Natural Language Engineering (2023), **29**, pp. 201–227
doi:10.1017/S1351324921000267

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

An analysis of property inference methods

Alex Rosenfeld[1]*† and Katrin Erk[2]

[1]Intelligent Automation, Inc., Rockville, MD 20855, USA and [2]Department of Linguistics, The University of Texas at Austin, Austin, TX 78705, USA
*Corresponding author. E-mail: alexbrosenfeld@gmail.com

(Received 29 January 2019; revised 15 June 2021; accepted 21 June 2021; first published online 14 January 2022)

Abstract

Property inference involves predicting properties for a word from its distributional representation. We focus on human-generated resources that link words to their properties and on the task of predicting these properties for unseen words. We introduce the use of label propagation, a semi-supervised machine learning approach, for this task and, in the first systematic study of models for this task, find that label propagation achieves state-of-the-art results. For more variety in the kinds of properties tested, we introduce two new property datasets.

Keywords: Semantics; Machine learning; Lexical knowledge acquisition

1. Introduction

There is a large body of research that demonstrates that distributional vectors are a powerful tool for modeling word similarity (Turney *et al*. 2010; Erk 2012; Clark 2015). For many tasks, the knowledge of word similarity alone is all that is needed; other tasks require some form of "grounding," a mapping from textual distributions to some other representation, such as a visual vector (Feng and Lapata 2010; Bruni *et al*. 2012; Lazaridou, Bruni, and Baroni 2014) or a distribution over geographic locations (Wing and Baldridge 2011). In cognitive science, word vectors are mapped to semantic primitives to explore perception of concepts. Through studying patients with category-specific deficits, several cognitive scientists have proposed that people mentally represent concepts as a collection of semantic primitives or feature norms (Tyler *et al*. 2000; Randall *et al*. 2004). As word vectors are connected with word usage, researchers have mapped word vectors to these primitives to explore the connection between usage and concept (Johns and Jones 2012). Other research has explored mappings between word vectors and different kinds of semantic primitives (or properties) to explore what information is conveyed through word use (Johns and Jones 2012; Rubinstein *et al*. 2015; Herbelot and Vecchi 2015; Fagarasan, Vecchi, and Clark 2015; Gupta *et al*. 2015).

We focus on the latter kind of mapping: the task of *property inference*, the prediction of properties for a word based on its distributional vector. In other words, connecting word usage (via word vectors) to real-world characteristics. We present a toy example of this task in Figure 1. In this example, we try to predict the properties of the word *dog* using what we know about the words *cat* and *truck*. We associate various cat properties to the word *cat*, for example, is a pet, has four legs, has claws, etc. We analogously associate various truck properties to the word *truck*. For the word *dog*, we look to similarity in usage (where usage is modeled as a word vector) to predict its

†Research performed while attending The University of Texas at Austin.

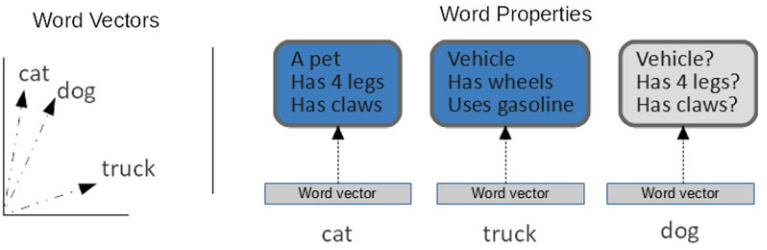

Figure 1. Property inference toy example.

properties. As the word *dog* is used much more similarly to the word *cat* than the word *truck*, we infer that *dog* must have more cat-like properties, for example, is a pet, has claws, is an animal, meows, etc., than truck-like properties.

Property inference has mainly been addressed with the aim of better understanding the linguistic information that distributional vectors contain. For example, do word vectors reflect taxonomic properties of the word? Social significance of the word? Physical properties of the object represented by the word? Rubinstein *et al.* (2015) used property inference to examine if distributional vectors express more taxonomic or attributive properties. Herbelot and Vecchi (2015, 2016) used this task to analyze how well distributional vectors capture the proportion of category members to which properties apply. Făgărăsan *et al.* (2015) explored how well distributional vectors reflect definitional conceptual properties. Gupta *et al.* (2015) explored how well distributional vectors reflect geographic and demographic properties. While we explore properties as a way to extract insight from word vectors, other research explores generating definitions of a word from its word vector as an alternative way to describe the semantic content of a word (Noraset *et al.* 2017).

Beyond an exploration of the characteristics of distributional models, property inference can in principle be useful to many tasks within computational linguistics. Inferred properties can provide partial meaning information for unknown words, and it can assist in information extraction by allowing a better description of entities through their properties.

But if we want to use property inference as a basis for further inference tasks, it is important to know which methods perform best on what kind of data. However, the existing papers mentioned above use completely different property inference methods with no comparison between them. They also use different property collections.

In this paper, we perform the first systematic comparison of existing approaches to property inference. We introduce the use of label propagation for the task, which is a semi-supervised machine learning approach that infers the target value of an input by aggregating the target values of similar inputs. We show that it is well suited for the task and that achieves state-of-the-art performance. We further propose modifications of existing methods that are based on known characteristics of distributional models and that we find to improve performance.

We also introduce two new property datasets to the property inference task. Most property inference research has tried to predict properties that are salient to humans as definitional properties of objects. We expand the scope of the task by introducing two new datasets, one that has properties encoding social categories and one that focuses on hypernyms as properties. By analyzing a variety of property datasets, we explore the effect of differences in property datasets on model performance.

The results of our work can be used for several downstream tasks. One downstream task is to make machine learning models more interpretable. We live in a day and age where accurate machine learning models are readily available and are entrenched in our daily life. Companies use machine learning models to interact with customers, analyze documents, and gauge public reaction. However, these models often do not provide explanations for their predictions. This

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 1. Reference table for previous work. "Properties" are the property datasets used. "Metrics" are the quantitative approach used to evaluate the results. "Correlation" refers to correlating gold values with predicted values (using either Pearson correlation or Spearman's ρ). "Accuracy" refers to treating property inference as a binary classification task (using either F1 or accuracy). "Ranking" refers to using a ranking metric like MAP or normalized rank score

Reference	Method	Properties	Metric
Johns and Jones (2012)	JJ1/JJ2	McRae, VV	Correlation
Fagarasan *et al*. (2015)	PLS	McRae	Ranking
Rubinstein *et al*. (2015)	linear SVM	McRae	Accuracy/correlation
Herbelot and Vecchi (2015, 2016)	PLS	Modified McRae Animal Dataset	Correlation
Gupta *et al*. (2015)	Logis. Reg.	FreeBase	Accuracy/Ranking
Derby, Miller, and Devereux (2019)	Feature2Vec	McRae/CSLB	Recall@K

could be problematic as it is important to understand *why* a model rejected a loan application or *why* a model determined that public reaction was negative. Our work can be used to provide such justification by translating hidden layers and other components of machine learning models into sets of human-interpretable properties.

Our work can also be used to better generate embeddings that capture semantic qualities. Turton, Vinson, and Smith (2020) use property inference methods to predict properties from word embeddings and then use these predicted properties as a new embedding representation. By incorporating stronger property inference methods, we can produce embeddings that better capture semantic qualities. These embeddings can then be used to make strides in other tasks, such as lexical entailment (Vulić and Mrkšić 2017).

