Functional Distributional Semantics at Scale

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

Abstract

Functional Distributional Semantics is a linguistically motivated framework for modelling lexical and sentence-level semantics with truth-conditional functions using distributional information. Previous implementations of the framework focus on subject–verb–object (SVO) triples only, which largely limits the contextual information available for training and thus the capability of the learnt model. In this paper, we discuss the challenges of extending the previous architectures to training on arbitrary sentences. We address the challenges by proposing a more expressive lexical model that works over a continuous semantic space. This improves the flexibility and computational efficiency of the model, as well as its compatibility with present-day machine-learning frameworks. Our proposal allows the model to be applied to a wider range of semantic tasks, and improved performances are demonstrated from experimental results.

1 Introduction

Functional Distributional Semantics (FDS; Emerson and Copestake, 2016; Emerson, 2018) aims to capture the truth-conditional aspects of words through learning from distributional information of a corpus. Whilst truth-conditional semantics deals with predications over discrete entities, FDS aims to generalize about predications over a space of *entity representations* with probabilistic semantics.

Contrasted with most distributional methods which map words to vectors, FDS can model various aspects of meaning in a linguistically rigorous manner. For example, vagueness is represented by the probabilistic nature of predications, and hypernymy, defined formally as the subsumption of the extensions between two word senses, can be represented by the subsumption of regions of space (Emerson, 2020b).

Going beyond simple vector spaces, some models of distributional semantics represent words as tensors for composition (e.g., Coecke et al., 2010; Baroni et al., 2014), as static distributions for uncertainty and entailment (e.g., Vilnis and McCallum, 2015), as posterior distributions for context-specific meaning (e.g., Bražinskas et al., 2018), and as regions for set-theoretic properties (e.g., Dasgupta et al., 2022). Among them, only a region-based approach favours logical interpretations (for a discussion, see: Emerson, 2020b, 2023).

In order to be computationally tractable, most models of distributional semantics are trained based on instances defined by context windows (e.g., Mikolov et al., 2013a; Pennington et al., 2014) or incomplete linguistic structures such as immediate dependencies (e.g., Levy and Goldberg, 2014; Czarnowska et al., 2019). All previous instances of FDS (further discussed in §2) are only trained on SVO triples. Consequently, these models underutilize much contextual information.

We hope to extend FDS learning to arbitrary sentences, but not larger linguistic units (e.g., paragraphs), for handling them requires non-trivial extensions such as robust coreference resolution, which is beyond the scope of this work. To this end, we propose to adopt a continuous semantic space and a more expressive lexical model in place of the previous world model on a discrete space. Our new formulation provides a computationally efficient and linguistically principled solution to applying FDS to arbitrary sentences. Furthermore, this also situates the framework closer to modern machine learning models which are mostly built upon continuous latent spaces, thus favouring comparisons among and integration with them. For example, Liu and Emerson (2022) integrated a pre-trained computer vision model with a continuous space to FDS, applying it to annotated images. Joint learning of the visually-grounded and corpus-based models was however left as future work due to the incompatibility of latent spaces.

In this paper, we first give an introduction to FDS

in §2, explaining why it is difficult for previous implementations to scale up. Then, we present in detail the proposed formulation and how to train the model in §3–§4. Finally, we demonstrate how our model can be applied to a number of semantic evaluation data sets and present the results in §5.

2 Functional Distributional Semantics

The core idea of Functional Distributional Semantics is that a sentence refers to a set of entities, and a word is a predicate that is true or false of entities. Compared to other approaches to distributional semantics, it aligns more with model-theoretic semantics, which approaches meaning in the same way in terms of a *model structure*.

However, fixing a specific set of entities would make it impossible to generalize to new situations. In order for the model to be learnable, predicates do not directly take entities as input, but rather entity representations, referred to as *pixies* for brevity. A predicate is represented as a function from pixies to probabilities of truth. This allows the model to account for vagueness.

FDS does not submit to a fixed interpretation of pixies nor process of obtaining them. Rather, pixies are introduced to merely convey information of latent entities. In the work of Liu and Emerson (2022), pixies are dimensionality-reduced vectors obtained from a pretrained network. In this work, they are learnt to best represent entities according to our particular formulation by probabilistic graphical models, which are introduced below and in detail in §4.

2.1 Probabilistic Graphical Models

The framework is formalized in terms of a family of probabilistic graphical models, each of which generates predicates in a semantic graph. It consists of the *world model*, which handles the joint distribution of pixies, and the *lexical model*, which handles truth-conditional semantics. Given an *argument structure* (predicate–argument structure minus predicates, i.e., a directed graph with labelled edges and unlabelled nodes), a predicate can be generated for each node, in three steps. First, a pixie is generated for each node, which together represent the entities to be described. Then, a truth value in $\{\top, \bot\}$ is generated for each entity and each predicate in the vocabulary $\mathcal{V}$. Finally, a single predicate is generated for each entity. This is shown in Fig. 1, for the simple predicate–argument

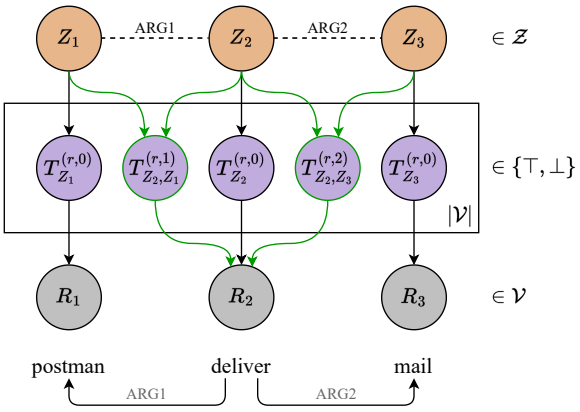

Figure 1: Probabilistic graphical model of FDS for generating words in an SVO triple (e.g., '*postman deliver mail*'). The Z nodes are pixie-valued random variables; T nodes are truth-valued; R nodes are predicate-valued. Only the R nodes are observed (e.g., R_1=*postman*, R_2=*deliver*, R_3=*mail*). This figure contrasts two positions where argument information can be used. In previous work, argument information (i.e., ARG1 and ARG2) only contributes to the world model (in dashed lines). In our formulation, it only contributes to the lexical model (in green lines).

structure of an SVO triple. Different argument structures have different graphical models.

