

Article

Mapping Distributional Semantics to Property Norms with Deep Neural Networks

Dandan Li and Douglas Summers-Stay *

U.S. Army Research Laboratory, Adelphi, MD 20783, USA; happydandan2016@gmail.com

* Correspondence: douglas.a.summers-stay.civ@mail.mil; Tel.: +1-301-394-1990

Received: 23 April 2019; Accepted: 19 May 2019; Published: 25 May 2019



Abstract: Word embeddings have been very successful in many natural language processing tasks, but they characterize the meaning of a word/concept by uninterpretable “context signatures”. Such a representation can render results obtained using embeddings difficult to interpret. Neighboring word vectors may have similar meanings, but in what way are they similar? That similarity may represent a synonymy, metonymy, or even antonymy relation. In the cognitive psychology literature, in contrast, concepts are frequently represented by their relations with properties. These properties are produced by test subjects when asked to describe important features of concepts. As such, they form a natural, intuitive feature space. In this work, we present a neural-network-based method for mapping a distributional semantic space onto a human-built property space automatically. We evaluate our method on word embeddings learned with different types of contexts, and report state-of-the-art performances on the widely used McRae semantic feature production norms.

Keywords: distributional semantics; word embeddings; property norm; neural networks

1. Introduction

Semantic word representation plays important roles in a broad range of natural language processing (NLP) tasks, including translation, query expansion, information extraction, and question answering. There are two main branches of previous work: (1) distributional semantic models learned with different types of contexts from large text corpora [1–3]; (2) property-based representation in terms of constituent properties generated by participants in property norming studies [4,5], extracted from manually-curated knowledge bases, such as FreeBase and Wikidata [6], or learned from a text [7].

Distributional semantic models characterize the meaning of a word through the contexts in which it appears. These models rely on the distributional hypothesis—that words occurring in similar contexts tend to have similar meanings [8,9]. Word2vec (W2V) is a popular word embedding method that learns word vectors from unprocessed text using a fixed-size context window. However, a context window with an unsuitable size may miss significant contexts and include irrelevant ones. Levy and Goldberg [3] propose dependency-based word embeddings (DEPs), generalizing the skip-gram model to include negative sampling, and moving from bag-of-words contexts to nonlinear syntactic contexts that are derived from dependency parse-trees generated by an automated parser. The empirical evidence shows that distributional models can do a good job in capturing word similarities. However, the basis vectors of distributional models tend to be uninterpretable, unlike property-based representations where each bit encodes the presence or absence of a particular property for that concept. For example, distributional models can tell us that *airplane* is similar to *aircraft* and *pilot* with different similarity scores, but it is difficult to differentiate how *airplane* is related to *aircraft* from how it is related to *pilot* based on these models. This is one of the main drawbacks of distributional models [10].

There is a wide consensus in cognitive psychology that the meaning of a concept is a complex assembly of properties that characterize how they are related to the concept. For example, the concept *bluebird* can be represented by properties like *is_a_bird*, *to_fly*, *has_feathers*, and *is_blue*. There are several ways to obtain a comprehensive set of concept properties. The widely used property norm dataset is from [4], which consists of 541 concepts and 2526 properties. Devereux, Tyler, Geertzen, and Randall [5] offer the largest property norm dataset to date that includes 638 concepts. All individual properties are generated by participants in property norming studies. Baroni, Murphy, Barbu, and Poesio [7] present a fully automatic method for extracting a set of concept descriptions directly from an English part-of-speech-tagged corpus. Existing semantic knowledge bases, such as ConceptNet and FreeBase, are good sources of properties/attributes and are used fairly extensively in NLP tasks [6,11–13]. Producing property norms by human annotation can be expensive, however. Moreover, it is still unrealistic to extract accurate properties from a large-scale text corpus [14]. This raises the question of how we predict property norms for new concepts.

The human brain organizes long-term, explicit, semantic memory in a way that is very different from a knowledge base [15]. People seem to be able to call up a concept from memory

- by thinking of concepts whose meaning is similar;
- by completing an analogy;
- based on properties, including properties people did not previously realize characterized that concept;
- by deductive reasoning based on related concepts.

Suppose I don't actually know what properties a "jar" has. What I know instead is that a "jar" must be something like a "plate" or a "pan" because they always appear in similar textual contexts. Figure 1 shows a detailed description of the example. Now, two questions are raised: (1) How can I infer properties of "jar" from known properties of "plate" and "pan" based on a distributional semantic model? (2) What kind of distributional model is good at doing the above inference? These questions lead us to develop the research topics of this work.

