Learning Functional Distributional Semantics with Visual Data

Anonymous ACL submission

Abstract

Functional Distributional Semantics is a recently proposed framework for learning distributional semantics that provides linguistic interpretability. It models the meaning of a word as a binary classifier rather than a numerical vector. In this work, we propose a method to train a Functional Distributional Semantics model with grounded visual data. We train it on the Visual Genome dataset, which is closer to the kind of data encountered in human language acquisition than a large text corpus. On four external evaluation datasets, our model outperforms previous work on learning semantics from Visual Genome.

1 Introduction

The target of distributional semantics models is to understand and represent the meanings of words from their distributions in large corpus. Many approaches learn a numerical vector for each word, which encodes its distributional information. They can be roughly divided into two categories: frequency-based methods such as co-occurrence matrix (Sahlgren, 2006), and prediction-based methods such as Word2vec (Mikolov et al., 2013). More recently, progress has been made in learning word representations in a specific context, which are also called contextualized embeddings. Examples include ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019).

Functional Distributional Semantics is a framework that not only provides contextualized semantic representations, but also provides more interpretability. It was first proposed by Emerson and Copestake (2016), and it explicitly separates the modeling of words and the modeling of objects and events that words can refer to. This is a fundamental distinction in predicate logic. While logic is not necessary for all NLP tasks, it is an essential tool for modeling many semantic phenomena (for example, see: Cann, 1993; Allan, 2001; Kamp and Reyle, 2013). For semantic research questions, having a logical interpretation is a clear advantage over vector-based models. We will explain the framework in Section 2.

Another common issue with most distributional semantic models, as discussed by Emerson (2020b), is the symbol grounding problem – if meanings of words are defined in terms of other words, the definitions are circular. During human language acquisition, words are learned while interacting with the physical world, rather than from text or speech alone.

An important goal for a semantic theory is to explain how language relates to the world, and how this relationship is learned. We focus on the Visual Genome dataset, because it provides relatively fine-grained annotations of images, as we will explain in Section 3.

Our main theoretical contribution is to adapt the Functional Distributional Semantics framework, so that it is more suitable for visual data. In order to make it compatible with modern techniques for machine vision, while retaining its logical interpretability, we replace the RBM of previous work with a Gaussian MRF, as explained in Section 4.

Our main empirical contribution is to demonstrate the effectiveness of the resulting model. We evaluate our model from both intrinsic and extrinsic perspectives. In Section 5.1, we intrinsically evaluate the major components of our model, to see how well they fit the training data. In Section 5.2, we evaluate our model on four external evaluation datasets, comparing against previous approaches to learning from Visual Genome, as well as strong text-based baselines. Not only do we confirm Herbelot (2020)'s finding that learning from grounded data is more data-efficient than learning from text alone, but our model outperforms the previous approaches, demonstrating the value of our functional approach.

2 Functional Distributional Semantics

In this section, we summarize previous work on Functional Distributional Semantics, which was first proposed by Emerson and Copestake (2016). The framework takes model-theoretic semantics as a starting point, defining meaning in terms of *truth*. Given an *individual* (also called an *entity*), and given a *predicate* (the meaning of a content word), we can ask whether the predicate is true of that individual. Note that an individual could be a person, an object, or even an event, following neo-Davidsonian event semantics (Davidson, 1967; Parsons, 1990).

Functional Distributional Semantics therefore separates the modeling of words and individuals. An individual is represented in a high-dimensional feature space. The term *pixie* refers to the representation of an individual (Emerson and Copestake, 2017). A predicate is formalized as a binary classifier over pixies. It assigns the value *true* if an individual with those features could be described by the predicate, and it assigns false otherwise. Such a classifier is called a *semantic function*.

The model is separated into a *world model* and a *lexicon model*. The lexicon model consists of semantic functions. Following situation semantics (Barwise and Perry, 1983), the world model defines a distribution over *situations*. Each situation consists of a set of individuals, connected by semantic roles. In our work, we only consider two types of semantic roles: ARG1 and ARG2. For example, the sentence 'a computer is on a desk' describes a situation with three individuals: the computer, the desk, and the event of the computer being on the desk. The computer is the ARG1 of the event, and the desk is the ARG2.

Compared with other distributional semantics models, Functional Distributional Semantics is interpretable in formal semantic terms. Emerson (2020a) proposed an autoencoder-like structure which can be trained efficiently from semantic dependency graphs.

Because individuals are explicitly modeled, grounding the pixies is more theoretically sound than grounding word vectors. The framework has clear potential for learning grounded semantics, which we explore in this paper.