In previous work, $\mathcal{Z}$ is sparse binary-valued vectors, and the joint distribution of pixies is determined by a Cardinality Restricted Boltzmann Machine (CaRBM) using the argument structure. The lexical model comprises unary semantic functions, each of which maps one pixie to the probability that the predicate is true of the pixie.

In §3, we will propose to move the information about the predicate–argument structure from the world model to the lexical model and set $\mathcal{Z} = \mathbb{R}^d$. Concretely, the dependencies among pixies are removed and extra truth-valued random variables $T^{(r,a)}_{Z_i,Z_j}$ are added (also shown in Fig. 1).

2.2 Model Learning from DMRS

The model is trained on graphs of Dependency Minimal Recursion Semantics (DMRS; Copestake et al., 2005; Copestake, 2009). A DMRS graph is derived using the broad-coverage English Resource Grammar (ERG; Flickinger, 2000, 2011), providing a compact representation of the predications expressed by a sentence. Figs. 1 and 2 show three simplified DMRS graphs (with quantifiers and scope removed). Model parameters are optimized in an unsupervised manner to maximize the likelihood of generating the observed predicates

given the argument structure of a DMRS graph.

In principle, the formalism of semantic graphs for learning is not restricted to DMRS, but any that include predicate–argument structures. Bender et al. (2015) argued that deriving semantic graphs compositionally and automatically using a broad-coverage grammar is more scalable and consistent than manual annotation, as is common for other formalisms such as Abstract Meaning Representation (AMR; Banarescu et al., 2013).

In §2.3–§2.4, we discuss the linguistic and computational challenges of training previous FDS models on more complex sentences.

2.3 Linguistic Challenges

Vocabulary. Addressing SVO triples only requires training and testing on nouns and verbs. With arbitrary sentences, the vocabulary of predicates expands to (1) adjectives, adverbs and adpositions, which are also predicates, (2) conjunctions, which not only contribute to extensional logic operations (e.g., *and*, *or* and *else*) but also intensional, modal or temporal ones (e.g., *until*, *if* and *since*), and (3) quantifiers. In addition, scope-taking predicates like quantifiers and conjunctions are barely meaningful when the scopes of them are underspecified. Therefore, it is not straightforward to apply the framework to arbitrary sentences without further linguistic assumptions.

Overloaded Argument Roles. The world model with CaRBM uses shared weights for argument roles of different predicates. However, argument roles are overloaded in DMRS. For example, ARG1 of the inchoative predicate `_break_v_1` and causative `_break_v_cause` specify what is broken and what breaks something, respectively. Consequently, predicate-specific thematic interpretations of argument roles are missed out. Argument roles also vary across different parts of speech: the ARG1 of nouns mostly denotes their prepositional complements, that of verbs denotes the agent, and that of an adjective denotes the element to be modified. Dealing with a larger vocabulary of predicates magnifies the problem with the coarse generalization by the undirected graphical models.

2.4 Computational Challenges

With Discrete Pixie Space. Training the model requires computing the likelihood of the observed data, thus the prior of the latent variables. However, it is intractable to compute the probability of a set of pixies in the discrete CaRBM because it requires normalizing over all possible sets of pixie values. Emerson (2020a) approximated the probability using belief propagation methods (Yedidia et al., 2003), which is still computationally expensive. This problem only gets worse when considering larger semantic graphs.

With Continuous Pixie Space. Switching to more tractable continuous distributions makes normalization easier. Nevertheless, the problem is still not simple. Fabiani (2022) explored the use of a continuous space, using a Gaussian Markov Random Field for the world model, and parameterizing the inverse covariance matrix according to the argument roles. Such a matrix has a size of $nd \times nd$ for a DMRS graph with n predicates with pixie dimension d. The complexity of computing its determinant scales to $\mathcal{O}(d^3n^3)$, which is feasible for simple graphs such as SVO triples but computationally prohibitive for larger graphs.

3 Enriching the Lexical Model

In this section, we describe our enriched lexical model and explain how it provides a solution to the linguistic and computational challenges mentioned.

3.1 Neo-Davidsonian Event Semantics

We follow Neo-Davidsonian event semantics (Davidson, 1967; Parsons, 1990) as with previous work, assuming that verbal and adjectival predicates refer to events. For example, to evaluate the claim that 'x *eats* y', we decompose it into three claims: e is an eating event, the ARG1 of this eating event is x, and the ARG2 of this eating event is y.

The event argument naturally allows FDS to be applied to not just nouns and verbs but arbitrary sentences with various types of modifications. For example, for x *eats* y *very quickly*, we have $\text{eat}(e_1, x, y) \wedge \text{quick}(e_2, e_1) \wedge \text{very}(e_3, e_2)$.

3.2 Semantic Functions

As mentioned in §2.1, we introduce truth-valued random variables for argument roles. The probability of truth is determined by either a unary function, as in (1), or a binary function, as in (2), over continuous-valued pixies.

$$P\left(T_{Z_e}^{(r,0)}=\top \,\middle|\, z_e\right) = t^{(r,0)}(z_e) \quad (1)$$

$$P\left(T_{Z_e,Z_x}^{(r,a)}=\top \,\middle|\, z_e, z_x\right) = t^{(r,a)}(z_e, z_x) \quad (2)$$