Finally, our work can help cognitive scientists generate sets of feature norms. Feature norms are widely used by cognitive scientists to model cognitive processes (such as word-picture interference Vieth, McMahon and de Zubicaray 2014 and recognition memory Montefinese, Zannino and Ambrosini 2015) as well as study neurological disorders like aphasia (Vinson and Vigliocco 2002; Vinson *et al*. 2003). However, feature norms are difficult to generate as they require eliciting information from participants and then normalizing this information. Our work can be used to expand feature norm datasets.

2. Related work

We focus on property inference based on manually constructed resources that link concepts[a] to their properties. Like previous work, we focus on weighted binary properties: each property either does or does not apply (like "an animal" or "is green"), but it is associated with a value that indicates its importance to the concept. We make this more concrete later when we introduce the property datasets.

We study approaches that learn a mapping from distributional representations to property values and then use this mapping to infer properties for concepts that have distributional representations but no known properties. As we describe next, existing work on property inference has used a wide variety of methods. Table 1 provides a brief description of these methods. But there has been no comparison across them. In this paper, we explore which methods work best in this task, and under what circumstances.

[a] Here and below, we call the words that are associated with properties *concepts*.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 2. McRae *et al*. feature norms for the concept *knife*. # is the number of participants out of 30 who responded that *knife* had that feature norm

Feature norm	#
is dangerous	14
found in kitchens	18
used with forks	16
a weapon	11
a utensil	19
a cutlery	15

The oldest and simplest method that we explore is that of Johns and Jones (2012). Their goal was to formally model the psychological process of how a person can learn word meaning from context. Their model is based on Hintzman (1986, 1988), who posits that people learn word meaning by transferring properties from familiar words to unfamiliar words that fit the same contexts.

Johns and Jones' model predicts the value of a property for a novel concept simply as a weighted sum of values that familiar concepts have for this property, weighted by the distributional similarity of each familiar concept to the novel one. Let F be a set of familiar concepts, and e a novel concept. For property q and concept $c \in F$, let $v_q(c)$ be the value of property q for concept c and let $\vec{c}$ be the distributional vector for c. Then the predicted value of q for e, $v_q(e)$, is

$$v_q(e) = \sum_{c \in F} v_q(c) \cos(\vec{c}, \vec{e})^{\lambda_1}$$

where λ_1 is a hyperparameter. A large value of λ_1 means that the influence of less similar concepts c is largely ignored. We call this method **JJ1** for one-step Johns and Jones. They also propose a two-step version of their method (**JJ2**) that first extends property annotation to a large set U of un-annotated concepts using JJ1, then predicts the value of a property q for the novel concept e from similarity to concepts in both F and U:

$$v_q(e) = \sum_{c \in F} v_q(c) \cos(\vec{c}, \vec{e})^{\lambda_2} + \sum_{u \in U} v_q(u) \cos(\vec{u}, \vec{e})^{\lambda_2}$$

where λ_2 is the hyperparameter for the second step.

The property datasets that Johns and Jones used were two feature norm datasets, one by McRae *et al*. (2005) and one by Vinson and Vigliocco (2008). Feature norms are definitional features elicited from human participants. As we also use these datasets below, we look at them in some more detail. The McRae et al. feature norms focus on concepts for concrete objects. Vinson and Vigliocco additionally have concepts for nominal and verbal forms of events. Examples of the McRae et al. and Vinson–Vigliocco feature norms are in Tables 2 and 3.

While the motivation for Johns and Jones was to explain human concept learning, the remaining approaches come from computational linguistics. Făgărăsan *et al*. (2015) mapped a distributional vector to a feature norm vector using partial least squares regression (**PLS**). PLS is useful when there is a high degree of correlation between predictor variables (here, dimensions of distributional vectors) and between response variables (here, properties). PLS projects the predictor variables and the response variables to a new space in a way that takes into account the relations between the two sets of variables (Ng 2013). PLS is trained on a predictor matrix X and

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 3. Vinson–Vigliocco feature norms for the verb *chirp*. # is the number of participants out of 30 who responded that the verb *chirp* had that feature norm

Feature norm	#	Feature norm	#
bird	20	talk	2
noise	17	summer	1
high-pitched	9	humans	1
communicate	8	fast	1
sound	7	unpleasant	1
short	4	feather	1
song	3	pleasant	1
sweet	2	annoy	1
music	2	intentional	1
beak	2	soft	1
action	2	wing	1

Table 4. Dataset statistics for the property datasets. "# Words" is the number of words in the datasets. "# Properties" is the number of unique properties in the dataset. "Props per words" is the average number of properties per word

Dataset	Source	# Words	# Properties	Props per word
McRae	McRae *et al*. feature norms	491	2526	13.46
VV	Vinson–Vigliocco feature norms	446	1029	28.36
GenInqMS	General Inquirer	8488	181	4.64
WN-Hyp	WordNet	4161	4431	10.75

a response matrix Y. The two matrices are decomposed into $X = TP^{\intercal}$ and $Y = UQ^{\intercal}$ such that the covariance between T and U is maximized. P and Q are orthogonal. Linear regression is performed from T to U to find a β such that $U = T\beta$. The resulting model can then be used to predict the property vector $f(x)$ for a distributional vector x using $f(x) = xP\beta Q^{\intercal}$.

Evaluating on the McRae *et al*. feature norms, Făgărăsan *et al*. found that a two-word window space with positive pointwise mutual information (PPMI) and singular value decomposition (SVD) dimensionality reduction performed best under a mean average precision (MAP) evaluation.

Rubinstein *et al*. (2015) used property inference to determine if word vectors reflect more taxonomic or attributive properties. They used linear support-vector machine (SVM) classification and regression to predict a small subset of the McRae *et al*. feature norms from word vectors. Like SVM classification, SVM regression maps the data into a kernel-induced space (Drucker *et al*. 1996). It then performs linear regression in that space. Also, like SVM classification, SVM regression depends only on a subset of the training data, as it ignores training data that is close to the prediction.

SVM regression learns a function $g{:}\mathbf{R}^n \rightarrow \mathbf{R}^n$ with training data $(x_1, y_1), \ldots, (x_n, y_n)$ that is a linear function of the similarities between x and the training points x_i (modeled by a kernel

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

function $K(x, x_i)$). It has the form:

$$g(x) = \sum_{i=1}^{N} a_i K(x, x_i) + b$$

The a_i represent the contribution of sample i in the predicted property value. The b is a constant expressing bias. Learning of the function g is constrained by a hyperparameter ϵ that governs how far the predicted $g(x_i)$ is allowed to deviate from the gold y_i. Learning of g is further constrained by the second hyperparameter, a regularization constant C that constrains the values that the a_i are allowed to take on. Rubinstein et al. used a linear kernel in their experiments.

Herbelot and Vecchi (2015, 2016) used PLS on a variant of the McRae *et al*. feature norms where annotators rated how many members of a given concept have a given property. We note that they also incorporate an animal dataset (Herbelot 2013), which maps 72 animal concepts to a fixed set of 54 properties. The goal is to create a dense property dataset where each concept has some value for each property. As our focus is on sparse properties, we do not explore this dataset in our analysis.

Gupta *et al*. (2015) used logistic regression to learn referential properties, such as longitude and latitude, of cities and countries from distributional vector representations. The property dataset they use is derived from Freebase (Bollacker *et al*. 2008), a former online database that contained structured information on entities (geolocation, member-of relations, etc.). Gupta *et al*.'s property dataset represents properties as either a numeric quantity, for example, *Germany* has the property "geolocation::latitude::52.52," or as a categorical value, for example, *Germany* has the property "adjectival form::German." Given that the nature of these properties are more akin to filling a slot rather than atomic features, we do not use this dataset in our task.