3 Visual Genome

The Visual Genome dataset contains over 108 thousand images and five different formats of annotations, including regions, attributes, relations, object instances and question answering. In this work, we only consider the datatype of relations, which are formulated as predicate triples. Each triple contains two objects in the image and one relation between them. The objects are identified with bounding boxes, as illustrated in in Fig. 1. The object predicates are nouns or noun phrases, and the relation predicates can be descriptive verbs, prepositions or prepositional phrases.

Figure 1: An example image in Visual Genome, annotated with the relation ['Computer', 'ON', 'Desk']

There are many works using Visual Genome as a grounded information source. For example, Fukui et al. (2016) use Visual Genome to ground its visual question answering system. The fine-grained annotations make Visual Genome a compelling dataset for studying lexical semantics. Kuzmenko and Herbelot (2019) and Herbelot (2020) learn semantics from the annotations, while discarding the images themselves. They trained word embeddings with a count-based method and a Skip-gram-based method, respectively. This methodology, of extracting word relations from an annotated image dataset, was also analyzed and justified by Schlangen (2019).

To our knowledge, there has been no previous attempt to use grounded visual data to train a Functional Distributional Semantics model, nor to utilize the visual information of Visual Genome to learn natural language semantics.

4 Model and Methods

We will explain the probabilistic structure of our model in Section 4.1, and how we train the components in Sections 4.2 and 4.3. In Section 4.4, we present an inference model to infer latent pixies from words and the context.

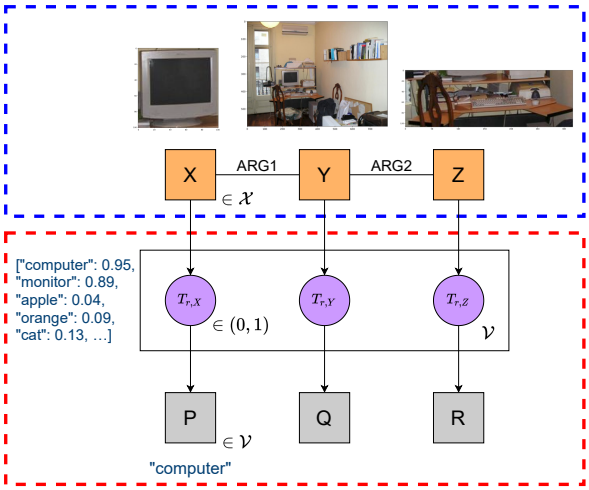

Figure 2: Our probabilistic graphical model. The top blue box contains the world model, which learns the joint distribution of the observed pixies X, Y and Z from their corresponding images. The bottom red box shows the lexicon model, where each semantic functions in the vocabulary $\mathcal{V}$ is applied to each pixie. For each pixie, one predicate is generated, with probability proportional to the probability of truth.

4.1 Probabilistic Graphical Model

We define a graphical model which jointly generates pixies and predicates, as shown in Fig. 2. It has two parts. The world model is shown in the top blue box, which models the distribution of situations, or in another words, the joint distribution of pixies. It is an undirected graphical model, with probabilistic dependence according to the ARG1 and ARG2 roles, as further explained in section 4.2. The lexicon model is shown in the bottom red box, which models each predicate as a semantic function. It is a directed graphical model. For each pixie, it produces a probability of truth for each predicate (which are not observed), as well as generating a single predicate (which is observed), as further explained in Section 4.3.

Given a labelled image triple, the model can be trained by maximizing the likelihood of generating the data, including both observed predicates and observed pixies. The likelihood can be split into two parts, as shown in Eq. 1, where s is a situation (a pixie for each individual), and g is a semantic dependency graph (a predicate for each individual). The first term is the likelihood of generating the observed situation, which is modeled by the world model. The second term is the likelihood of generating the dependency graph given an observed situation, which is modeled by the lexicon model. Therefore, we can optimize parameters of the two parts separately.

$$\log P(s, g) = \log P(s) + \log P(g|s) \qquad (1)$$

4.2 World Model

The world model learns the joint distribution of pixies, as shown in the top half of Fig. 2. The individuals are grounded by images, so we can obtain the pixie vectors by extracting visual features for individuals from their corresponding images. For object pixies, they are grounded by their corresponding bounding boxes. For event pixies, Visual Genome does not have labelled bounding boxes for them and their meaning tends to be more abstract, so we use the whole image to ground them. As a feature extractor, we use ResNet101, a Convolutional Neural Network (CNN) pre-trained on ImageNet. To further reduce redundant dimensions, we perform PCA on the last layer of the CNN. We take the output of PCA as the pixie space $\mathcal{X}$.