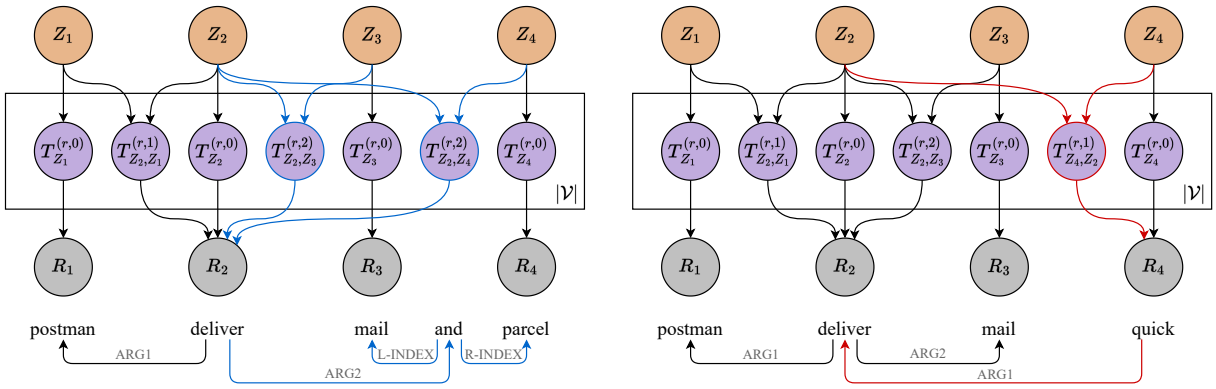

Figure 2: Probabilistic graphical models which can generate the two sentences '*A postman delivers mail and parcels*' (left) and '*A postman delivers mail quickly*' (right) respectively, illustrating how the example in Fig. 1 can be extended with a coordinating conjunction and an adverb. Blue and red lines show the correspondence between dependencies in the graphical models and DMRS argument structures.

We may interpret $t^{(r,0)}(z_e)$ as the probability that r is true of the entity e (represented by z_e) and $t^{(r,a)}(z_e, z_x)$ as the probability that the a-th argument role of r holds between e and x (represented by z_e and z_x). For example, given a predicate r that takes two arguments (e.g., the transitive '*eat*'), the probability of the predication being true is:

$$P\left(T^{(r,0)}_{Z_e} \wedge T^{(r,1)}_{Z_e,Z_x} \wedge T^{(r,2)}_{Z_e,Z_y} = \top \,\middle|\, z_e, z_x, z_y\right) = t^{(r,0)}(z_e)\, t^{(r,1)}(z_e, z_x)\, t^{(r,2)}(z_e, z_y) \quad (3)$$

In the same spirit as Paperno et al. (2014)'s proposal, this decomposition of arity-dependent predicates allows dropped arguments to be handled naturally. For the example '*y is eaten*', we have:

$$P\left(T^{(r,0)}_{Z_e} \wedge T^{(r,2)}_{Z_e,Z_y} = \top \,\middle|\, z_e, z_y\right) = t^{(r,0)}(z_e)\, t^{(r,2)}(z_e, z_y) \quad (4)$$

3.3 Addressing the Challenges

Lexical Model beyond Nouns and Verbs. In our lexical model, nouns, verbs, adjectives, and adverbs all introduce truth-valued random variables but not adpositions, whose uses are considered too flexible to be modelled by our implementation (discussed in §4.3). Proper nouns that mostly denote distinct entities are discarded and arguments that take proper nouns are dropped, as it results in an unreasonably large vocabulary otherwise. We also discard pronouns which require coreferences. Argument roles are propagated through coordinating conjunctions: if a predicate takes a coordinating conjunction as an argument, the argument role is applied to each conjunct. We also neglect quantifiers and modal verbs. Fig. 2 illustrates how the example in Fig. 1 can be extended with additional truth-valued random variables to handle coordinating conjunctions and adverbs. The proposed lexical model thus addresses the vocabulary challenge and also provides a workaround to the problem with overgeneralization of arguments in §2.3.

Computational Efficiency. The information of the predicate–argument structure, which was previously encoded in the world model via dependencies between pixies, is now embedded in the design of the semantic functions. As discussed in §2.4, the main computational challenge in FDS is normalizing joint distributions for sets of pixies. In contrast, the computational cost of binary semantic functions can be kept essentially the same as for unary functions, as discussed further in §4.3. By offloading the complexity from the world model to the lexical model, we can use a simple prior distribution that is trivially normalized, as discussed further in §4.5.

Summary. As compared to previous implementations, our proposal makes FDS more scalable by covering a much broader class of predicates, drastically reducing the computational complexity, and providing a more appropriate treatment of predicate-specific argument roles for richer sentence structures.

4 Variational Autoencoder

As mentioned in §2.2, each training instance is a DMRS graph, which can be characterised in terms of n predicates $R = \{r_1, \ldots, r_n\}$, and the argument structure $A = \left\{(i, j, a) : r_i \xrightarrow{\text{ARG}a} r_j\right\}$.

To optimize the parameters θ of the generative

model, we use a variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014). The intractable true posterior distributions $p_\theta(z \mid R, A)$ over the pixies $z = \{z_1, \ldots, z_n\}$ are first approximated by tractable distributions chosen a priori (discussed in §4.1). Instead of directly performing maximum likelihood estimation on the observed DMRS graphs, the lower bound in (5) is maximized following the β-VAE (Higgins et al., 2017), using a probabilistic encoder q_ϕ (discussed in §4.2) and decoder p_θ (discussed in §4.3). §4.4 and §4.5 reformulate the two terms in (5) respectively based on empirical insights for training stability. Parameters of the encoder and decoder are thus jointly learnt via gradient descent.

$$\begin{aligned}\mathcal{L}_{\phi,\theta}(R \mid A) = \mathbb{E}_{q_\phi(z|R,A)}\left[\ln P_\theta(R \mid z, A)\right] \\ - \beta D_{\mathrm{KL}}\left(q_\phi(z \mid R, A) \parallel p_\theta(z \mid A)\right)\end{aligned} \tag{5}$$

4.1 Approximate Posterior Distributions

Given an observed DMRS graph with n latent pixies Z_i, the approximate posterior is partitioned into n independent Gaussians with spherical covariance. Gaussian distributions provide convenient closed forms for analytical computation. For instance, sampling of pixies can be avoided in §4.3. For each Z_i, the encoder q_ϕ predicts a mean vector μ_{Z_i} and a variance $\sigma^2_{Z_i}$. This gives the distribution in (6), where $\mathcal{N}$ is the Gaussian density function.

$$q_\phi(z \mid R, A) = \prod_{i=1}^{n} \mathcal{N}(z_i; \mu_{Z_i}, \sigma^2_{Z_i} I) \tag{6}$$