Derby *et al*.[b] (2019) developed a property prediction method, Feature2Vec, based on property embeddings. Properties were embedded into a word vector space and comparison between a word vector and property embedding was used to predict if that word has that property. They evaluated their method on the McRae feature norms and the Centre for Speech, Language and the Brain concept property norms (Devereux *et al*. 2014), which replicate and expand the McRae feature norms. For property prediction, the results for their model was in the same ballpark as PLS.

We should also mention another strand of research that aimed to learn properties from textual data but used patterns to extract properties directly; these methods have met somewhat mixed success (Almuhareb and Poesio 2004; Devereux *et al*. 2009; Baroni and Lenci 2010).

3. Properties

We chose four property datasets for our experiments that express different types of semantic information and have different characteristics. The first two are feature norm datasets (McRae *et al*. 2005 and Vinson and Vigliocco 2008) that are small in size and are constructed from elicitation by subjects. The McRae feature norms capture semantic representations of concrete objects while the Vinson–Vigliocco feature norms explore the connections between objects and events. In particular, the Vinson–Vigliocco feature norms distinguish between concrete objects (e.g., a bear), nouns related to events (e.g., a growl), and verbs related to events (e.g., growling). The next one is derived from General Inquirer (GI) (Stone 1997). This dataset contains a larger vocabulary and places words within a relatively small set of categories that captures sentiment and social significance information. Finally, we derive a property dataset from WordNet (Fellbaum 1998), which is also features a larger vocabulary, but the properties are derived from a large taxonomic hierarchy.

[b] We do not compare to Derby et al. as the work was developed after submission.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 5. Some of the GI property values for the noun *board*. Property is the GI category. Description is a description of what social aspect the category represents. Value is the fraction of noun word senses for *board* in GI that are in the category

Property	Description	*Value*
Academ	Academic	0.5
COM	Communication	0.5
ComForm	Form of communication	0.25
ComnObj	Communication object	0.25
Econ	Economic	0.25
EnlTot	Enlightenment	0.25
HU	Human	0.25
Object	Object	0.5
POLIT	Political	0.25
Power	Power	0.25
Tool	Tool	0.25
WlbPhys	Phys. Wellbeing	0.25
WlbTot	All Wellbeing	0.25

Two property datasets we explore are the McRae *et al*. feature norms[c] (**McRae**) and the Vinson–Vigliocco feature norms (**VV**). In line with previous research, we encode the properties of a word as a vector in $\mathbf{R}^n$. For McRae and VV the encoding is as follows: For a property q, the property value for a concept c is the percentage of participants that said c has q. Properties that were not mentioned for a concept have a value of 0.

The other two datasets are new to the task. The first is constructed from GI (Stone 1997). GI is a dataset historically used in sentiment analysis. In this dataset, part-of-speech tagged word senses are placed into categories that indicate sentiment and social significance, for example, there is a category for positive word senses and for word senses dealing with politics. We construct a property dataset that we call General Inquirer Mean Sense (**GenInqMS**) from GI as follows.[d] Each lemma–part-of-speech (POS) pair in GI has all of the properties of all of its senses. For example, "argue-v" will have the property "Ngtv" (negative words), because one of its senses in GI, ARGUE#1 (to have an argument), is in the Ngtv category. The property weight is the fraction of senses that have that property. For example, "argue-v" will have a property value of 0.5 for "Ngtv" as one of its two senses has that category (the other being ARGUE#2, which represents the "to present reasons" sense of *argue*). Table 5 shows an example.

The second new dataset, **WN-Hyp** (for WordNet hypernyms), is derived from WordNet hypernymy relations. The concepts in this dataset are the noun and verb lemmas listed in WordNet (Fellbaum 1998), with part-of-speech tag attached. Due to the large number of lemmas and synsets in WordNet, we only use words that appear at least 800 times in the BNC as compiled by Kilgariff

[c] While the Centre for Speech, Language and the Brain concept property norms (Devereux *et al*. 2014) are also widely used for property inference tasks, they share a similar vocabulary and construction methodology to the McRae et al. feature norms, so we do not explore them as part of our analysis.

[d] This and the following dataset will be made publicly available upon acceptance.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 6. Some of the WN-Hyp property values for the noun *dance*. Property is a WordNet synset. Value is the fraction of occurrences of *dance* in SemCor that have this synset as a hypernym

Property	Value
abstraction.n.06	0.211
activity.n.01	0.053
art.n.01	0.789
cognition.n.01	0.053
creation.n.02	0.789
entity.n.01	1.000
event.n.01	0.053
group.n.01	0.158
humanistic_discipline.n.01	0.053
object.n.01	0.789
party.n.02	0.158
performing_arts.n.01	0.053
physical_entity.n.01	0.789
social_gathering.n.01	0.158
whole.n.02	0.789

(1997). As properties, we use WordNet synsets. A property q will have a nonzero value[e] for a lemma–POS pair w if q is a hypernym of a synset containing w. This includes non-direct hypernyms, for example, "fruit.n.01" is a property of "lime-n" even though "fruit.n.01" is not a direct hypernym of any synset containing the word *lime*. We weight properties by synset frequency in the Semcor corpus (Miller *et al.* 1993; Langone, Haskell, and Miller 2004), which is an English corpus where words are tagged with their WordNet synset. The value of q for w is the percentage of occurrences of w in SemCor whose synsets have q as a hypernym. Lemma–POS pairs that did not appear in SemCor were removed. Table 6 shows an example from WN-Hyp.

WordNet is often used in hypernymy detection, which is a task related to property inference and has a large body of literature of its own, for example, Bernier-Colborne and Barriere (2018), Nickel and Kiela (2017), Pinter and Eisenstein (2018), Roller, Erk, and Boleda (2014), and Ustalov *et al.* (2017). Our focus is solely viewing WordNet synsets as taxonomic properties within an overall property prediction endeavor.

We note that the property values in GenInqMS and WN-Hyp should not be interpreted as the "importance" of the property to a given word, but rather the frequency of the property among all uses of the word. Like in distributional modeling, we consider the representation of a word to be a mixture of all its senses, that is, a mixture of properties relevant to all its senses. For WN-Hyp, the weight of a property reflects the relative frequency of the sense that the property goes with. For

[e] An alternative approach to encoding the properties of a word is to embed words, properties, and word senses in such a way that they live in the same space. Rothe and SchÜtze (2015) use this approach to generate embeddings for WordNet synsets and perform word sense disambiguation.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 7. Number of shared properties and Jaccard similarity for pairs of words in the BLESS dataset. We applied an independent sample t-test between successive relations and found each link is strongly statistically significant ($p < 0.001$)

	GenInqMS		WN-Hyp	
Relation	Shared	Jaccard	Shared	Jaccard
Co-hyponymy	1.66	0.55	7.97	0.58
Hypernymy	1.40	0.38	5.07	0.43
Meronymy	0.62	0.14	3.24	0.24
No relation	0.17	0.03	2.08	0.14

GenInqMS, we do not have corpus frequency information so we assumed equal frequency for all senses, but a property can still gain more weight if it is appropriate to multiple GI senses.