A situation s is a collection of pixies within a semantic graph. In this work, we only consider graphs with three nodes, connected by the roles ARG1 and ARG2, to match the structure of Visual Genome relations.

In previous work, the world model was implemented a Restricted Boltzmann Machine (RBM). However, an RBM uses binary-valued vectors, which is not compatible with the real-valued vectors produced by a CNN. Furthermore, an RBM does not give normalized probabilities, which means that computationally expensive techniques are required, such as MCMC, used by (Emerson and Copestake, 2016), or Belief Propagation, used by (Emerson, 2020a).

We model situations with a Gaussian Markov Random Field (MRF). For an n-dimensional pixie space, this gives a $3n$-dimensional Gaussian distribution, with parameters μ and Σ for the mean and covariance. As shown in the first term of Eq. 1, we would like to maximize $P(s)$. For a Gaussian distribution, the maximum likelihood estimate (MLE) has a closed-form solution, which is simply the sample mean and sample covariance. However, because the left and right pixies in Fig. 2 are conditionally independent given the event pixie, we force the top right and bottom left pixie blocks of the precision matrix Σ^{-1} to be zero.

$$P(s) = \mathcal{N}(s; \mu, \Sigma), \qquad (2)$$

4.3 Lexicon Model

The lexicon model learns a list of semantic functions, each corresponds to a word in predicate vocabulary $\mathcal{V}$. The semantic function $t_r(x)$ for a given predicate r is a logistic regression classifier over the pixie space, with a weight vector v_r. From the perspective of deep learning, this is a single neural net layer with a sigmoid activation function. As shown in Eq. 3, the output is a probabilistic truth value ranging between $(0, 1)$.

$$t_r(x) = \sigma(v_r \cdot x) \tag{3}$$

As shown in the second row of Fig. 2, all semantic functions are applied to each pixie. Based on the probabilities of truth, a single predicate is generated. The probability of generating a specific predicate r for a given pixie $\mathbf{x}$ is computed as shown in Eq. 4. The more likely a predicate is to be true, the more likely it is to be generated.

$$P(r|x) = \frac{t_r(x)}{\sum_i t_i(x)} \tag{4}$$

The lexicon model is optimized to maximize $\log P(g|s)$, the log-likelihood of generating the predicates given the pixies. This can be done by gradient descent.

4.4 Variational Inference

When learning from Visual Genome, pixies are grounded by images. However, when applying the model to text, the pixies are latent. We provide an inference model to infer latent pixie distributions given observed predicates. This inference model is used in Section 5.2 on textual evaluation datasets.

Exact inference of the posterior $P(s|g)$ is intractable, because this requires integrating over the high-dimensional latent space of s. This is a common problem when working with probabilistic models. Therefore we use a variational inference algorithm to approximate the posterior distribution $P(s|g)$ with a Gaussian distribution $Q(s)$. For simplicity,we assume that each dimension of $Q(s)$ is independent, so its covariance matrix is diagonal.

In Fig. 3, the graphical model illustrates this assumption, as there is no connection among the pixie nodes in the middle row. Following the procedure of variational inference, the approximate distribution $Q(s)$ is optimized to maximize the Evidence Lower Bound (ELBO), given in Eq. 5. This can be done by gradient descent.

$$\mathcal{L} = E_Q\Big[\log P(g|s)\Big] - \beta D_{\mathrm{KL}}\big(Q(s)||P(s)\big) \tag{5}$$

The first term measures how well $Q(s)$ matches the observed predicates, according to the lexicon model $P(g|s)$. The second term measures how well $Q(s)$ matches the world model $P(s)$. We would like to emphasize the likelihood of generating the observed predicates, so we down-weight the second term with a hyper-parameter β, similarly to a β-VAE (Higgins et al., 2017).