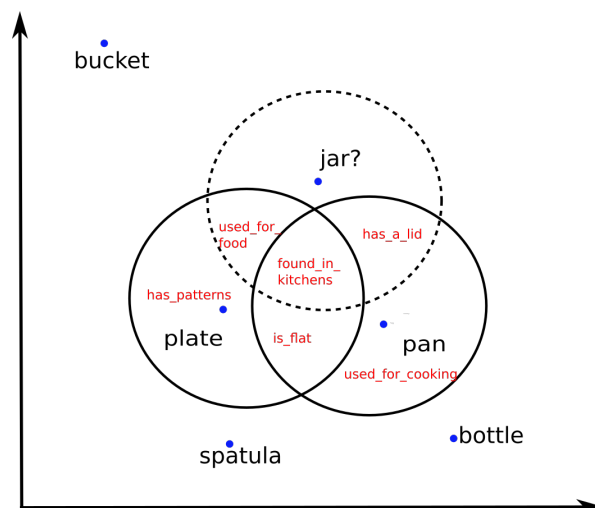


Figure 1. Inferring properties of "jar" from known properties of "plate" and "pan" based on distributional semantic models.

Our main contributions include the following:

- We predict properties for new concepts by learning a nonlinear mapping from a distributional semantic space to human-annotated property space based on multilayer perceptrons.
- We explore the performances of distribution models with different types of contexts on this task by qualitative and quantitative analyses.

- Using a publicly available dataset of property norms, we show that high-quality property norms can indeed be induced from standard distributional data.

Related work is surveyed in Section 2. We then describe the existing distributional models with different context types and the human-built property norms and analyze the capability of producing property-based representations of concepts from distributional models in Section 3. Section 4 presents our multilayer perceptrons-based method to learn properties of concepts from distributional models. In Section 5, experimental results and the evaluation are reported. Finally, Section 6 draws conclusions and discusses future work.

2. Related Work

Several previous works addressed property inference from distributional data. Herbelot and Vecchi [16] present an approach to automatically map a standard distributional semantic space onto a set-theoretic model. In their work, the properties of a concept for the McRae norms have five quantifiers: NO, FEW, SOME, MOST, ALL. They assume there is a linear relationship between the above two spaces and then estimate the coefficients of the function using partial least square regression (PLSR). Dernoncourt [17] introduces a mode predictor, which considers the distribution of quantifiers among properties. Făgărășan, Vecchi, and Clark [18] explore the possibility of generalizing property-based representation to a large scale dataset. Baroni and Lenci [19] explore the capability of producing property-based descriptions of concepts from computational models that are derived from word co-occurrence statistics. Rubinstein, Levi, Schwartz, and Rappoport [20] show that the extent to which distributional models represent semantic knowledge varies greatly depending on what type of knowledge is represented. Gupta, Boleda, Baroni, and Padó [6] pursue the hypothesis that distributional vectors also implicitly encode referential attributes that are extracted from FreeBase or Wikidata. Strudel [7] is an unsupervised algorithm to extract a structural and comprehensive set of concept descriptions directly from an English corpus and then represent concepts by weighted properties. Erk [21] proposes a probabilistic mechanism for distributional property inference. Boleda and Herbelot [22] present an overarching semantic framework called “formal distributional semantics”, which combines formal and distributional semantics together.

There are also several recent papers that focus on learning a mapping between different semantic modalities and creating visual attributes for unseen concepts. Bulat, Kiela, and Clark [23] explore the automatic prediction of property norms for new concepts by learning a mapping from images. A linear regression is learned to transform image vector representations into the representations of the concepts in a distributional semantic space learned from text [24]. Silberer, Ferrari, and Lapata [25] create a large-scale taxonomy of 600 visual attributes representing more than 500 concepts in the McRae dataset and 700 K images in ImageNet. Zellers and Choi [26] model attributes of action verbs (both visual and linguistic) to perform zero-shot activity recognition. In this work, we explore how to build a nonlinear mapping from a distributional semantic space to a human-built property space based on multilayer perceptrons (MLPs).