We perform an analysis of GenInqMS and WN-Hyp to evaluate their quality. To do this, we explore the connection between semantic relations (e.g., co-hyponymy, hypernymy, and meronymy) and the number of shared properties. If our datasets reflect semantic information about a concept, then co-hyponyms should share the most number of properties (co-hyponyms should be semantically similar and thus have overlapping sets of properties), followed by hypernyms (hyponyms generally have the properties of their hypernyms), then meronyms (part–whole relations share few properties), then pairs that have no relation at all (random pairs should share minimal properties). To do this, we leverage BLESS (Baroni and Lenci 2011), a semantic relation dataset. This dataset contains a pairs of words with the semantic relation that connect them, for example, *animal-alligator* is a pair of words and this pair is marked with the hypernymy relation. We focus our analysis to relations between nouns and the following relations: co-hyponymy, hypernymy, meronymy, and no relation (random-n in dataset).

As GenInqMS and WN-Hyp have real-value property values, we measure the number of shared properties using a sum of the min value for each property:

$$Shared(a, b) = \sum_i \min(a_i, b_i)$$

To correct for number of properties, we also provide the Jaccard similarity:

$$Jaccard(a, b) = \frac{\sum_i \min(a_i, b_i)}{\sum_i \max(a_i, b_i)}$$

For each value, we calculate the average across all pairs and present the results in Table 7.

We see that co-hyponymy pairs have the highest number of shared properties followed by hypernymy pairs, then meronymy pairs, then finally pairs of words with no semantic relationship. In addition, we ran an independent sample t-test between each pair of relations (co-hyponymy to hypernymy, hypernymy to meronymy, and meronymy to no relation) and determined that each link is strongly statistically significant ($p < 0.001$). This provides evidence that GenInqMS and WN-Hyp contain reliable property information.

4. Methods

In this paper, we compare both existing and new methods on the task of property inference. Of the techniques used in previous research, we evaluate both one-step and two-step Johns and Jones (JJ1, JJ2), SVM regression (SVM), and Partial Least Squares Regression (PLS).

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Both for PLS and for SVM regressions, we also experiment with a different kernel, one based on cosine similarity (***cosine PLS***, ***cosine SVM***). Cosine similarity is known to favor co-hyponymy (Baroni and Lenci 2011), sister terms in a hierarchy that tend to share many properties.

For SVM regression, the exchange of kernels is straightforward. For PLS, Rosipal *et al.* (2002) created a kernel version. Standard PLS learns a linear function from predictors to responses, while kernel PLS allows for a nonlinear relationship. In standard PLS, the matrices T and U can be derived from $XX^\intercal$ because covariances are a function of $XX^\intercal$. In kernel PLS, we instead derive T and U from a matrix K where $K_{i,j} = K(x_i, x_j)$ for rows x_i, x_j of X. Standard PLS is just kernel PLS with a linear kernel as can be seen by noting that $XX^\intercal$ is the kernel matrix K where $K_{i,j} = x_i \cdot x_j$.

Another method we explore is label propagation. Label propagation is a machine learning method where the predicted properties for a word are an aggregate of the properties of its neighbors. In particular, the predicted value of property q for word w, $v_q(w)$, is a weighted sum of that property weight across all words:

$$v_q(w) = \sum_{w'} W_{w,w'} v_q(w')$$

where $W_{w,w'}$ is a measure of the similarity in usage between w and w'. If $W_{w,w'} = cos(\vec{w}, \vec{w'})^{\lambda_1}$ for some λ_1, then this formula is equivalent to Johns and Jones. Unlike Johns and Jones, label propagation repeats application of the above formula until the predicted property values converge.

In addition to the natural connection to Hintzmann's work (1986, 1988) due to the parallel to the Johns and Jones method, label propagation can be seen as cognitively plausible under a "theory theory" analysis of word learning (Murphy 2004). Under theory theory, people do not learn concepts by connecting them to a single entity (as in prototype and exemplar theory) but instead people learn concepts by combining knowledge from several related concepts. For example, people were able to easily understand the operation and function of tablet computers (such as iPads), because they were able to draw from their knowledge of laptops, smartphones, touchscreens, and related technology.[f] Label propagation can be seen through a theory theory lens as predicted property values for a novel word come from a sum of property values across all words where the summands are weighted by their relation to the novel word.

The particular label propagation approach we use is modified adsorption (Talukdar and Crammer 2009) (**ModAds**). Modified adsorption differs from the original label propagation algorithm in that it incorporates something like a failsafe mechanism that checks if propagation "makes sense." If we already have good knowledge of a word's properties, then it is unnecessary to add to the word's meaning representation, and the mechanism just returns the meaning representation we already have. Alternatively, if a word is completely different from every word we have knowledge of, say it is technical jargon in an unfamiliar field, then it does not make sense to determine that word's properties. Then the mechanism just returns an empty meaning representation, that is, it does not attach any properties to the word. By incorporating these mechanisms, modified adsorption has greater flexibility when making predictions.

Modified adsorption models the above situations by breaking up the propagation process into three separate possibilities: "inject" (sticking with the known meaning), "continue" (propagating the meaning from similar words), and "abandon" (giving up and returning an empty meaning representation). In particular, for each word w, there are probabilities p_w^{inj}, p_w^{cont}, and p_w^{abnd} that measure the probability of each possibility happening when encountering w.[g]

With these probabilities and working within our property inference framework, the value of property q for word w, $v_q(w)$, is defined as the function that minimizes the following conditions:

[f] Example from https://nobaproject.com/modules/categories-and-concepts#knowledge.

[g] The formula for p_w^{inj}, p_w^{cont}, and p_w^{abnd} are derived from the entropy of transition probabilities. Further details can be found in Talukdar and Crammer (2009).

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

1. Inject: $p_w^{inj} \| v_q^{known}(w) - v_q(w) \|$ where $v_q^{known}(w)$ is the known value[h] of q for w. In other words, if a person decides to stick with the known meaning of a word (with probability p_w^{inj}), then the predicted value $v_q(w)$ should match the known value $v_q^{known}(w)$.
2. Continue: $p_w^{cont} \sum_{w'} W_{w,w'} \| v_q(w) - v_q(w') \|$ where $W_{w,w'}$ is related to the similarity in usage between w and w' (defined below). In other words, if a person decides to derive the meaning of a word (with probability p_w^{cont}), then the derived meaning should match the meaning of similar words.
3. Abandon: $p_w^{abdn} \| v_q(w) - r_w \|$ where r_w is 1 for known words and 0 for unknown words. In other words, if a person decides to give up on figuring out the meaning of a word (with probability p_w^{abdn}), then the predicted value $v_q(w)$ of an unknown word should be 0 to be as uninformative as possible.

As a machine learning algorithm, these formulas are summed to form a loss function and $v_q(w)$ is found by minimizing the loss. These three formulas are weighted by hyperparameters μ_{inj}, μ_{cont}, and μ_{abdn}.

Similar to two step Johns and Jones, we use modified adsorption as a semi-supervised approach. That is, we include an extra set of words into the training set that lack any property information but do contain usage information (word vectors). For our purposes, these extra words act as a secondary passage from the words we know to the words we want to know. For example, we may be given a weird hand tool that we cannot connect with the tools that we are familiar with, but, if we are also given a large set of other unknown tools that have similarities to both the tools we know and the unknown tool, we can put the pieces together to understand the purpose of the tool. We note that incorporating extra words into the training set is a natural extension to label propagation-based methods but is not natural to apply to direct vector to property methods such as PLS and SVM.

The "continue" possibility depends on a graph $W_{w,w'}$ that measures the similarity in usage between each pair of words. We explore three different ways of constructing the graph. The first, which we call ***ModAds NN***, is a k-nearest-neighbor graph (see Talukdar and Crammer 2009). For nodes u and v, the weight of the edge from u to v is 1 if v is one of the k most similar words (by cosine similarity between the word vectors) to u.