Exactly computing the first term is intractable. Emerson (2020a) used a probit approximation, but we instead follow Daunizeau (2017), who derived the more accurate approximations given in Eqs. 6 and 7, where x has mean μ and variance Σ. The second approximation is particularly important, as we aim to maximize the log-likelihood.

$$E[\sigma(x)] \approx \sigma\left(\frac{\mu}{\sqrt{1 + 0.368\Sigma}}\right) \tag{6}$$

$$E[\log \sigma(x)] \approx \log \sigma\left(\frac{\mu - 0.319\Sigma^{0.781}}{\sqrt{1 + 0.205\Sigma^{0.870}}}\right) \tag{7}$$

The second term of Eq. 1 is the Kullback Leibler (KL) divergence between two Gaussians, which has the closed-form formula given in Eq. 8, where k is the total dimensionality.

$$\begin{aligned} D_{\mathrm{KL}}(Q||P) = \frac{1}{2}\Big[\log\frac{|\Sigma_P|}{|\Sigma_Q|} - k + \mathrm{tr}(\Sigma_P^{-1}\Sigma_Q) \\ +(\mu_Q - \mu_P)^T\Sigma_P^{-1}(\mu_Q - \mu_P)\Big] \end{aligned} \tag{8}$$

As illustrated in Fig. 3, variational inference allows us to calculate quantities such as the probability that an animal which has a tail is a horse. To obtain the inferred distribution for a single pixie, we need to marginalize the situation distribution $Q(s)$. From the independence assumption, this simply means taking the parameters for the desired pixie. Then we can apply the semantic function for r on the inferred pixie x, as shown in Eq. 9, which can be approximated using Eq. 6.

$$t_r(x) \approx E_Q\big[\sigma(v_r \cdot x)\big] \tag{9}$$

Although $Q(s)$ assumes independence, its parameters are jointly inferred based on all predicates. This is because the KL-divergence in Eq. 8 depends on Σ_P, which is nonzero between each pair of pixies linked by a semantic role.

For example, in Fig. 3, the truth of 'horse' for X depends on the observed predicate 'tail' or 'paw'. This is not a direct dependence between words, but rather relies on three intermediate representations

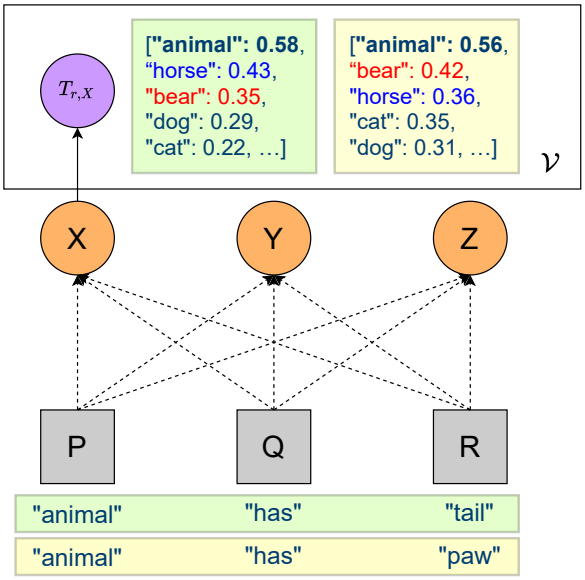

Figure 3: Graphical inference model: The pixies X, Y and Z in the middle row are jointly inferred from the observed predicates P, Q and R in the bottom row, using variational inference. Semantic functions are applied to X, to give probabilities of truth. As well as the observed predicate, the model predicts that other predicates may also be true. Two examples are given, in green and yellow, showing how the predicted truth values for X depend on all observed predicates.

(the three pixies), all of which are expressed in terms of visual features. The first term of the ELBO connects the semantic function for 'tail' or 'paw' to the variational parameters for Z. The second term of the ELBO connects the variational parameters for Z and Y (based on the world model covariance for ARG2) as well as Y and X (based on the world model covariance for ARG1). Finally the semantic function for 'horse' is applied to the variational distribution for X.

In this example, the model correctly infers that an animal with a tail is more likely to be a horse and an animal with paws is more likely to be a bear. We notice that the truth values are generally low for all semantic functions. Even the highest truth is only around 0.58. This illustrates that the model is not very certain, which might be expected since the model is performing inference on visual features, but the training image data is noisy.

For some evaluation datasets, we need to perform inference given a single predicate. This can be done by marginalizing the joint distribution. Which pixie variable to choose, out of the three, should depend on the Part-Of-Speech (POS) of the word. For nouns, the pixie node X or Z should be used, as a noun should play the role of ARG1 or ARG2. For verbs and prepositions, the node Y should be used, as they usually describe the relation.

5 Evaluation

To train our model, we follow the same pre-processing and filtering of Visual Genome as Herbelot (2020). Details of pre-processing and hyper-parameters are given in the appendix.

5.1 Intrinsic Evaluation

In this section, we examine whether a Gaussian MRF is a suitable choice for the world model, and whether the pixies in the pixie space are linearly separable such that the logistic semantic functions can successfully classify them.

5.1.1 World Model Evaluation

The world model learns a Gaussian distribution for the observed situations. In this section, we justify this choice by evaluating the fitting errors.