3. Distributional Models and Property Norms

Word2vec (W2V), a popular method, generates dense embeddings by either a skip-gram model or using a continuous bag-of-words (CBOW). The skip-gram model’s training objective is predicting the words in a context window of $2k$ words surrounding the target word. For example, when we set $k = 1$ (denoted by W2V1), the contexts of the target word w_t are w_{t-1} and w_{t+1} and predict these from the word w_t . However, a context window with a larger size k may capture “coincidental” contexts that are not directly related to the target word. Dependency-based word embeddings (DEPs) proposed by Levy and Goldberg [3] use negative sampling to generalize the skip-gram model, thereby moving from bag-of-words contexts to disconnected syntactic contexts. We use the sentence “A careful astronomer discovered something funky in Uranus’ orbit” to demonstrate the differences of the above distributional models. Table 1 lists the contexts of W2V1, W2V3, and DEP for the target word “astronomer”.

Table 1. Contexts extracted by different models: W2V1, W2V3, dependency-based word embedding (DEP).

Distributional Models	Contexts
W2V1	careful, discovered
W2V3	a, careful, discovered, something, funky
DEP	careful/amod, discovered/nsubj ⁻¹

We see that W2V with a smaller context window and DEP with syntactic context can capture more focused information about the target word and also filter out “coincidental” contexts such as *a* and *funky*. Empirical analyses [27,28] show that narrow context window and syntactic context window models tend to give high ratings to semantic-similar word pairs like *airplane/aircraft*, while wide-context models give high ratings to topically related pairs like *airplane/pilot*. Word pairs semantic-similarity is also called “AP-similarity” [21], which encompasses synonymy, hypernymy, and co-hyponymy. Erk [21] proposes that AP-similarity can be characterized as property overlap, and interpreting AP-similarity as property overlap allows us to draw property inferences from observed distributional similarity based on narrow context or syntactic context models. In our example, the contexts of W2V1 and DEP found for the noun target *astronomer* are the adjectival modifier *careful* and the verb *discovered*. The selectional constraints for noun modifiers and verbs often indicate semantic properties of the noun target. For example, the selectional constraints of the adjective *careful* can be *human*, *animal*, or some *abstract entities*, but can’t be *location* or *currency*. Similarly, the selectional constraints for the subject of *discovered* are *animate* or *organization*, based on VerbNet. Erk [21] concludes that if two noun targets agree in many of their modifiers and always occur in the same argument positions of the same verbs, they will have high distributional similarity and share many semantic properties. In the experiment part, we will undertake a further empirical analysis about the effect of context types on the property inference from distributional models.

Semantic property norms have been used to explore and enhance many aspects of the semantic representation and the processing of concepts in cognitive science. The semantic property norms described in [4] are widely used. The dataset is collected from approximately 725 participants for 541 living (alligator) and nonliving (airplane) basic-level concepts. Each named concept corresponds to an English noun, which is normed by 30 participants through a questionnaire. The 541 concepts are annotated by a total of 2526 properties. It is a very sparse dataset in which each concept has an average of 13 properties. The concepts with multiple meanings are disambiguated by providing some cues; for example, *bat* is represented by two forms *bat_(animal)* and *bat_(baseball)*. Table 2 lists the top five properties of the concept for *basket* in the McRae dataset and the number of subjects out of 30 participants that listed a property (known as production frequencies).

Table 2. Top 5 properties for the concept *basket* in McRae and their production frequencies.

Concept	Top 5 Properties	Production Frequency
basket	is_weaved	18
	used_for_holding_things	18
	has_a_handle	17
	made_of_wicker	10
	made_of_wood	9

4. Learning Properties of Concepts from Distributional Models with Multilayer Perceptrons

We model the problem of learning the property-based representation of a concept as a multinomial classification problem. Let \mathcal{X} denote the distributional semantic space and \mathcal{Y} denote the property semantic space. Then we learn a function

$$\Phi : \mathcal{X} \rightarrow \mathcal{Y}, \quad (1)$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^m$. Given a test concept $x \in \mathbb{R}^n$, we want to estimate an m -dimensional vector (whose elements sum to 1), which represents the probabilities for m property classes. Existing works [16,18] estimate the function using the PLSR method, which models the relationship between two spaces as a linear function. But linearity is a strong assumption. Multilayer perceptrons (MLPs) can learn very complicated relationships between their inputs and outputs because nonlinear transformations are introduced to the hidden layers of the architecture.