A k-nearest-neighbor graph weighs all of the k-nearest neighbors equally. However, closer neighbors should share more properties than further neighbors. We can capture this by forming a weighted sum of nearest-neighbor graphs. Suppose we have k_i-nearest-neighbor graphs G_i for $1 \leq i \leq n$. We assume $k_i < k_j$ for $i < j$, though the k_i do not have to be consecutive. Suppose further that we have graph weights $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{R}^n$ for the nearest-neighbor graphs. Then we define a new graph G_α for use in ModAds by defining the weight on the edge from u to v as:

$$w_{uv} = \frac{\sum_i \alpha_i w_{uv}^{(i)}}{\sum_i \alpha_i}$$

where we write $w_{uv}^{(i)}$ for the weight of the edge w_{uv} in the i-th nearest-neighbor graph G_i.

One version that we explore, ***ModAds equal***, gives all nearest-neighbor graphs equal weights. We also explore another version, ***ModAds decay***, where the contribution of a nearest-neighbor graph G_i decays exponentially the more neighbors it includes. To do that, we set the weight for the i-th nearest-neighbor graph to be $\alpha_i = 2^{-i}$.

We also test a modification that applies to all models in which property values of zero are shifted to a negative number. A word has a positive value for a property when there is some evidence

[h] $v_q^{known}(w) = 0$ for unknown words.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

that the word has the property, for example, a participant's response. However, in some property datasets, these positive values can be close to 0, therefore not providing a good separation from properties that do not apply. In order to prevent machine learning algorithms from confusing small property values with zero property values, we shift properties with a zero property value to be a negative number, namely the negative of the average of the positive property values. We explored shifting the zero property values (denoted **shifted**) for all methods except JJ. As JJ is a weighted sum of property values, shifting zero values would make the method unusable.

5. Experimental framework

Data To measure distributional similarity, we use a count-based distributional model from Roller *et al.* (2014). It uses a two word context window and is generated from the ukWaC (Baroni *et al.* 2009), Google Gigaword (Graff *et al.* 2003), Wikipedia (Baroni *et al.* 2009), and BNC (BNC 2007) corpora. The corpora are lemmatized and POS-tagged and only the content words (nouns, proper nouns, adjectives, and verbs) with frequency greater than 500 are retained. As a result, the targets and contexts in this space are 132,000 lemma–POS pairs with POS tags ranging over common noun, proper noun, verb, adjective, and adverb. A two-word window model was chosen as it models similarity more than relatedness (Agirre *et al.* 2009). PPMI was applied and SVD was used to reduce to 500 dimensions. We chose PPMI over PMI as negative PMI values are not a reliable measure. Negative PMI values indicate that two words do not cooccur more than random chance, which requires many orders of magnitude more data to capture accurately (Jurafsky and Martin 2009). There is work that suggests adding a nonnegative constant to the cooccurrence counts before applying PPMI (Levy and Goldberg 2014). However, the actual benefit of this addition is inconclusive, so we do not incorporate this approach into our work.

For semi-supervised algorithms that use an additional set of unlabeled words along with the training and test words (JJ2, ModAds), we use all nouns, verbs, adjectives, and adverbs that appear at least 800 times in the BNC corpus (1997), for a total of 4161 lemma–POS pairs.

Evaluation We perform 10-fold cross-validation to compare models. As each model has hyperparameters that need to be tuned, one of the nine training subsamples is used as a development set for that fold.

In Table 8, we display the grids we used to tune the hyperparameter settings. For *linear SVM shifted* on WN-Hyp, we used a smaller grid search with ϵ in $2^{-7}, ..., 2^{-1}$ due to technical issues, but inspection of results indicates that optimum is reached within that grid. For *ModAds equal* and *ModAds decay*, we optimized $n \in \{3, 4, 5, 6\}$ with $k_1 = 1, k_2 = 5, k_3 = 10, \ldots, k_n = 5 \cdot 2^{n-2}$. For replicability purposes, we will make the best performing hyperparameters for each model publicly available.

Our metrics for model performance are Spearman's ρ and MAP.[i] Spearman's ρ measures the correlation between rankings of gold and predicted property values, so it shows to what extent a model captures the relative weight of a property for a concept. In contrast, MAP measures the extent to which a model ranks properties that apply above properties that do not apply to a concept.

We compute Spearman's ρ separately for each concept and average over the results. In other words, this is a macro Spearman's ρ. For MAP, we calculate average precision for each conceptand then compute the MAP across all concepts.

To better ground the results of our evaluation, we provide two baselines, **Property Frequency** and **Property Sum**. For Property Frequency, the predicted value of a property for a given word is

[i] Pearson correlation has also been used as an evaluation metric in the property inference task (Rubinstein *et al.* 2015), but initial experiments have shown that our data do not satisfy the normality assumption of this metric.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 8. Hyperparameter settings for our experiments. A list of numbers in the "Value(s)" column represents the grid used in the grid search. "Reference" provides the source for the value(s)

Method	Hyperpar.	Reference	Value(s)
Johns and Jones	λ_1	This work	1, 5, 10, 50,100, 500, 1000
Johns and Jones	λ_2	This work	1, 5, 10, 50, 100, 500, 1000
SVM	C	Hsu, Chang, and Lin (2003)	$2^{-5}, 2^{-3}, ..., 2^3, 2^5$
SVM	ϵ	Hsu *et al*. (2003)	$2^{-13}, 2^{-11}, ..., 2^{-1}$
PLS	latent dim.	This work	25, 50, ..., 175, 200
ModAds	μ_{inj}	Talukdar and Crammer (2009)	1
ModAds	μ_{cont}	Talukdar and Crammer (2009)	$10^{-8}, 10^{-4}, 10^{-2},$
			1, 10, 100, 1000
ModAds	μ_{abdn}	Talukdar and Crammer (2009)	$10^{-8}, 10^{-4}, 10^{-2}$,1, 10, 100, 1000
ModAds NN	k	This work	1, 5, 10, 20
ModAds equal/decay	n	This work	3, 4, 5, 6

the number of words in the training data that have that property. For Property Sum, the predicted value for a property is the sum of the values for that property among all words in the training data.

6. Quantitative results

The results of our evaluation are in Tables 9 and 10. In general, ModAds approaches outperform other methods by a wide margin. By Spearman's ρ, shifted *ModAds NN* achieves the best performance across all datasets. On the MAP evaluation, shifted *ModAds decay* shows the best performance on the two feature norm datasets, while shifted *linear SVM* has the best performance on the other two datasets, GenInqMS and WN-Hyp.

Concerning the cosine kernel, PLS with cosine kernel outperforms PLS with a linear kernel across all datasets and under both evaluations. For SVM, the results are more mixed under a Spearman's ρ evaluation. Under a MAP evaluation, using a cosine kernel over a linear kernel with SVM greatly increases performance for McRae and VV but greatly decreases performance for GenInqMS and WN-Hyp.

Shifting zero values generally increases performance under a MAP evaluation, but the effect of shifting on performance is mostly negligible under Spearman's ρ.

Under Spearman's ρ, ModAds with a single kNN graph outperforms a mixture of kNN graphs, but the situation is reversed using MAP. Using Spearman's ρ, there is little difference between having all α_i be equal versus having them decay. However, under a MAP evaluation, decaying α_i perform slightly better than equal α_i.

The very simple JJ1 approach consistently outperforms *linear SVM* and *linear PLS* and is at a similar level as the cosine variants of SVM and PLS but is outperformed by ModAds. Under both evaluations and across all property datasets, JJ2 performed at the same level or worse than JJ1.