4.2 Amortized Variational Inference

We devise an encoder that uses both the local predicate–argument structure and global topical information from the whole sentence. For example, the encoder should predict different pixie distributions for '*deliver*' in the contexts of Fig. 2 (delivering mail) and Fig. 3 (delivering a song). The encoder architecture is described by (7), (8), (9) and illustrated in Fig. 3. It is similar to the encoder of Bražinskas et al. (2018), but leverages argument structure. It can also be seen as a simple instantiation of Deep Sets (Zaheer et al., 2017) or a graph-convolutional network (GCN) with complement edges (De Cao et al., 2019). The mean μ_{Z_i} and log variance $\ln \sigma^2_{Z_i}$ are inferred based on a hidden layer $h^{(Z_i)}$, where the logarithm ensures a positive variance. The input embeddings $e^{(r,a)}$ represent predicates standing in particular relation to the target predicate, as detailed in Fig. 3. f can be

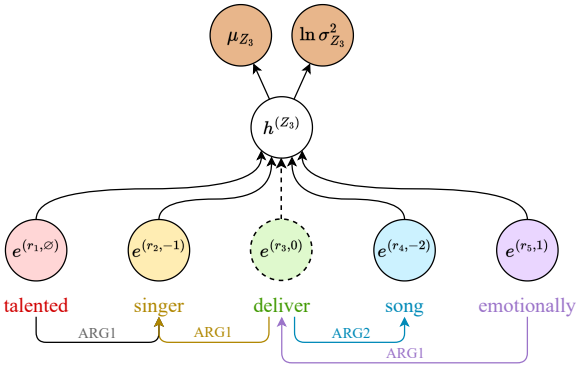

Figure 3: An encoder for inferring the posterior distribution of the pixie of *deliver* in the sentence *talented singer deliver song emotionally*. The inputs represent context predicates standing in particular relation to the target predicate. The embedding $e^{(r,a)}$ represents the predicate r with relation a, where negative a indicates an argument role of the target predicate, positive a an argument role of a context predicate, 0 the target predicate itself, and $\varnothing$ the absence of a direct argument role. The embedding with dropout is shown in dashed lines.

the identity function or a non-linear function, e.g., the hyperbolic tangent. We perform experiments on both choices.

$$h^{(Z_i)} = f\left(\frac{1}{n}\sum_{j=1}^{n} e^{(r_j,a_{j,i})}\right) \tag{7}$$

$$\mu_{Z_i} = W^\top h^{(Z_i)} + c_1 \tag{8}$$

$$\ln \sigma^2_{Z_i} = w^\top h^{(Z_i)} + c_2 \tag{9}$$

During VAE training, the parameters of $t^{(r_i,0)}$ and $e^{(r_i,0)}$ will be optimized to maximize $t^{(r_i,0)}(z_i)$. There is a chance that the distributions of pixies are inferred purely from the embedding of intrinsic arguments and the remaining embeddings are trivially optimized to very small values. To prevent such a learning shortcut, we apply dropout to the embeddings $e^{(r_i,0)}$ with a certain probability where $h^{(Z_i)}$ aggregates without it.

In contrast to our work, Emerson (2020a) used a two-layer GCN as the encoder. Scaling a GCN to larger graphs requires a deeper network to incorporate long-distance, yet crucial topical information. However, a deeper network is computationally expensive and hard to train. We believe that it is worthwhile to start with a simpler and more efficient architecture for our new formulation.

4.3 Probabilistic Decoder

The generative model can be seen as a probabilistic decoder. It consists of the unary and binary

semantic functions of predicates. The functions are implemented as linear classifiers in (10) and (11), where S denotes the sigmoid function and $z_{i,j}$ denotes the concatenation of z_i and z_j.

$$t^{(r_i,0)}(z_i) = S\left({v^{(r_i,0)}}^\top z_i + b^{(r,0)}\right) \quad (10)$$

$$t^{(r_i,a)}(z_i, z_j) = S\left({v^{(r_i,a)}}^\top z_{i,j} + b^{(r_i,a)}\right) \quad (11)$$

Linear classifiers provide a number of advantages over complex ones, albeit less expressive. First, they are computationally less expensive. Second, the frequency of word occurrence in a corpus has a long tail, so there are inadequate instances for training more powerful classifiers for the rare predicates. Last but not least, since the pixies are normally distributed given the observation as defined in §4.2, we may use the probit approximation (Murphy, 2012, §8.4.4.2) for computing the expectation of (1) and (2) over the approximate posterior. (12) shows such approximation for the unary semantic function.[1] Computing the first term in (5) otherwise requires sampling, which is more computationally expensive and can result in poor estimations when the variance is high.

$$\mathbb{E}_{q_\phi}\left[t^{(r,0)}(z_i)\right] \approx S\left(\frac{{v^{(r,0)}}^\top \mu_{Z_i} + b^{(r,0)}}{(1 + \frac{\pi}{8}\sigma^2_{Z_i})^{\frac{1}{2}}}\right) \quad (12)$$

4.4 Contrastive Objective on Truth

The first term of (5) requires computing the probability of generating the observed predicates R given the distributions of pixies z and the argument structure A. In previous work, such a probability is set to be proportional to the probabilities of truth of the predications. Consequently, training on this objective only considers the relative probabilities of truth but not absolute probabilities. Truth regularization was introduced to increase the absolute probabilities for better interpretability (Emerson, 2020a). However, both improved and deteriorated model performances were reported by Liu and Emerson (2022) with such regularization. Moreover, we find from experiments that training using the original objective is unstable and requires careful tuning of the regularization coefficient, which furthermore is sensitive to the value of β.

Instead of maximizing the relative probabilities, we propose a contrastive objective on absolute probabilities of truth: we aim to maximize the truth of the observed predicate and the falsehood of negatively sampled predicates, analogous to Skip-gram negative sampling (Mikolov et al., 2013b).