Fig. 4 shows density histograms for two example pixie dimensions and their corresponding best-fit (MLE) Gaussian curves. The left histogram is an example for a majority of the pixie dimensions, which is tightly matched by the best-fit Gaussian. In other cases, as shown on the right, there are imbalanced tails and asymmetry. Despite their skewness and kurtosis, which make them look more like a Gamma distribution, they are still generally bell-shaped and the departure is not so heavy.

To quantify the errors, we measure the Wasserstein distance, the area of the histogram missing from the best-fit Gaussian. Across all 100 pixie dimensions, the mean percentage missing is 7% with a variance of 1%. A more flexible model might give better modeling performance, which could be a future improvement direction. Nonetheless, we consider this level of error to be acceptably low.

5.1.2 Lexicon Model Evaluation

In this experiment, we investigate if our approach to model the semantic functions as logistic regression classifiers is suitable. In particular, a logistic regression classifier is a linear classifier, which means if the data is not linearly separable, it would have inferior performance.

We computed the Area Under Curve for the Receiver Operating Characteristic (AUC-ROC), for all predicates in the vocabulary. For each predicate we randomly select equal amount of negative example pixies with its positive examples. The average

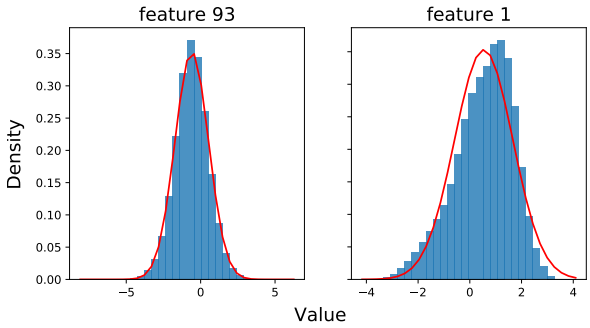

Figure 4: Density histograms for two selected pixie dimensions, across the 2.8M training instances. Best-fit Gaussian curves of the histograms are shown in red.

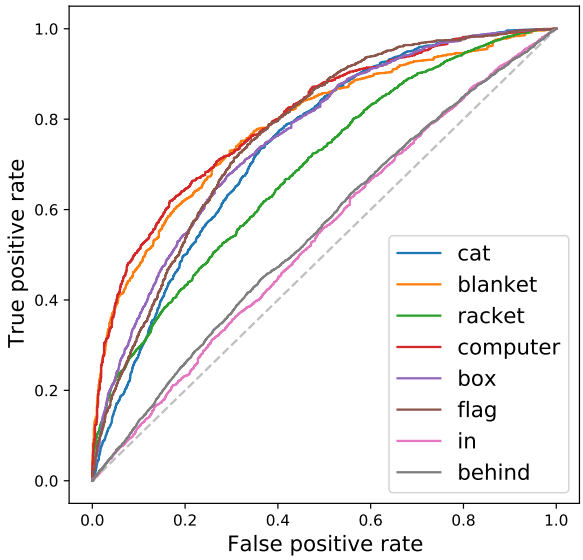

Figure 5: ROC curves of the semantic functions for selected predicates. Higher left is better performance.

score is 0.79 for object predicates, and 0.58 for event predicates. We also present the ROC for a few example predicates in Fig. 5.

We can see that object classifiers have generally better performance. The classifier for 'racket' shows slightly worse performance than the others, whose reason might be its lower frequency. Compared to object predicates, the semantic functions for event predicates generally perform worse. There are two potential reasons which could be improved in future work. Firstly, we used visual features generated from the whole image to represent the event pixie, which is often not specific enough to identify the event. Secondly, a logistic regression classifier might not be sophisticated enough for this classification problem.

5.2 Extrinsic Evaluation

In this section, we use external semantic evaluation datasets, to give a direct comparison against previous work, and to test whether our model can generalize beyond the training data. We evaluate on two lexical similarity datasets in Section 5.2.2, and two contextual datasets in Section 5.2.3. We compare against two types of baseline: models trained on a large corpus and models trained on Visual Genome.

For these datasets, our model must assign similarity scores for predicate or triple pairs, which we compute as follows. The pixie values are inferred from the first predicate or triple in the pair. Then all semantic functions from the predicate vocabulary are applied to that pixie. Then the ranking of the second predicate in the pair over all potential predicates in the evaluation dataset is taken as the similarity score. Therefore, smaller ranking means higher similarity between predicates.