It is well known that MLPs are universal approximators. The universal approximation theorem states that the standard multilayer feed-forward networks with a single hidden layer that contains sufficient hidden neurons and with an arbitrary activation function can approximate any function in $\mathcal{C}(\mathbb{R}^n)$ [29,30]. However, it is difficult to learn that function because a super wide and shallow network is very good at memorization but not good at generalization. Multiple hidden layers are much better at generalizing because they can learn features at various levels of abstraction. Given a collection of input data x and corresponding target values y , our MLP model expresses the prediction \hat{y} through three hidden layers on top of each other.

$$\begin{aligned} h^1 &= \phi(W^1 x + b^1), \\ h^2 &= \phi(W^2 h^1 + b^2), \\ h^3 &= \phi(W^3 h^2 + b^3), \end{aligned} \quad (2)$$

where ϕ is any activation function and W^l , and b^l are the weights and biases at layer l (for $l \in \{1, 2, 3\}$). In our experiment, we use the rectified linear unit (ReLU), $\phi(z) = \max(0, z)$. Since our problem is modeled as multinomial classification, the probabilities for class labels are available through the use of the softmax activation function in the output layer.

$$\hat{y} = \text{softmax}(W^4 h^3 + b^4), \quad (3)$$

where the softmax function squashes the outputs of each unit to be in $[0, 1]$, and the total sum of the outputs is equal to 1.

When a large feed-forward neural network is trained on a small training set, it typically performs well on the training set but poorly on real test data, which leads to overfitting. Dropout is a form of regularization for addressing this problem that is developed by [31,32]. Dropout combats overfitting by randomly deactivating hidden nodes with probability p (commonly be set at 0.5) during training time. Training a neural network with dropout can be seen as training a collection of thinned networks with extensive weight sharing, while at test time these thinned networks with shared weights can be combined into a single neural network to be used. In our MLP model, dropout is applied to hidden units in the hidden layers with a probability of 0.5. With dropout, our MLP model becomes (for $l \in \{1, 2\}$)

$$\begin{aligned} r^l &\sim \text{Bernoulli}(p), \\ \tilde{h}^l &= r^l * h^l, \\ h^{l+1} &= \phi(W^{l+1} \tilde{h}^l + b^{l+1}), \\ \hat{y} &= \text{softmax}(W^4 \tilde{h}^3 + b^4), \end{aligned} \quad (4)$$

where $*$ denotes an element-wise product, and r^l is a vector of independent Bernoulli random variables, each of which takes the value 1 with probability p and the value 0 with probability $1 - p$. A more detailed description of dropout can be found in [32].

To measure how close our predicted value \hat{y} is to the true value y , categorical crossentropy is used as a loss function of our model. We learn values for W^l and b^l that minimize the loss function by the Adam optimization algorithm, which is computationally efficient and works well in practice [33]. Table 3 displays a summary of several parameters chosen for our MLP model.

Table 3. A summary of several parameters chosen for our multilayer perceptron (MLP) model.

Description	Value
Model Type	Multilayer Perceptron (MLP)
Number of Layers	5 (3 hidden layers)
Number of Units in Each Layer	300:800:1500:2000:2526
Hidden Layer Activation Function	ReLU
Output Layer Activation Function	Softmax
Regularization	Dropout
Loss Function	Categorical Cross Entropy
Optimizer	Adam

5. Experiments and Evaluation

5.1. Data Preparation

The experiments use 300-dimensional W2V and DEP vectors that have been pretrained on diverse English corpora: (1) English Wikipedia 2015, (2) UMBC web corpus [34], and (3) English Gigaword newswire corpus [35]. The concatenated corpus comprises about 10B words and yields a vocabulary of about 500K words after filtering words with a frequency lower than 100. The word embeddings W2V1, W2V5, and W2V10 are learned with context window sizes of 1, 5, and 10 respectively. For learning DEP, the corpus is tagged with parts-of-speech and parsed into Stanford dependencies by CoreNLP [36]. Those vectors are described in [28] and available for downloading (<http://u.cs.biu.ac.il/~nlp/resources/downloads/embeddings-contexts/>).

The McRae-based property norm space contains 541 concepts as described in Section 3. We delete 3 concepts (*axe*, *armour*, and *dunebuggy*) from the McRae dataset because they are not available in the word2vec vocabulary. The dataset is very sparse, with a total of 2526 properties for 538 concepts and an average of 13 properties per concept. For the concepts with multiple meanings in McRae, these are disambiguated by providing some cue, for example, *bat_(animal)* and *bat_(baseball)*. In distributional semantic space, their vector representations are also different by computing the average vectors of the concept and its cues, respectively. The 538 concepts are split randomly into 400 training data and 138 test data. There are 437 out of 2526 properties that are not seen in the training set. We normalize each property vector by the sum of its production frequencies at training time. Our goal is to learn a mapping from distributional semantic space (400×300) to property norm space (400×2526) and then predict property norms for 138 new concepts from their distributional semantic representation.