The objective is given in (13) and (14), for unary and binary semantic functions respectively. Each term $\mathcal{C}_i$ or $\mathcal{C}_{i,j,a}$ corresponds to a truth value node in Fig. 1 and 2, and $N(i)$ denotes the negative samples for the predicate r_i.

$$\mathcal{C}_i = \ln \mathbb{E}_{q_\phi}\left[t^{(r_i,0)}(z_i)\right] + \sum_{r' \in N(i)} \ln \mathbb{E}_{q_\phi}\left[1 - t^{(r',0)}(z_i)\right] \quad (13)$$

$$\mathcal{C}_{i,j,a} = \ln \mathbb{E}_{q_\phi}\left[t^{(r_i,a)}(z_i, z_j)\right] + \sum_{r' \in N(i)} \ln \mathbb{E}_{q_\phi}\left[1 - t^{(r',a)}(z_i, z_j)\right] \quad (14)$$

Underlying this objective is the assertion that randomly drawn predicates are usually false of the inferred pixies. This objective departs from the generative model in §2.2 and directly operates on probabilities of truth instead of generation probabilities. The proposed objective achieves a very similar goal as the original one, i.e., to maximize the probabilities of truth of the observed predicates and minimizes those of the unobserved, while truth regularization is unnecessary and changes in β do not lead to instability.

For each observed predicate, we draw K samples from the unigram distribution. However, we restrict the distribution to predicates that are compatible with the observed argument roles. Each predicate has a set of possible argument roles (those that appear somewhere in the training data). We restrict to predicates whose possible argument roles are a superset of the observed roles.

4.5 Alternative Variance Regularization

Since we have removed the dependencies among pixies and we have no prior knowledge about the latent space, the KL term in (5) is not informative. In fact, we empirically find that it can even be harmful: (1) adopting a standard normal prior with $\beta > 0$ always yields worse performance on the development set (discussed in §5.2) than when $\beta = 0$, and (2) when $\beta = 0$, the inferred variance occasionally takes very large values when f is the identity function, rendering inference uninformative.

We devise an alternative regularization term (15) that replaces the KL divergence in (5), where d is the dimensionality. This term is derived from the

[1] For brevity, we use $\mathbb{E}_{q_\phi}$ to denote $\mathbb{E}_{q_\phi(z|R,S)}$ hereafter.

KL divergence of q_ϕ from a standard normal distribution, which pulls variances to one but neglects the means. This way, the variance is still regularized to avoid extreme values, while not imposing a strong belief about the expected locations of pixies.

$$\mathcal{D} = \frac{d}{2} \sum_{i=1}^{n} \left(\sigma_{Z_i}^2 - \ln \sigma_{Z_i}^2 \right) \quad (15)$$

For each instance, the final training objective to maximize is reformulated to (16).

$$\tilde{\mathcal{L}}_{\phi,\theta}(R \mid A) = \sum_{i=1}^{n} \mathcal{C}_i + \sum_{(i,j,a)\in A} \mathcal{C}_{i,j,a} - \beta\mathcal{D} \quad (16)$$

5 Experiments

Evaluating a semantic model is not an easy task. We focus on tasks that involve semantic composition and contextualized meaning. In particular, we select RELPRON (Rimell et al., 2016) and GS2011 (Grefenstette and Sadrzadeh, 2011) (and GS2013 (Grefenstette and Sadrzadeh, 2015), a re-annotated version of GS2011), the two data sets evaluated by Emerson (2020a). This allows a direct comparison between our approaches. In addition, our proposed approach formally incorporates adjectives, which gives us the opportunity to evaluate on GS2012 (Grefenstette, 2013). Our implementation is available online.[2]

5.1 Training Data

The data we train on is DMRS graphs extracted from Wikiwoods[3] (Flickinger et al., 2010; Solberg, 2012) using Pydelphin[4] (Copestake et al., 2016). Wikiwoods provides linguistic analyses of 55m sentences (900m tokens) in English Wikipedia. Each sentence 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). The preprocessed data consists of DMRS graphs of 36m sentences, where 254m tokens are involved in training (preprocessing details described in §A.1). We preprocess the evaluation data into DMRS graphs following ERG analyses.

5.2 Model Configurations

We test for two model configurations: $\text{FDSAS}_{\text{tanh}}$ and FDSAS_{id}. They differ in activation functions (discussed in §4.2). Each of them comprises 54m parameters. All other hyperparameters are simply fixed (reported in §A.2). Since only RELPRON provides a development set but not GS2011, GS2013, or GS2012, each of our models is tuned on the development set of RELPRON (described in §A.2) and have their outputs averaged over three random seeds. For fair comparisons, we only report results of previous works that train their models on a corpus in an unsupervised manner. We select the best result from each of their models.

Model	MAP	
	Dev.	Test
Vector addition (add.) (Rimell et al., 2016)	0.496	0.472
Sim. Practical Lexical Function (Rimell et al., 2016)	0.496	0.497
Vector add. (Czarnowska et al., 2019)	0.485	0.475
Dependency vector add. (Czarnowska et al., 2019)	0.497	0.439
Pixie Autoencoder (PixieAE) (Emerson, 2020a)	0.261	0.189
Ensemble of PixieAE & vector add. (Emerson, 2020a)	0.532	0.489
$\text{BERT}_{\text{BASE}}$ (tuned template with full stop)	0.677	0.667
$\text{BERT}_{\text{BASE}}$ (tuned template without full stop)	0.302	0.200
$\text{FDSAS}_{\text{tanh}}$	0.486	0.477
FDSAS_{id}	0.657	0.580

Table 1: Results on RELPRON.

5.3 Evaluation on Semantic Composition

RELPRON is a data set of subject and object relative clauses. It consists of terms (e.g., '*telescope*'), paired with up to 10 corresponding properties (e.g., '*device that astronomer use*'). Each property comes in lemmatized words. The development set contains 65 terms and 518 properties and the test set contains 73 terms and 569 properties. The task is to rank all properties for each term so that the correct ones come before the incorrect ones. Performance is measured using Mean Average Precision (MAP).

5.3.1 Using FDS

Following Emerson (2020a), for each property, the encoder is used to compose from the relative clause and infer the pixie distribution of the target subject or object. Then, for each term, we rank the properties by the log of the expected probability that the term is true of the target pixie. This is obtained by applying the semantic function of the term to the inferred pixie distribution using (12).

