Weeds et al., 2004) and invCL (Lenci and Benotto, 2012), which measures context inclusion of word pairs. They both use a distributional space that is constructed by first counting co-occurrences of adjacent predicates in the preprocessed DMRS graphs, then the resulting matrix is transformed using positive pointwise mutual information. Each row vector $v^{(r_i)}$ represents

[1] http://ltr.uio.no/wikiwoods/1212/

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

a predicate r_i. WeedsPrec and invCL are computed as:

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

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

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

Apart from the two DIH measures, we also consider 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 $H(c_j)$ is the Shannon entropy of the associated context c_j. Then, SLQS compared the generality of two words by the ratio of their respective medians. We also report SLQS-cos, which multiplies the SLQS measure by cosine similarity of $v^{(r_1)}$ and $v^{(r_2)}$, since the SLQS measure only considers generality but not similarity. N is chosen to be 50 following Santus et al. (2014). We also include cosine similarity (Cosine) of the row vectors as an extra baseline.

6.4 Results

Model	LEDS	Evalution	Kotlerman2010	WBLESS
Cosine	.78	.53	.70	.62
WeedsPrec	.90	.65	.67	.71
InvCL	.90	.62	.68	.71
SLQS	.48	.53	.49	.57
SLQS-cos	.48	.53	.49	.56
FDS	.65	.46	.47	.51
FDS$_{\forall/2}$	.76	.60	.56	.66
FDS$_{\forall}$	.73	.55	.56	.66

Table 5: AUC on the test sets.

Table 5 shows the results on the four test data sets. The DIH baselines are competitive and nearly outperform all models across the test sets. FDS$_{\forall}$ and FDS$_{\forall/2}$ both outperform FDS considerably across the test sets. This reflects that including the proposed universal quantification 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	.51	.37	.68	.92
WeedsPrec	.75	.62	.63	.84
OnvC	.75	.57	.65	.87
SLQS	.61	.55	.59	.52
SLQS-cos	.58	.53	.57	.55
FDS	.60	.29	.56	.58
FDS$_{\forall/2}$	.78	.59	.56	.69
FDS$_{\forall}$	.79	.63	.52	.70

Table 6: AUC on the sub-categories of WBLESS.

Table 6 shows the results on the WBLESS sub-categories. It is shown that 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. FDS$_{\forall}$ and FDS$_{\forall/2}$ outperform FDS across nearly all sub-categories, with much higher distinguishing power for co-hyponymy and hypernymy. These imply that the universal quantification objective makes FDS more sensitive to the relative generality than the similarity of word pairs.

7 Conclusion

We discussed how Functional Distributional Semantics (FDS) (1) can represent hypernymy, (2) is related to the Distributional Inclusion Hypothesis, and (3) can handle simple universal quantification. The experiments on synthetic data sets confirm that FDS learns hypernymy when a corpus strictly follows the Distributional Inclusion Hypothesis (DIH), and the proposed objective is effective for the reverse of the hypothesis. The experiments on real data sets show that the proposed new training objective for FDS improves hypernymy detection from corpus. We hope that this work provides insights into hypernymy learning from corpora by FDS models.

Limitations

Hypernymy is established between word senses. However, the proposed representation of hypernymy in FDS compares the semantic functions of DMRS predicate pairs and considers each DMRS predicate to have one sense. Therefore, such representation can fall short of polysemous words.

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.

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.

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*.

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.

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 Derivations

Consider (9). $\forall z \in \mathbb{R}^d$:

$$t^{(r_H,0)}(z) > t^{(r_h,0)}(z)$$

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

S is monotnoic, so $\forall z \in \mathbb{R}^d$:

$${v^{(r_H,0)}}^\top z + b^{(r_H,0)} > {v^{(r_h,0)}}^\top z + b^{(r_h,0)}$$

$$b^{(r_H,0)} - b^{(r_h,0)} > (v^{(r_h,0)} - v^{(r_H,0)})^\top z$$

On a unit d-cube, this is equivalent to

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

Note that $\arg\min_{z_i} \left((v^{(r_h,0)} - v^{(r_H,0)})^\top z \right) = sgn(v_i^{(r_H,0)} - v_i^{(r_h,0)})$ where sgn is the sign function. Hence, we have

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

On a unit d-sphere, this is equivalent to

$$b^{(r_H,0)} - b^{(r_h,0)} > \min_{\|z\|_2=1} \left((v^{(r_h,0)} - v^{(r_H,0)})^\top z \right) \tag{16}$$

By Cauchy–Schwarz inequality, we have

$$\begin{aligned} \left(v^{(r_h,0)} - v^{(r_H,0)} \right)^\top z &\leq \left\| v^{(r_H,0)} - v^{(r_h,0)} \right\|_2 \|z\|_2 \\ &= \left\| v^{(r_H,0)} - v^{(r_h,0)} \right\|_2 \end{aligned}$$

Hence, (16) is true if

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

B Training Details

B.1 Hyperparameters and Tuning

For all the experiments, the hyperparameters of the FDS models largely follow that of FDSAS$_{\text{id}}$ in Lo et al. (2023) except that we set β_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 that pixies are only meaningful within the unit d-sphere or d-cube. Some minor changes to the hyperparameters are made for the synthetic and real data sets respectively.

Here are the changes exclusive to the experiments on the synthetic data sets. We set K to 1 and the learning rate to 0.01. For experiments on H_{chains}, d is set to 2. For H_{WN}, d is set to 40. For the remaining topologies, d is set to 10. The models are trained for 125 epochs for H_{WN}, and 5000 epochs for the rest.

B.2 Computational Configurations

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