5.2. Baselines

We compare our MLP model with two baselines: partial least squares regression (PLSR) and k-nearest neighbor (KNN).

- PLSR: refs. [16,18] estimate the function (1) in Section 4 using PLSR. PLSR, a bilinear factor model, is used to find the fundamental relations between the distributional semantic space \mathcal{X} and the property semantic space \mathcal{Y} by modeling the covariance structures in these two spaces. In our experiments, we set the number of components to 50 and 100, respectively.
- KNN: For each concept in the test dataset, we choose the k most similar concepts from the training dataset based on their cosine similarities in the distributional semantic space. Then the property vector of the concept in the test dataset is represented by the average of property vectors of these k most similar concepts. We set k to 5 and 10 in our experiments.

5.3. Quantitative Evaluation

In the following experiments, we use the properties in McRae as the gold standard that models are compared against. We study the properties of 138 test concepts from McRae, in which each test concept has 13.39 properties on average. For each test concept, we rank the properties from the predicted

property vector in terms of their values and pick the top 10 properties from the list. Given the top 10 ranked properties generated by different methods from different distributional models, precision and recall are used for evaluation. The definitions of two standard performance measurements at the 10th property in the ranked list are specified as follows.

Precision is the fraction of the predicted properties that are correct, i.e.,

$$P = \frac{\text{number of correct properties}}{\text{total number of properties returned}}.$$

Recall is the fraction of the properties that have been predicted correctly, i.e.,

$$R = \frac{\text{number of correct properties}}{\text{total number of gold standard properties}}.$$

Tables 4 and 5 report percentage average precisions and recalls across 138 test concepts by each method matched against the McRae gold standard. Four kinds of word embeddings, W2V1, W2V5, W2V10, and DEP are used in the experiments. From the average precision and recall, we see that MLP with dropout beats all other methods. We can obtain the following order of the methods: MLP >> PLSR >> KNN, where >> indicates that the method on the left-hand side is significantly better than the one on the right-hand side. The better experimental results of our method further illuminate the advantage of using MLP; namely, the method has the ability to learn and model complex nonlinear relationships between its inputs and outputs. Looking further, we observe that there are significant differences in the results when using MLP with and without dropout. The same architecture (300:800:1500:2000:2526) trained with dropout gives evident improvements across all distributional models on the McRae dataset over that without dropout. For the PLSR method, the performances of PLSR ($c = 50$) are comparable to those of PLSR ($c = 100$). KNN with $k = 10$ achieves better performances than that with $k = 5$ for all distributional models.

Table 4. The percentage average precisions for all methods.

Methods	Parameters	Distributional Models (Precision)			
		W2V1	W2V5	W2V10	DEP
MLP	Dropout	42.46	42.10	41.15	41.59
	No Dropout	40.72	40.65	40.86	40.79
PLSR	$c = 50$	38.26	37.75	37.75	37.53
	$c = 100$	37.31	38.26	38.18	37.68
KNN	$k = 5$	36.01	34.63	34.85	35.79
	$k = 10$	36.01	35.86	36.37	36.66

Table 5. The percentage average recalls for all methods.

Methods	Parameters	Distributional Models (Recall)			
		W2V1	W2V5	W2V10	DEP
MLP	Dropout	32.40	32.06	31.28	31.95
	No Dropout	30.90	30.86	31.27	31.11
PLSR	$c = 50$	29.04	28.43	28.72	29.01
	$c = 100$	28.60	29.06	29.12	28.98
KNN	$k = 5$	27.76	26.47	26.74	27.26
	$k = 10$	27.89	27.73	27.95	28.08

Comparing the results of different types of contexts, we see that MLP with dropout, PLSR ($c = 50$), and KNN ($k = 5$) all perform much better on W2V1 than on other distributional models. For KNN with $k = 10$, DEP does the best job on the precision and recall performances. These results are as

we expected, as narrow context window or syntactic context window models tend to make better property inferences than wide context models. However, it is difficult to say that narrow context or syntactic context models always works best for all methods. We found that for MLP without dropout, W2V10 outperforms W2V1, W2V5, and DEP on both recall and precision.