Finally, because there are discrepancies between vocabularies used in Visual Genome and the evaluation datasets, we follow Herbelot (2020) in filtering the evaluation datasets according to the Visual Genome vocabulary, and use the filtered datasets to evaluate all models. For the two lexical datasets, we exactly follow Herbelot's filtering conditions to give a direct comparison.

For the contextual datasets, this filtering is too strict, resulting in zero vocabulary coverage. For these datasets, we apply looser filtering, with details given in the appendix. This also requires retraining our model and the Visual Genome baselines on a more loosely filtered training set.

5.2.1 Baselines

Visual Genome Baselines: We re-implement two previously proposed models learning distributional semantics from Visual Genome, described in Section 3. A simple count-based model was proposed by Kuzmenko and Herbelot (2019), which we refer to as VG-count. Herbelot (2020) improved on this and proposed EVA, a Skip-gram model that can be trained on the same kind of co-occurrence data.

Large Corpus Baselines: We trained two Skip-gram Word2vec models (Mikolov et al., 2013) using 1 billion and 6 billion tokens from Wikipedia, using Gensim (Řehůřek and Sojka, 2010). We will refer to them as Word2vec-1B and Word2vec-6B. The window sizes are set to be 10 in two directions, so they contextualize with far more words than our model. We also use Glove (Pennington et al., 2014) trained on 6 billion Wikipedia tokens as another strong baseline, which we refer to as Glove-6B. For all three baselines, the dimensionality is set to 300.

	Models	Lexical datasets		Contextual datasets	
		MEN	**SimLex-999**	**GS2011**	**RELPRON**
Large corpus baselines	Word2vec-1B	0.641	0.384	0.265	0.381
	Word2vec-6B	0.652	0.397	0.278	0.401
	Glove-6B	**0.736**	0.409	**0.293**	**0.432**
VG baselines	VG-count	0.336	0.224	0.063	0.038
	EVA	0.543	0.390	0.068	0.032
Proposed approach	Our model	0.639	**0.431**	0.171	0.075

Table 1: Evaluation results. For MEN, SimLex-999 and GS2011, the metric is Spearman correlation; for RELPRON, mean average precision. All models are evaluated on subsets of the data covered by the VG vocabulary.

Compared to the large corpus baselines, our model has fewer parameters per word (100 vs. 300), and is trained on far fewer data points (2.8M relation triples vs. 1B or 6B tokens).

5.2.2 Lexical similarity and relatedness

We use two lexical similarity/relatedness datasets, MEN (Bruni et al., 2014) and Simlex-999 (Hill et al., 2015), both of which give scores for pairs of words. MEN contains 3000 word pairs, and SimLex-999 contains 999 pairs. After filtering for the Visual Genome vocabulary, we have 584 pairs for MEN and 169 pairs for SimLex-999.

MEN evaluates relatedness, while SimLex-999 evaluates similarity. For example, 'coffee' and 'cup' are related, but not similar. Capturing similarity rather than relatedness is hard for most text-based distributional semantics models because they build concept representations based on their co-occurrence in corpora, which generally reflects relatedness but not similarity. However, similarity might be more directly reflected in terms of visual features which can be captured by our model.

The results are shown in Tab. 1. Our model outperforms the two baselines trained on Visual Genome, and matched the performance of Word2vec-1B (the difference is statistically insignificant, $p>0.5$).

If we force our model to evaluate on the full 1000 word pairs in the MEN test set (assigning the median similarity score to the out-of-vocabulary pairs), it still achieves 0.304. Using the loosely filtered training set, our model can achieve the even higher score of 0.670 (on the same strictly filtered subset of MEN). This illustrates that one limit of our model's performance is the size of the Visual Genome dataset. In contrast, the performance of Word2vec does not improve much as the training data increases from 1B to 6B, which suggests there is a limit on how much can be learnt from local textual co-occurrence information alone.

On SimLex-999, our model achieves 0.431, which outperforms all baselines. Compared to Glove-6B, the strongest baseline, it is weakly significant ($p<0.15$). This might justify our point that there is advantage of learning similarity from visual features. Additionally, our model can use parameters and data more effectively and efficiently than Word2vec and Glove, achieving better performance with less training data and fewer parameters.

Compared with VG-count and EVA, our model can understand more semantics because it learns from the visual information. As far as we know, we have achieved a new state of the art on learning lexical semantics from Visual Genome. Combining results across all four datasets (including the contextual datasets below), the difference between our model and EVA is highly significant ($p<0.001$).