5.3.2 Results

As a baseline, we adopt $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), a language model with 110m parameters, using the Transformers library (Wolf et al., 2019). It performs masked prediction on a cloze sentence, e.g., '*[CLS] a device that an astronomer uses is a*

[2] https://github.com/aaronlolo326/TCSfromDMRS
[3] http://ltr.uio.no/wikiwoods/1212/
[4] https://github.com/delph-in/pydelphin

[MASK]. [SEP]'. As RELPRON properties are lemmatized and contain no articles, they must be converted into cloze sentences using a template. Experimenting with different cloze templates, the best one on the development set uses singular nouns, the article *a/an*, an inflected verb (using Pattern (Smedt and Daelemans, 2012)), and a full stop.

Table 1 shows the results on RELPRON. Our best model outperforms all existing work, except the $\text{BERT}_{\text{BASE}}$ baseline. Nevertheless, it is important to note that $\text{BERT}_{\text{BASE}}$ has twice as many parameters and is trained on ten times more tokens compared to each of our models. As mentioned by Emerson (2020a), vector space models are good at capturing topical relatedness, whereas the PixieAE uses FDS and learns different information. Our large improvement over Emerson's ensemble model suggests that our formulation manages to combine the best of both worlds.

The $\text{BERT}_{\text{BASE}}$ baseline achieves a new state of the art. Nevertheless, our experiments show BERT's sensitivity to the template. While Emerson (2020a) discussed template tuning for BERT, they did not mention punctuation, which we find to be crucial for high performance. Aligning with Kementchedjhieva et al. (2021)'s observation, we found that BERT often generates a full stop with over 90% probability when the template does not end with one, although the [SEP] token already indicates the end of a sentence. This shows that ending the sequence with a full stop is more important to BERT than grammaticality. Performance is also degraded if either of the [CLS] or [SEP] tokens are missing. In contrast, FDS models operate on DMRS, abstracting over punctuation and inflection, and extra tuning of templates is unnecessary.

Rimell et al. (2016) also designed RELPRON to have *confounders*, non-corresponding terms and properties with lexical overlap, e.g., '*soil*' with '*activity that soil support*' (which corresponds to '*farming*') and '*fuel*' with '*phenomenon that require fuel*' (which corresponds to '*propulsion*'). There are 33 confounders in the test set and Emerson (2020a) reported that a vector addition model incorrectly ranked all the confounding properties in the top 4 for the overlapping term. In contrast, $\text{FDSAS}_{\text{tanh}}$, FDSAS_{id} and $\text{BERT}_{\text{BASE}}$ rank them 65st, 70th and 70th on average respectively.

5.4 Evaluation on Verb Disambiguation

GS2011 tests if a model is able to disambiguate ambiguous transitive verbs given the context of a subject and an object noun. It comprises 199 entries and 2,500 judgements by 25 annotators. Each entry of the data set provides an SVO triple (e.g., '*service meet need*') from the British National Corpus (BNC) and a transitive landmark verb (e.g., '*visit*' and '*satisfy*') from WordNet (Miller, 1995). Using a score from 1 to 7, the annotators rate the semantic similarity of the verb pair when each of the verbs takes the given subject and object. We also report the results on GS2013, the re-annotated version of GS2011 with a total of 9,950 judgements, where each pair is annotated by 50 annotators.

GS2012 also tests for verb disambiguation. It additionally includes an adjective for both the subject and object in the entries of GS2011 (e.g., '*social service meet educational need*'). It comprises 194 entries and 9,700 judgements by 50 annotators. A good model is expected to utilize the adjectives for better contextualization.

For each of these data sets, we measure the correlation of models' predictions with either separate or averaged annotators' judgements using Spearman's ρ. We compute the inter-annotator agreements (IAAs) by averaging the Spearman's ρ of each annotator's judgement against the other annotators'. IAA is believed to provide the theoretical maximum value for any model's performance.

5.4.1 Using FDS

We follow Emerson (2020a) that a score between a verb pair is the log of the expected probability that the landmark verb is true of the other verb pixie.

5.4.2 Results

We adopt $\text{BERT}_{\text{BASE}}$ as a baseline using the best template tuned on the development set of RELPRON. Tables 2, 3 and 4 show the results.

Care must be taken when comparing the face values of correlations for two reasons. First, models are trained on data of different sizes and sources. Hashimoto and Tsuruoka (2015) mentioned that their models trained on 1.9m sentences of BNC yield comparable results to those trained on 33m sentences from Wikipedia, which might be due to GS2011 being produced based on BNC. Training on a different corpus (e.g., Wikipedia) can better reflect how well a model generalizes. Hashimoto and Tsuruoka (2016) showed that models trained on

Model	Training Data			ρ	
	Sources	#Sentence (m)	#Token (m)	Separate	Averaged
Kartsaklis and Sadrzadeh (2013); Grefenstette (2013); Van de Cruys et al. (2013); Polajnar et al. (2015); Fried et al. (2015); Tian et al. (2016); Emerson and Copestake (2017)				< 0.4	< 0.5
Hashimoto et al. (2014)	B	6	-	0.41	0.50
Hashimoto and Tsuruoka (2015)	W	80 [33]	-	-	0.614
Hashimoto and Tsuruoka (2016) (Ensemble)	W+B	86 [35]	-	0.524	0.680
Gupta et al. (2015)	W+B+U	-	-	0.406	-
Gamallo (2019)	W+B	-	2,500	0.46	-
Wijnholds et al. (2020)	U	130	3,200	-	0.54
Emerson (2020a) (PixieAE)	W	55 [31]	900 [72]	0.406	0.504
$\text{BERT}_{\text{BASE}}$	W+O	-	3,300	0.394	0.519
$\text{FDSAS}_{\text{tanh}}$	W	55 [36]	900 [254]	0.438	0.553
FDSAS_{id}	W	55 [36]	900 [254]	0.444	0.552
Inter-annotator agreement				0.578	0.739

Table 2: Results on GS2011. Sources: W: Wikipedia, B: BNC, U: ukWaC (Baroni et al., 2009), O: BookCorpus (Zhu et al., 2015). Numbers of sentences and tokens are for raw data. In brackets are numbers after preprocessing; for our models, we report the number of tokens contributing to semantic functions. '-' means not reported.