5.4. Qualitative Evaluation

Table 6 reports the top five predicted properties returned by MLP (dropout), PLSR ($c = 50$), and KNN ($k = 10$) based on W2V1 for eight test concepts. We also list their top five gold-standard properties annotated by participants in McRae. From Table 6, we see that MLP can provide mostly reasonable properties for these eight concepts. Properties annotated with * in the table are not listed in McRae. Although we refer to the property norms in McRae as the “gold standard”, these annotated properties are sometimes not completely true representations of concepts because the annotation depends on the knowledge background or linguistic habits of participants. A property with zero production frequency for a concept in McRae shouldn’t be read as a sign that it is not a reasonable property of the concept. It simply means that the property is not elicited from the conceptual knowledge of participants during the questionnaire. For example, the property *has_a_tail* is not listed in the gold-standard vector of *walrus* in McRae, but it is a plausible property of *walrus*.

Table 6. Top 5 properties returned by MLP, PLSR, and KNN. Properties annotated with * are not listed in McRae.

Concept	Method	Top 5 Predicted Properties
jar	McRae	has_a_lid, made_of_glass, used_for_holding_things, a_container, is_breakable,
	MLP	used_for_holding_things, made_of_plastic, made_of_glass, has_a_lid, is_breakable
	PLSR	made_of_plastic, *found_in_kitchens, *made_of_metal, *is_round, used_for_holding_things
	KNN	*made_of_metal, made_of_plastic, *found_in_kitchens, used_for_holding_things, *a_utility
sparrow	McRae	a_baby_bird, beh_flies, has_feathers, beh_lays_eggs, has_wings
	MLP	a_baby_bird, beh_flies, has_wings, has_feathers, beh_lays_eggs
	PLSR	a_baby_bird, beh_flies, has_feathers, has_wings, has_a_beak
	KNN	a_baby_bird, beh_flies, has_feathers, has_wings, has_a_beak
spatula	McRae	a_utility, has_a_handle, made_of_plastic, used_for_cooking, is_flat,
	MLP	a_utility, made_of_plastic, made_of_metal, found_in_kitchens, *used_for_eating
	PLSR	made_of_metal, found_in_kitchens, made_of_plastic, *a_tool, a_utility
	KNN	made_of_metal, found_in_kitchens, *a_tool, made_of_plastic, has_a_handle
sofa	McRae	found_in_living_rooms, furniture, is_comfortable, used_by_sitting_on, has_cushions
	MLP	is_comfortable, furniture, is_soft, used_by_sitting_on, has_cushions
	PLSR	is_comfortable, *made_of_wood, is_soft, used_by_sitting_on, used_for_sleeping
	KNN	is_comfortable, is_soft, *made_of_wood, *worn_for_warmth, used_for_sleeping
bracelet	McRae	worn_on_wrists, made_of_gold, made_of_silver, a_fashion_accessory, a_jewelry
	MLP	*worn_around_neck, made_of_silver, a_fashion_accessory, a_jewelry, made_of_gold
	PLSR	made_of_metal, made_of_gold, made_of_silver, *worn_around_neck, is_round,
	KNN	*worn_for_warmth, *clothing, *is_long, *worn_by_women, *different_colours
doll	McRae	has_own_clothes, used_for_playing, a_toy, used_by_girls, has_hair
	MLP	*is_small, *is_soft, *is_white, *is_comfortable, *is_large
	PLSR	*is_small, *is_white, *worn_by_women, *different_colours, *clothing
	KNN	*is_comfortable, *worn_for_warmth, *worn_at_night, *is_warm, *clothing
walrus	McRae	an_animal, is_large, beh_swims, lives_in_water, is_fat
	MLP	an_animal, a_mammal, hunted_by_people, *has_a_tail, beh_swims
	PLSR	an_animal, is_large, lives_in_water, beh_swims, *a_baby_bird
	KNN	an_animal, *has_a_tail, has_teeth, *is_green, *is_furry
platypus	McRae	an_animal, lives_in_water, a_mammal, beh_swims, has_a_bill
	MLP	an_animal, beh_swims, *has_a_tail, *has_4_legs, *is_brown
	PLSR	an_animal, *is_small, *has_a_tail, *has_4_legs, *is_large
	KNN	an_animal, *is_green, *has_4_legs, *beh_eats, *has_a_tail

For some concepts, like *sparrow*, all three methods can detect gold-standard properties in their top five lists. However, looking further, we observe that the top five properties returned by MLP

are consistent with the top five gold standard properties annotated by McRae, whereas the property *has_a_beak* described by PLSR and KNN is not in the top five properties of McRae. Moreover, the properties that are not seen in the training set will have no opportunity to be ranked as the top properties of a concept. For example, all three methods cannot detect *has_own_clothes* and *used_by_girls* for the *doll* concept in their top property lists because these two properties don't appear in the training set.