5.2.3 Contextual semantics

We consider two contextual evaluation datasets. GS2011 (Grefenstette and Sadrzadeh, 2011) gives similarities of verbs in a given context. Each data point is a pair of subject-verb-object triples, where only the verbs are different. For example, ['table','show', 'result'] and ['table', 'express', 'result'] are judged highly similar. The dataset has 199 distinct triple pairs and 2500 judgment records from different annotators. The evaluation metric is Spearman correlation across all judgments. As Van de Cruys et al. (2013) point out, the second verb in each pair is often nonsensical when combined with the corresponding subject and object. Therefore, we only compare the triple pairs in a single direction, inferring pixies from the first triple and applying the second verb's semantic function.

RELPRON (Rimell et al., 2016) evaluates compositional semantics. It contains a list of terms,

each associated with around 10 properties. Each property is a noun modified by a subject or object relative clause. For example, the term 'theater' has the subject property ['building', 'show', 'film'] and object property ['audience', 'exit', 'building']. The task is to find the correct properties for each term, evaluated as Mean Average Precision (MAP). The development set contains 65 terms and 518 properties; the test set, 73 terms and 569 properties.

Under the loosely filtered condition, our subset of GS2011 contains 252 similarity judgments; RELPRON, 57 terms and 150 properties.

Rimell et al. (2016) find that vector addition performs surprisingly well at combining contextual information. Therefore, for all baselines, we represent a triple by adding the three word embeddings. As aforementioned, we re-train our model and the two VG baselines with loosely filtered data.

The results are shown in Tab. 1. The corpus models outperform the VG models. However, this is perhaps expected given that the vocabulary in GS2011 and RELPRON is more formal, and even when they are covered in Visual Genome, their frequencies are low: for RELPRON, 54% of the covered vocabulary has frequency below 100, compared to only 6% for MEN. Furthermore, GS2011 evaluates similarity of verbs, but we saw in Section 5.1.2 that our model is less accurate for verbs.

However, our model outperforms both VG baselines on both datasets. This suggests that our model is less affected by data sparsity. For the baselines, if a training triple contains multiple rare predicates, the sparsity problem is compounded. However, our model relies on the images, whose visual features are shared across the whole training set.

5.3 Truth regularization

To make the probabilistic truth values more interpretable, Emerson (2020a) proposes a regularization term which penalizes the model if all truth values stay close to 0. This would modify the loss function in Eq. 4, to give Eq. 10, with a hyperparameter α that we set to 0.5.

$$\mathcal{L} = \log \frac{t_r(x)}{\sum_i t_i(x)} + \alpha \log t_r(x) \qquad (10)$$

We find that adding the log-truth term improves performance on intrinsic evaluation, but decreases performance on extrinsic evaluation. Applying the analysis in Section 5.1.2, the average AUC-ROC is 0.86 for object predicates and 0.60 for event predicates. This is illustrated in Fig. 6 for the same

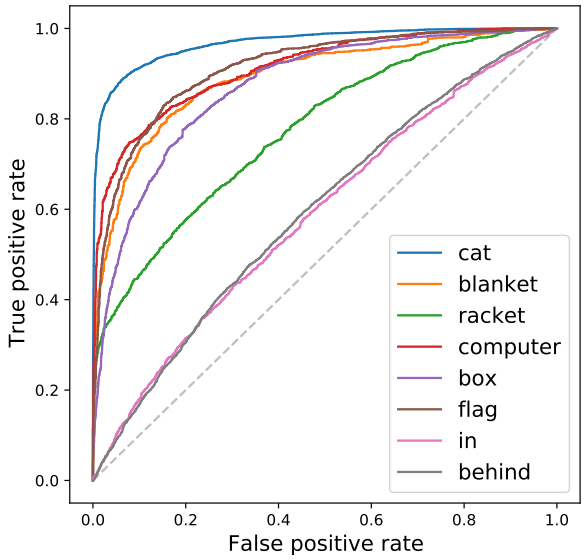

Figure 6: ROC curves of the semantic functions for selected predicates, for the truth-regularized model.

example predicates as Fig. 5. In contrast, when evaluating on MEN and SimLex-999, this model achieves only 0.602 and 0.381 respectively.

The log-truth term makes predicates true over larger regions of pixie space. As shown by the intrinsic evaluation, this is helpful when considering each classifier individually. However, the regions of different predicates also overlap more, which seems to hurt their overall performance on the external datasets. To quantify this, we calculate the total truth of all predicates, for 1000 randomly selected images. For the original version of our model, on average 0.83 predicates are true for an image. This is slightly below 1, illustrating the problem Emerson aimed to avoid. However, with the log-truth term, it becomes 25.5, which may have over-corrected the problem.