Model	Training Data		ρ	
	#Snt. (m)	#Token (m)	Sep.	Avg.
Grefenstette and Sadrzadeh (2015)	-	-	0.26	-
Tilk et al. (2016)	138	-	0.34	-
Hong et al. (2018)	-	2,000	0.367	-
$\text{BERT}_{\text{BASE}}$	-	3,300	0.426	0.562
$\text{FDSAS}_{\text{tanh}}$	55 [36]	900 [254]	0.439	0.573
FDSAS_{id}	55 [36]	900 [254]	0.457	0.601
Inter-annotator agreement			0.587	0.777

Table 3: Results on GS2013.

Model	Training Data		ρ	
	#Snt. (m)	#Token (m)	Sep.	Avg.
Grefenstette and Sadrzadeh (2015)	-	-	0.27	-
Tian et al. (2016)	-	-	0.33	-
Gupta et al. (2015)	-	-	0.357	-
Paperno et al. (2014)	-	2,800	0.36	-
$\text{BERT}_{\text{BASE}}$	-	3,300	0.404	0.608
$\text{FDSAS}_{\text{tanh}}$	55 [36]	900 [254]	0.444	0.655
FDSAS_{id}	55 [36]	900 [254]	0.449	0.660
Inter-annotator agreement			0.459	0.687

Table 4: Results on GS2012.

Wikipedia and BNC produce disagreeing outputs, and ensembling them is useful as seen in Table 2.

Second, there is no development set. It is not easy to conclude from a large number of model variants with high variances in test set results. For instance, Hashimoto et al. (2014) reported results for 10 settings, where 8 and 9 out of 10 have $\rho < 0.35$ for separate and averaged judgements respectively. Gamallo (2019) presented 11 model variants and FDSAS_{id} only loses to one of them.

All models trained on substantially more data lose to our models across three data sets, except Gamallo (2019)'s. Bootstrap tests on separate judgements across three data sets show that FDSAS_{id} outperforms $\text{BERT}_{\text{BASE}}$ significantly ($p < 0.02$). We also improve over the PixieAE that adopted FDS on GS2011. FDSAS_{id} performs nearly on par with IAA on GS2012, showing that our approach appropriately handles adjectives.

Trained on similar sources and comparable numbers of sentences, Hashimoto and Tsuruoka (2015)'s model outperforms ours by a considerable margin. They concluded that the use of verb matrices allows direct interaction between verbs and their arguments which helps with verb disambiguation. In contrast, the binary semantic function introduced in (11) allows very limited interaction between the two pixies z_0 and z_a, which in the verb disambiguation case correspond to the verb and argument entities respectively. Two advantages of this formulation are that the number of parameters required grows just linearly with respect to the pixie dimension, and the probit approximation in (12) is still applicable. Increasing the expressiveness of the function while keeping a reasonable number of model parameters is an interesting avenue for future work.

6 Conclusion

We analyzed the linguistic and computational challenges of Functional Distributional Semantics and presented a new formulation where we have improved: applicability to diverse natural language structures, computational efficiency, compatibility with contemporary models, and performances on a range of semantic tasks. We believe this work bridges truth-conditional semantics to practical distributional semantics at scale.

Limitations

From a linguistic perspective, we only handle the extensional fragment of natural language. Consequently, modality and temporal information are excluded from the framework. Nevertheless, we train on encyclopediac text which is believed to be a reasonable domain for the extensional restriction.

From a computational perspective, although the reformulated model is already more computationally efficient than previous implementations of FDS, the variable sizes and topologies of input graphs make efficient batching difficult. It is thus not maximally optimized for training on GPUs (statistics are given in §A.3). We currently set the batch size to 1 and perform gradient accumulation to attain a larger effective batch size.

The framework is now only applicable to English because the training data is DMRS graphs parsed from texts using the English Resource Grammar (ERG). This implies: (1) sentences not parsable by the grammar are not available for training, (2) the correct parse for each sentence may not be ranked top by the parser, and (3) for the model to be applicable to other languages, we either need a broad-coverage grammar on these languages for parsing texts to semantic graphs, or adequate semantic dependency graphs of sentences already annotated in these languages. Still, the ERG is a broad-coverage grammar so (1) is largely mitigated.

Ethics Statement

We anticipate no ethical issues directly stemming from our experiments. However, as with all distributional semantic models, our trained model is likely to have picked up social biases present in the training corpus. Any real-world application of a trained model would need to mitigate risks due to such biases.

Acknowledgements

We thank the reviewers, as well as Eric Chamoun, Michael Schlichtkrull and Justin Tang, for their thoughtful feedback and suggestions.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology (LiLT)*, 9.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK. 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.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36, A Festschrift for Joachim Lambek:345–384.

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.

Shib Dasgupta, Michael Boratko, Siddhartha Mishra, Shriya Atmakuri, Dhruvesh Patel, Xiang Li, and Andrew McCallum. 2022. Word2Box: Capturing set-theoretic semantics of words using box embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2263–2276, Dublin, Ireland. Association for Computational Linguistics.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, chapter 3, pages 81–95. University of Pittsburgh Press. Reprinted in: Davidson (1980/2001), *Essays on Actions and Events*, Oxford University Press.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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.

Guy Emerson and Ann Copestake. 2017. Semantic composition via probabilistic model theory. In *IWCS 2017 - 12th International Conference on Computational Semantics - Long papers*.

Primož Fabiani. 2022. Gaussian pixie autoencoder: Introducing functional distributional semantics to continuous latent spaces. Technical report, University of Cambridge, Computer Laboratory.

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).

Daniel Fried, Tamara Polajnar, and Stephen Clark. 2015. Low-rank tensors for verbs in compositional distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 731–736, Beijing, China. Association for Computational Linguistics.

Pablo Gamallo. 2019. A dependency-based approach to word contextualization using compositional distributional semantics. *Journal of Language Modelling*, 7(1):99–138.