To make the patterns in our results clearer, we look at two factors: the effect of evaluating with Spearman's ρ versus MAP, and the differences between the property datasets. Spearman's ρ measures to what extent a model is correct in predicting the relative magnitudes of property values. MAP, on the other hand, tests to what extent a model ranks the properties that apply to a concept above the properties that do not apply. The property datasets fall into two groups. The

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 9. Overall Comparison of Property Prediction Methods via the Spearman's ρ Evaluation. Statistical comparison was performed using a paired t-test between the best performer(s) and the second best performer. Notes: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

Method	McRae	VV	GenInqMS	WN-Hyp
Property frequency	0.049	0.128	0.161	0.063
Property sum	0.042	0.112	0.161	0.060
Johns and Jones (1 step)	0.114	0.232	0.229	0.130
Johns and Jones (2 step)	0.107	0.228	0.228	0.128
linear SVM	0.077	0.151	0.228	0.068
linear SVM shifted	0.089	0.159	0.228	0.067
cosine SVM	0.082	0.185	0.216	0.069
cosine SVM shifted	0.082	0.186	0.219	0.070
linear PLS	0.077	0.177	0.222	0.067
linear PLS shifted	0.075	0.188	0.191	0.057
cosine PLS	0.082	0.187	0.230	0.068
cosine PLS shifted	0.083	0.198	0.231	0.070
ModAds NN	0.244	0.279	0.327	0.280
ModAds NN shifted	**0.281*****	**0.321*****	**0.367*****	**0.313*****
ModAds equal	0.161	0.257	0.242	0.145
ModAds equal shifted	0.244	**0.321*****	0.333	0.269
ModAds decay	0.161	0.258	0.241	0.145
ModAds decay shifted	0.243	**0.321*****	0.333	0.270

first group contains the McRae and VV feature norm datasets, which have human participants provide a small number of definitional properties for each concept – properties that they found noteworthy or salient in some way. So the property values on these datasets do not reflect what a participant considers true about a concept, but rather what a participant considers noteworthy, such that many true properties receive a value of zero. For example, the property *has_a_tail* is not listed for the concept *tiger*, but it is listed for other concepts such as *squirrel* and *zebra*. The second group, GenInqMS and WN-Hyp, can be characterized as categorization-based, where taxonomy creators placed a concept in all the categories that applied.

With these points in mind, we turn back to our results. ModAds shows very good performance on both evaluation measures and most datasets. One possible reason for this is that it directly implements Hintzman's idea of property transfer through contextual similarity – like JJ, which in spite of its simplicity does surprisingly well. We think that one important reason why ModAds surpasses JJ is its ability to both expand and restrict the pool of available evidence. ModAds is a semi-supervised approach and as such can use a larger pool of evidence. This also holds for JJ2, which however does not profit from its access to more words than JJ1 sees. But the larger pool of words in ModAds is counterbalanced by its use of a nearest-neighbor graph: Properties for a word are learned only from a small number of most similar words, ignoring evidence from less closely related words. This cutoff of more distant neighbors is most pronounced in *ModAds NN*, which

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 10. Overall Comparison of Property Prediction Methods via the MAP Evaluation. Statistical comparison was performed using a paired t-test between the best performer and the second best performer. Notes: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

Method	McRae	VV	GenInqMS	WN-Hyp
Property frequency	0.108	0.174	0.215	0.303
Property sum	0.101	0.160	0.215	0.304
Johns and Jones (1 step)	0.327	0.357	0.504	0.457
Johns and Jones (2 step)	0.326	0.356	0.505	0.457
linear SVM	0.293	0.267	0.579	0.496
linear SVM shifted	0.317	0.314	**0.581*****	**0.514*****
cosine SVM	0.337	0.361	0.545	0.468
cosine SVM shifted	0.345	0.359	0.545	0.482
linear PLS	0.317	0.312	0.495	0.442
linear PLS shifted	0.156	0.293	0.473	0.259
cosine PLS	0.333	0.325	0.570	0.484
cosine PLS shifted	0.353	0.370	0.571	0.501
ModAds NN	0.330	0.350	0.531	0.451
ModAds NN shifted	0.343	0.380	0.531	0.458
ModAds equal	0.341	0.364	0.542	0.460
ModAds equal shifted	0.354	0.390	0.543	0.467
ModAds decay	0.345	0.368	0.547	0.465
ModAds decay shifted	**0.361*****	**0.392*****	0.547	0.470

uses a single nearest-neighbor graph, while the mixture of nearest-neighbor graphs in *ModAds equal* and *ModAds decay* acts as a smoothing operation that allows for some inference based on more distant neighbors.

Measured in terms of Spearman's ρ, ModAds outperforms all other approaches across all four datasets. Interestingly, it is *ModAds NN* that has the best performance under Spearman's ρ. So our hypothesis is that for the task of estimating relative magnitude of property values (as measured by Spearman's ρ), the semi-supervised nature of the approach is beneficial in that the model can draw on more evidence, but that the noise introduced by the additional evidence must be counterbalanced by only drawing on a few highly similar neighbors. Note that under a MAP evaluation, *ModAds decay* and *ModAds equal* have better performance than *ModAds NN*. So the additional smoothing that they introduce seems to be helpful for distinguishing properties that do apply from properties that do not apply, while for a more fine-grained evaluation in terms of Spearman's ρ the smoothing introduces too much noise.

Under a MAP evaluation, we do not have a single winning model: ModAds performs best on the McRae and VV dataset, while SVM and PLS shine on the GenInqMS and WN-Hyp datasets. As we said above, the feature norm datasets McRae and VV omit some properties which, while true, were less noteworthy to the participants. This can cause issues with SVM and PLS, as they rely on knowing not only to what extent a property applies, but also to what extent it does not apply.

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

This is less of an issue with ModAds. As a random walk model, it can smooth property values to better handle missing values. In contrast, because of the construction of GI and WordNet, a zero property value in GenInqMS and WN-Hyp does specifically mean that the concept does not have that property. Thus, SVM and PLS approaches can function well.

The use of a cosine kernel instead of a linear kernel in SVM and PLS mostly leads to an increase in MAP. A notable exception is for SVM regression applied to GenInqMS and WN-Hyp, where a linear kernel outperforms a cosine kernel and in fact produces the best results on these datasets under MAP. This suggests that it is possible to find a linear mapping between distributional vectors and properties without needing to employ the use of word similarity. PLS considers all properties jointly rather than separately as SVM does. The MAP results for PLS indicate that cosine similarity seems to help discover the latent structure connecting distributional vectors and property vectors.

Zero value shifting improves performance under MAP on all datasets except for GenInqMS. A possible reason for this lies in a general property of shifting, namely that zero value shifting is redundant if positive property values are well separated from zero property values. GenInqMS, unlike the other property datasets, has a strong separation between positive property values and zero. For McRae and VV, the property values are the percent of participants that said a given concept has that property. This results in McRae and VV property values being spread out between 0 and 1. For WN-Hyp, property values are based on the percentage of times a word has a given sense in the Semcor corpus and thus also has property values that are spread out. In contrast, property values in GenInqMS are calculated as the fraction of a word's senses that have a given property. As concepts in GI have few senses (1.15 senses on average), the denominator of these fractions are small and, thus, very few property values approach zero. In fact, the value 1 makes up 88.5% of positive property values in GenInqMS. As very few positive property values appear in the gap between 0 and 1, the positive property values are well separated from 0.