To further evaluate the quality of property inference, we perform a nearest neighbor search for the predicted vector of a concept. The predicted vectors are produced by MLP (dropout), PLSR ($c = 50$), and KNN ($k = 10$) based on W2V1. The ideal performance is that the predicted vector of a concept should be close to its gold standard vector in McRae [16]. Table 7 shows the top five neighbors of the predicted vectors among the 138 gold standard property vectors for the eight concepts above. The results look promising, where five out of eight gold standard vectors are the 1-nearest neighbor to their predicted vectors based on the MLP method.

Table 7. Top 5 neighbors returned by MLP, PLSR, and KNN.

Concept	Method	Top 5 Neighbors
jar	MLP	jar, bucket, plate, spatula, whistle
	PLSR	bucket, spatula, plate, pan, skillet
	KNN	spatula, tongs, bucket, grater, pan
sparrow	MLP	sparrow, raven, finch, buzzard, parakeet
	PLSR	raven, sparrow, finch, buzzard, parakeet
	KNN	sparrow, raven, finch, buzzard, parakeet
spatula	MLP	spatula, fork, tongs, grater, bucket
	PLSR	spatula, tongs, grater, pan, hatchet
	KNN	spatula, tongs, hatchet, grater, bucket
sofa	MLP	sofa, cushion, bench, jeans, cabinet
	PLSR	sofa, cushion, cabinet, bench, jeans
	KNN	sofa, cushion, socks, bench, cabinet
bracelet	MLP	bracelet, tie, crown, fork, plate
	PLSR	bracelet, tongs, bucket, crown, thimble
	KNN	skirt, socks, cape, tie, jacket
doll	MLP	rice, cottage, cushion, shrimp, bear
	PLSR	sparrow, finch, butterfly, sheep, raven
	KNN	socks, bracelet, cape, skirt, bench
walrus	MLP	platypus, buffalo, elk, walrus, caribou
	PLSR	walrus, ox, buffalo, platypus, otter
	KNN	walrus, ox, platypus, otter, cougar
platypus	MLP	otter, platypus, walrus, ox, buffalo
	PLSR	ox, walrus, platypus, buffalo, elk
	KNN	cougar, ox, buffalo, elk, walrus

6. Conclusions

This work presents a multilayer perceptrons-based method for mapping a distributional semantic space onto a human-built property space automatically. The quantitative evaluation implied that using MLP with dropout can give significantly better performance on property inference than models using PLSR or k-nearest neighbors. The results of using vectors obtained from various kinds of contexts were more mixed and depended on what parameters were used. This may indicate that dependency-based windows are missing some important context from nearby words that don't have the particular dependencies such models focus on. Based on the qualitative analysis, we found that compared with PLSR and KNN, MLP can not only detect gold standard properties but also give a reasonable ranking for returned properties. In future work, we will conduct further analyses of

prediction performance on different types of properties. We are also interested in exploring how to generalize property-based representations to a large-scale dataset.

Author Contributions: Methodology, D.L. and D.S.-S.; writing—original draft preparation, D.L.; writing—review and editing, D.S.-S.