6 Conclusion

In this paper, we proposed a method to train a Functional Distributional Semantics model with visual data. Our model outperformed the previous works and achieved a new state of the art on learning natural language semantics from Visual Genome. Further to this, our model achieved better performance than Word2vec and Glove on Simlex-999 and matched Word2vec-1B on MEN. This shows that our model can use parameters and data more efficiently than Word2vec and Glove. Additionally, we also showed that our model can successfully be used to make contextual inferences.

References

Keith Allan. 2001. *Natural language semantics*. Blackwell.

Jon Barwise and John Perry. 1983. *Situations and Attitudes*. Massachusetts Institute of Technology (MIT) Press.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*.

Ronnie Cann. 1993. *Formal semantics: an introduction*. Cambridge University Press.

Jean Daunizeau. 2017. Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

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

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

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 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*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Aurélie Herbelot. 2020. Re-solve it: simulating the acquisition of core semantic competences from small data. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 344–354, Online. Association for Computational Linguistics.

I. Higgins, L. Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. 2017. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Hans Kamp and Uwe Reyle. 2013. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy*. Springer.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Elizaveta Kuzmenko and Aurélie Herbelot. 2019. Distributional semantics in the real world: Building word vector representations from a truth-theoretic model. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 16–23, Gothenburg, Sweden. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

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

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

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.

Magnus Sahlgren. 2006. *The Word-Space Model : Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Institutionen för lingvistik.

David Schlangen. 2019. Natural language semantics with pictures: Some language & vision datasets and potential uses for computational semantics. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 283–294, Gothenburg, Sweden. Association for Computational Linguistics.

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.

A Training and evaluation details

A.1 Filtering

For EVA, Herbelot (2020) filtered the Visual Genome dataset with a minimum occurrence frequency threshold of 100 in both ARG1 and ARG2 directions. After filtering, the resulting subset contains 2.8M relation triples and the vocabulary size is 1595. When evaluating the external datasets, it only includes the noun predicates. For the results reported in the intrinsic evaluations and in Tab. 1 where we specify 'strict filtering', we follow the same filtering conditions with EVA.

We also train our model with a less strict filtering setting, where the minimum frequency threshold is set at 10 in at least one direction. Under this filtering setting, the resulting subset contains 3.4M relation triples and the vocabulary size is 6788. When evaluating the external datasets, we includes all the covered predicates regardless of their POS. The results under the 'loose filtering' columns in Tab. 1 are evaluating under this setting. Additionally, every time we use the model trained under this setting, we will emphasize it is 'under less strict filtering condition'.

A.2 CNN

For the visual feature extractor, the pretrained CNN, we used ResNet101 (He et al., 2016), which has 101 layers deep and trained on ImageNet (Deng et al., 2009).

A.3 PCA

During the PCA transform, we reduce the pixie dimension from 1000 to 100, whose eigenvalue components cover 93.2% of the total information. After the PCA, we re-scaled each dimension by dividing them over the square root of their corresponding eigenvalues and scale up by a factor of 1.15, such that the determinant of the covariance matrix of the world model is close to 1.

A.4 Lexicon model training

When training the lexicon model, we used L2 regularisation with a weight of $5\mathrm{e}{-8}$ and the Adam optimizer (Kingma and Ba, 2015). We train the lexicon model for 40 epochs and the learning rate is set at 0.01 with a step scheduler which reduces the learning rate by a factor of 0.4 every 5 epochs. The hyper-parameters are tuned on the training data to maximize the number of predicates such that at least one image annotated with that predicate has a truth value of at least 0.1. For the model trained on strictly filtered data, the number of such predicates reaches 1343 out of the vocabulary size 1595, while for loosely filtered model, the number is 4453 out of 6788.

A.5 Variational inference optimization

For the variational inference, the hyper-parameter β is set to be 0.1. We run gradient descent for 800 epochs with initial learning rate of 0.03 and a step scheduler which reduces the learning rate by a factor of 0.6 every 50 epochs. The hyper-parameters for variational inference are tuned to maximize the ELBO on the filtered subset of MEN. (The ELBO does not depend on the similarity scores, just the input triples.) Tuning these hyper-parameters has no effect on the training of the world model or the lexicon model. The scores shown in Table 1 are the results averaged over 5 random seeds.

A.6 Statistical tests

All statistical tests are two-tailed bootstrap tests, which follows the recommendations of Dror et al. (2018). We use 1000 samples for each test.