The above results suggest the following guidelines as to which approach to use for a given property dataset. When the goal is to capture the relative strength of a property, modified adsorption with a k-nearest-neighbor graph and shifting zero values is the best approach regardless of the attributes of your property dataset of interest. However, if the goal is to separate likely properties from unlikely properties, the attributes of the property dataset affect the results. For property datasets where a zero value is not necessarily a negative result (such as an elicitation-based dataset), zero shifting provides immediate benefit as does modified adsorption with decaying influence. In contrast, for property datasets where zero values do convey negative data (such as strict word categories), traditional classifier approaches (SVM and PLS) work best and zero shifting is largely optional.

7. Qualitative results

In Section 6, we undertook a quantitative analysis to compare the various methods. In this section, we will analyze the results of specific methods to gain insight into how these methods work and how to interpret the results.

7.1. Analysis of category effects on model efficacy

7.1.1. Part of speech

First, we will explore the connection between part of speech and model effectiveness. In Table 11, we present the MAP of the best-performing models, split by part of speech. For VV and WN-Hyp, we see that models are much more effective at predicting properties for nouns than verbs. What this indicates is that feature norms and taxonomic properties may not be effective at encoding a verb's features. In contrast, for GenInqMS, we see a minor drop in MAP between the two parts

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 11. Mean average precision for best performing models separated by part of speech

Dataset	Overall MAP	Noun MAP	Verb MAP	Adjective MAP
McRae	0.361	0.361	N/A	N/A
VV	0.392	0.439	0.339	N/A
GenInqMS	0.581	0.604	0.585	0.533
WN-Hyp	0.504	0.639	0.204	N/A

of speech. What this indicates is that sentiment and social significance may be effective representations for verbs. This makes sense as the feature norms and taxonomic properties of the words *money* (a noun), *buy* (a verb), and *financial* (an adjective) are quite different, but each of these words are placed well within economic transactions, a social event. Under GI, all of these words have the Econ@ (economic) and WltTot (wealth domain) properties. Similarly, *church* (a noun), *pray* (a verb), and *spiritual* (an adjective) all correspond to different kinds of concepts, but all have the Relig (religious) property in GI.

7.1.2. Verb type analysis

We mentioned in Section 7.1.1 that GI has less of a drop in average precision from Noun MAP to Verb MAP compared to other property datasets. In this section, we explore this further by analyzing what kinds of verbs are difficult for each dataset.

The verb types we analyze come from Semin and Fiedler (1988). We chose this categorization as GI has word lists for each of these verb types, which allow us to map verbs from VV, GenInqMS, and WN-Hyp to these verb types. The first verb type is descriptive verb (DAV), which are descriptive action verbs. These refer to a particular activity with a physically invariant feature that has a clear start and end, for example, *call*, *kiss*, *talk*, and *stare*. The second verb type is interpretive verb (IAV), which are interpretive action verbs. These refer to interpretations of an observable event, for example, *help*, *cheat*, *inhibit*, and *imitate*. The third verb type is SV, which are state verbs. These refer to mental or emotional states, for example, *like*, *hate*, *notice*, and *envy*.

In order to compare verb type difficulty across datasets, we used z-score to standardize the average precision scores for each dataset. We then take the average standardized average precision score for each dataset and verb type to derive a value we can use to compare verb types and property datasets on an even footing. For example, if the average GenInqMS standardized AvgPrec score for the DAV verb type is 1.0 (very above average) and the average VV standardized AvgPrec score for the same verb type is −1.0 (very below average), then we can say that it is relatively easier to predict descriptive action verbs under the GenInqMS properties than the VV properties. To maintain comparability between datasets, we use AvgPrec results from the same model, *ModAds decay shifted*, as it was shown to perform very well across datasets. We also do not include the McRae property dataset in this analysis as it does not have verbs.

We present the results of this analysis in Figure 2. First, we find that WN-Hyp has much worse relative average precision scores across the board compared to VV and GenInqMS. This is unsurprising as WN-Hyp has a much greater difference between Overall MAP and Verb MAP (68% decrease) than VV (13% decrease) or GenInqMS (1% increase). Second, we see that GenInqMS has a higher AvgPrec for IAVs than DAVs, whereas VV has the opposite results. A possible explanation of the GenInqMS case is that GI has categories that deal with social circumstance and sentiment, which can be more readily applied to IAV verbs (which generally have a stronger connotation component), but not DAV verbs (which generally do not) (Semin and Fiedler 1988). In contrast, VV properties are perceptual and so may be more amenable to DAV verbs (which are

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

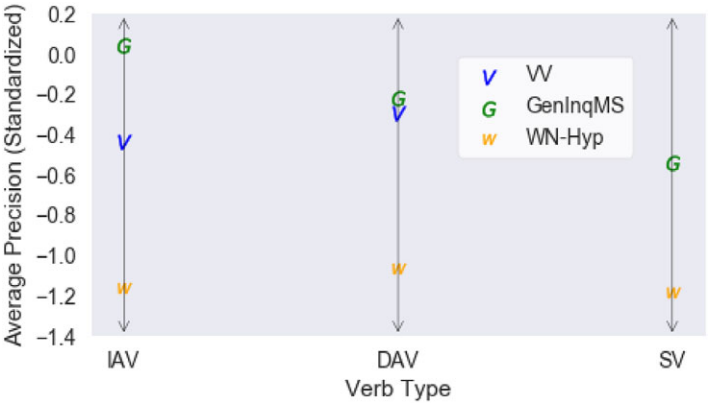

Figure 2. Comparison of average precision scores by property dataset and verb type. VV includes less than 5 SV verbs, so is omitted from this figure.

associated to a clear observable event) than IAV verbs (which are more rooted in perspective). The third observation is that GenInqMS has a lower average precision for SV than the other verb types. A possible explanation for this is that SV verbs are not associated to an observable event, like DAV and IAV are by definition, and thus are harder to represent via word vectors. Word vectors are representations of usage and we use words to describe our observations. Without a strong observational component, it may be hard for a distributional model to get the right context information to produce a quality vector for SV verbs.

7.1.3. Distribution of properties

In this section, we generalize the discussion in the previous section by broadening our scope beyond verb types to all categories of words. To do this, we analyze the factors that contribute to property inference being more difficult to some word categories than others. For this analysis, we define a word category to be all words that have a given property, for example, for GenInqMS, the word category DAV is the list of all words that have the property DAV and, for McRae, the word category "made_of_metal" is the list of all words that have that property.

To analyze contributing factors, we will explore the correlation between category MAP and various metrics that have been explored in feature norm research. Additionally, we propose a new metric, consistency, which we will show is strongly correlated with average precision. These metrics are as follows:

1. **Distinctiveness**: Distinctiveness measures how much a property distinguishes a word (Devlin *et al*. 1998; Garrard *et al*. 2001). For example, the property "associated with polkas" is very distinguishing of a word like *accordion* as very few words have that property. The distinctiveness of a word category is calculated as the inverse of the number of words in that category.
2. **Intercorrelational density**: Intercorrelational density measures the extent that a word's properties are correlated to a given property (McRae, De Sa, and Seidenberg 1997, McRae *et al*. 1999). For example, if a word has the property "an animal,' intercorrelational density measures how many of that word's other properties are often used with animal words, such as "eats" and "has ears." High intercorrelational density means the other properties of the word are often seen with the target property ("an animal" in the example above). Intercorrelational density for a word and property is measured as the sum of the shared

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 12. Correlation between average precision and several measures of property distribution. Notes: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

Dataset	Distinctiveness	Intercorrelational Density	Consistency
McRae	-0.149^{**}	0.306^{***}	0.753^{***}
VV	-0.054	0.299^{***}	0.429^{***}
GenInqMS	-0.293^{***}	0.207^{***}	0.483^{***}
WN-Hyp	-0.053^{*}	0.243^{***}	0.728^{***}

variance between that property and other properties of that word. To derive an intercorrelational density for a word category, we take the average of the intercorrelational densities for words in that category.