Funding: This research was funded by U.S. Army Research Laboratory grant number W911NF-16-2-0102.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
2. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
3. Levy, O.; Goldberg, Y. Dependency-based word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), Baltimore, MD, USA, 23–25 June 2014; pp. 302–308.
4. McRae, K.; Cree, G.S.; Seidenberg, M.S.; McNorgan, C. Semantic feature production norms for a large set of living and nonliving things. *Behav. Res. Methods* **2005**, *37*, 547–559. [[CrossRef](#)] [[PubMed](#)]
5. Devereux, B.J.; Tyler, L.K.; Geertzen, J.; Randall, B. The Centre for Speech, Language and the Brain (CSLB) concept property norms. *Behav. Res. Methods* **2014**, *46*, 1119–1127. [[CrossRef](#)] [[PubMed](#)]
6. Gupta, A.; Boleda, G.; Baroni, M.; Padó, S. Distributional vectors encode referential attributes. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 12–21.
7. Baroni, M.; Murphy, B.; Barbu, E.; Poesio, M. Strudel: A Corpus-Based Semantic Model Based on Properties and Types. *Cogn. Sci.* **2010**, *34*, 222–254. [[CrossRef](#)] [[PubMed](#)]
8. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162. [[CrossRef](#)]
9. Firth, J.R. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*; Longmans: London, UK, 1957; pp. 1–32.
10. Murphy, G. *The Big Book of Concepts*; The MIT Press: Cambridge, MA, USA, 2002.
11. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. In Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, Suntec, Singapore, 2–7 August 2009; pp. 1003–1011.
12. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning with neural tensor networks for knowledge base completion. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 926–934.
13. Speer, R.; Chin, J.; Havasi, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017; pp. 4444–4451.
14. Devereux, B.; Pilkington, N.; Poibeau, T.; Korhonen, A. Towards Unrestricted, Large-Scale Acquisition of Feature-Based Conceptual Representations from Corpus Data. *Res. Lang. Comput.* **2009**, *7*, 137–170. [[CrossRef](#)]
15. Summers-Stay, D.; Voss, C.; Cassidy, T. Using a Distributional Semantic Vector Space with a Knowledge Base for Reasoning in Uncertain Conditions. *Biol. Inspired Cogn. Archit.* **2016**, *16*, 34–44. [[CrossRef](#)]
16. Herbelot, A.; Vecchi, E.M. Building a shared world: Mapping distributional to model-theoretic semantic spaces. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (Best Paper Award), Lisbon, Portugal, 17–21 September 2015; pp. 22–32.
17. Dernoncourt, F. Mapping distributional to model-theoretic semantic spaces: A baseline. *arXiv* **2016**, arXiv:1607.02802.
18. Făgărășan, L.; Vecchi, E.M.; Clark, S. From distributional semantics to feature norms: Grounding semantic models in human perceptual data. In Proceedings of the 11th International Conference on Computational Semantics, London, UK, 15–17 April 2015; pp. 52–57.

19. Baroni, M.; Lenci, A. Concepts and Properties in Word Spaces. *Ital. J. Linguist.* **2008**, *9*, 1–36.
20. Rubinstein, D.; Levi, E.; Schwartz, R.; Rappoport, A. How Well Do Distributional Models Capture Different Types of Semantic Knowledge? In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 726–730.
21. Erk, K. What do you know about an alligator when you know the company it keeps? *Semant. Pragmat.* **2016**, *9*, 1–63. [[CrossRef](#)]
22. Boleda, G.; Herbelot, A. Formal Distributional Semantics: Introduction to the Special Issue. *Spec. Issue Comput. Linguist. Form. Distrib. Semant.* **2017**, *42*, 619–635. [[CrossRef](#)]
23. Bulat, L.; Kiela, D.; Clark, S. Vision and Feature Norms: Improving automatic feature norm learning through cross-modal maps. In Proceedings of the NAACL-HLT, San Diego, CA, USA, 12–17 June 2016; pp. 579–588.
24. Frome, A.; Corrado, G.S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; Mikolov, T. DeViSE: A Deep Visual-Semantic Embedding Model. In Proceedings of the NIPS 2013: Neural Information Processing Systems Conference, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2121–2129.
25. Silberger, C.; Ferrari, V.; Lapata, M. Visually Grounded Meaning Representations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2284–2297. [[CrossRef](#)] [[PubMed](#)]
26. Zellers, R.; Choi, Y. Zero-Shot Activity Recognition with Verb Attribute Induction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 946–958.
27. Peirsman, Y. Word Space Models of Semantic Similarity and Relatedness. In Proceedings of the 13th ESSLLI Student Session, Sofia, Bulgaria, 6–17 August 2008; pp. 143–152.
28. Melamud, O.; McClosky, D.; Patwardhan, S.; Bansal, M. The Role of Context Types and Dimensionality in Learning Word Embeddings. In Proceedings of the NAACL-HLT, San Diego, CA, USA, 12–17 June 2016; pp. 1030–1040.
29. Cybenko, G. Approximation by Superpositions of a Sigmoidal Function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [[CrossRef](#)]
30. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
31. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
32. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
33. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
34. Han, L.; Kashyap, A.L.; Finin, T.; Mayfield, J.; Weese, J. UMBC EBIQUITY-CORE: Semantic textual similarity systems. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics, Atlanta, GA, USA, 13–14 June 2013.
35. Parker, R.; Graff, D.; Kong, J.; Chen, K.; Maeda, K. *English Gigaword*, 5th ed.; Linguistic Data Consortium, LDC2011T07; Linguistic Data Consortium: Philadelphia, PA, USA, 2011.
36. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014; pp. 55–60.