Edward Grefenstette. 2013. *Category-theoretic quantitative compositional distributional models of natural language semantics*. Ph.D. thesis, Oxford University, UK.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2015. Concrete models and empirical evaluations for the categorical compositional distributional model of meaning. *Computational Linguistics*, 41(1):71–118.

Abhijeet Gupta, Jason Utt, and Sebastian Padó. 2015. Dissecting the practical lexical function model for compositional distributional semantics. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 153–158, Denver, Colorado. Association for Computational Linguistics.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Jointly learning word representations and composition functions using predicate-argument structures. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1544–1555, Doha, Qatar. Association for Computational Linguistics.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2015. Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 1–11, Beijing, China. Association for Computational Linguistics.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2016. Adaptive joint learning of compositional and non-compositional phrase embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 205–215, Berlin, Germany. 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)*.

Xudong Hong, Asad Sayeed, and Vera Demberg. 2018. Learning distributed event representations with a multi-task approach. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 11–21, New Orleans, Louisiana. Association for Computational Linguistics.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2013. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1590–1601, Seattle, Washington, USA. Association for Computational Linguistics.

Yova Kementchedjhieva, Mark Anderson, and Anders Søgaard. 2021. John praised Mary because _he_? implicit causality bias and its interaction with explicit cues in LMs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4859–4871, Online. Association for Computational Linguistics.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

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.

Yinhong Liu and Guy Emerson. 2022. Learning functional distributional semantics with visual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3976–3988, Dublin, Ireland. Association for Computational Linguistics.

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

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

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

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. Massachusetts Institute of Technology (MIT) Press.

Denis Paperno, Nghia The Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–99, Baltimore, Maryland. Association for Computational Linguistics.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. Current Studies in Linguistics. Massachusetts Institute of Technology (MIT) Press.

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.

Tamara Polajnar, Laura Rimell, and Stephen Clark. 2015. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11, Lisbon, Portugal. Association for Computational Linguistics.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1278–1286.

Laura Rimell, Jean Maillard, Tamara Polajnar, and Stephen Clark. 2016. RELPRON: A relative clause evaluation data set for compositional distributional semantics. *Computational Linguistics*, 42(4):661–701.

Tom De Smedt and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research (JMLR)*, 13:2063–2067.

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

Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Learning semantically and additively compositional distributional representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1277–1287, Berlin, Germany. Association for Computational Linguistics.

Ottokar Tilk, Vera Demberg, Asad Sayeed, Dietrich Klakow, and Stefan Thater. 2016. Event participant modelling with neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 171–182, Austin, Texas. Association for Computational Linguistics.

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.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2013. A tensor-based factorization model of semantic compositionality. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1142–1151, Atlanta, Georgia. Association for Computational Linguistics.

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

Gijs Wijnholds, Mehrnoosh Sadrzadeh, and Stephen Clark. 2020. Representation learning for type-driven composition. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 313–324, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace Transformers: State-of-the-art natural language processing. Unpublished manuscript, arXiv preprint 1910.03771.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. 2003. Understanding Belief Propagation and its generalizations. In Gerhard Lakemeyer and Bernhard Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–269. Morgan Kaufmann Publishers.

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.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. *Advances in neural information processing systems*, 30.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

A Training Details

A.1 Preprocessing

Predicates in DMRS can be divided into two classes, namely abstract predicates and surface predicates. Abstract predicates constitute a very small class. They mostly represent grammatical constructions (e.g., apposition and passivization) and are ignored in this work. On the other hand, surface predicates are exclusively introduced by lexical entries, which include nouns, verbs, adjectives, adverbs, adpositions, conjunctions and overt quantifiers. As in previous work, we assume an extensional model structure with entities being existentially quantified, so we ignore predications that take scopal arguments., e.g., quantifiers and modal verbs. Furthermore, the predicates are lemmas. Derivational and morphological distinctions of word-forms are thus disregarded in the framework. This alleviates data sparsity and aligns to the extensional assumption without further temporal information from inflections, such as tense and aspect.

To keep a reasonable size of vocabulary, we filter out the semantic functions that occur fewer than 100 times and keep only the 100,000 most frequent embeddings for the encoder. After that, we further remove the DMRS graphs with only one distinct predicate. A total of 60,081 semantic functions of 41,046 predicates are trained.

A.2 Hyperparameters and Tuning

For the common hyperparameter values among all models, we set the probability of dropout in the encoder to 0.5, and model parameters are optimized with gradient descent using the Adam optimizer.

For $\text{FDSAS}_{\text{tanh}}$ and FDSAS_{id}, we set K to 32, d to 300, and the dimension of the encoder's embedding to 300. We set β to 0 for $\text{FDSAS}_{\text{tanh}}$ and β for FDSAS_{id} to 0.01. The initial learning rates of both the parameters in the encoder and semantic functions are set to 0.001. The learning rates are multiplied by 0.8 per each epoch. We perform gradient accumulation over 128 batches of size 1. We trained with distributed data parallelism using 3 GPUs, so the effective batch size is 384 and the effective learning rates are 0.000333.

As mentioned in §5.3.2, the performance of $\text{FDSAS}_{\text{tanh}}$ peaks early and plateaus on the development set of RELPRON within 2 epochs whereas FDSAS_{id} is still improving after 6 epochs. Since we train models in an unsupervised manner and the only development set we have is from RELPRON, we have to ensure that training is not stopped prematurely based on the development set for evaluation on all other test sets.

To ensure sufficient time for training, we set a minimum number of epochs to be trained for each of our models, and apply early stopping by taking the performance on the development set of RELPRON at the end of it as a benchmark. Concretely, if a later checkpoint performs better than the benchmark on the development set of REPLRON, we select such a checkpoint for evaluations. To take care of different training dynamics, we set $\text{FDSAS}_{\text{tanh}}$ to train for a minimum of 3 epochs and FDSAS_{id} for a minimum of 7 epochs, before performing the early stops.

A.3 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 $\text{FDSAS}_{\text{tanh}}$ and FDSAS_{id} model takes about 540 and 1260 GPU hours respectively.