3. **Consistency**: We propose a new measure called consistency, which measures how much the words in word category share properties. For example, words in a bird word category have a high consistency as birds share many properties, for example, have wings, fly, and have beaks. In contrast, a "used by humans" category has low consistency as there is less in common between a trombone and a submarine. Consistency between two words is measured by the Jaccard similarity between each word's set of properties, that is, the number of properties in common divided by the number of properties in the union. We then calculate a consistency score for a word category by averaging the consistency score across every pair of words in the category. To contrast with intercorrelational density, that one measures correlation between *properties* and consistency measures overlap between *words*.

We then correlate the MAP for a word category with these metrics. We remove every category that has less than five members. We present the results of this analysis in Table 12. We see that, except for GenInqMS, distinctiveness has a relatively weak negative correlation with category MAP. This indicates that the size of the category has little effect on the MAP. In contrast, intercorrelational density has a much stronger correlation. This makes sense as high intercorrelational density means that many properties appear in the same words, which means that predicting said properties should be easier.

From the table, we see that consistency has a much higher correlation across the board. What these high correlations indicate is that MAP for a word category is intimately connected to how contained the category is. If a word category has a low consistency, then the words in that category must have a wide variety of properties that are not shared between the words. This could result in the machine learning model not having a consistent idea of what makes up the category, which could lead to inaccuracy. In contrast, high consistency means that words in the category rarely have properties outside of a shared set. Thus, property inference is much simpler for these words.

7.2. Analysis of predicted properties

In this section, we explore what kinds of properties get predicted and how they relate to the gold properties. For each dataset, we analyze the top predicted properties for a selection of words as predicted by the best-performing model.

7.2.1. McRae feature norms

The first property dataset we will look at is the McRae feature norms. Under the MAP evaluation, the winning method was modified adsorption with a decaying transfer matrix and shifted

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 13. Top 10 properties for lime, potato, eggplant, and dandelion

Concept:	lime	potato	eggplant	dandelion
AvgPrec:	0.407	0.284	0.456	0.048
Properties:	**a_fruit**	**is_edible**	**a_vegetable**	is_green
	is_edible	**a_vegetable**	is_green	is_edible
	tastes_sweet	is_green	**is_edible**	a_vegetable
	grows_on_trees	is_white	**eaten_by_cooking**	has_leaves
	is_small	**eaten_by_cooking**	tastes_good	a_fruit
	is_round	eaten_in_salads	is_long	is_round
	is_yellow	**grows_in_gardens**	**grows_in_gardens**	is_long
	is_juicy	**is_nutritious**	is_nutritious	is_white
	has_seeds	is_round	is_white	is_small
	is_red	is_crunchy	eaten_in_salads	tastes_sweet

zero value properties (*ModAds decay shifted*). In Table 13, we present the top 10 properties for *lime*, *potato*, *eggplant*, and *dandelion*. These represent one fruit (lime), two vegetables (potato and eggplant), and one non-fruit non-vegetable plant (dandelion). In the fruit and vegetables, we see that colors are highly ranked. The fruit is predicted to be yellow and red (though, noticeably not predicted to be green) and the vegetables are predicted to be green and white. These colors are expected of these food types, that is, most fruits are yellow and red and most vegetables are green and white. However, these colors are not accurate for *lime* and *potato* as limes are not red and potatoes are not green. This may be the result of ModAds using similarities between *lime* and other words instead of drawing information from the vector for *lime* itself. Methods like linear SVM and linear PLS make predictions by applying trained weights to input vectors directly. Thus, they could find the "green" part of the *lime* vector and predict that *lime* corresponds to something green. In contrast, ModAds makes predictions using only similarities between words. Thus, *lime* is not predicted to be green as limes are much more similar to non-green fruits than green things, or, more specifically, the word *lime* is used more similarly to words for non-green fruits than words for things that are green. Thus, *ModAds decay shifted* predicts the word *lime* will have general fruit-like properties instead of lime-specific properties.

We can explore how *ModAds decay shifted* does with concepts that do not fit neatly into the semantic categories by analyzing the concept *dandelion*. Dandelions are neither a fruit or a vegetable. However, the method predicts that *dandelion* is both a fruit and a vegetable. The method predicts dandelions have some fruit properties (a_fruit and tastes_sweet) as well as some vegetable properties (a_vegetable and is_green). Thus, the method appears to place objects without a clear category into a combination of the closest categories.

7.2.2. Vinson–Vigliocco feature norms

The second property dataset we will look at is the Vinson–Vigliocco feature norms. Under the MAP evaluation, the winning method was again *ModAds decay shifted*. In Table 14, we present the top 10 properties for the verbs *skid*, *jog*, *ride*, and *pedal*. All of these concepts relate to transportation and movement but have very different average precisions. The difference in average

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 14. Top 10 properties for the verbs *skid*, *jog*, *ride*, and *pedal*

Concept:	skid (verb)	jog (verb)	ride (verb)	pedal (verb)
AvgPrec:	0.560	0.348	0.320	0.229
Properties:	**move**	**move**	**humans**	**humans**
	action	**leg**	**transport**	object
	object	**action**	**animal**	**action**
	slide	**humans**	ride	**transport**
	humans	feet	leg	**move**
	slip	animal	**move**	child
	car	go	**action**	2
	surface	**exercise**	2	ride
	noise	transport	fast	metal
	hurt	intentional	**object**	3-wheels

precision may be related to how the property dataset was constructed rather than the method itself. This can be seen by predicted properties. For example, the method predicts that *pedal* has the properties "child", "ride", "metal", and "3-wheels". None of these are gold properties for *pedal*, but are not inappropriate as pedals are used when riding, pedals are generally made of metal and children use pedals on their tricycles. Even the properties "2" and "object" are not too strange in the context of the Vinson–Vigliocco feature norms as "2" is often given as a property for paired objects and "object" is often given as a property for actions that interact with an object (see *ride* for example). In comparison, the gold properties that the method missed are not clearly better than the incorrect ones predicted. For example, *pedal*, in its set of gold properties, includes "intentional", "exercise", "travel", and "balance". While these are certainly appropriate, they are not noticeably better than "metal", "ride", and "object". Thus, the average precision seems to reflect the variability of what is considered a gold property rather than a defect of the method itself.

The variability in what is considered a gold property is most likely an artifact of how the dataset was designed. For each concept, 20 people were asked to provide a list of properties. A property is considered gold if at least one person provided it. Thus, "walk" is a gold property of *ride*, because one person suggested it. The variability in elicited properties can cause vastly different quantitative evaluations even though the real-world quality of the predicted properties does not differ. We should note that the McRae feature norms have less of an issue with this as McRae *et al*. used a cutoff to remove rare properties.

7.2.3. General Inquirer Mean Sense

The third property dataset we will look at is the General Inquirer Mean Sense dataset. The best-performing method under the MAP evaluation is SVM with a linear kernel and shifted zero value properties (*linear SVM shifted*). In Table 15, we present the top 10 properties for the adjective *southern* and the nouns *birthday*, *marine*, and *german*. These were chosen to highlight the disconnect between the curators of GI (the gold properties) and the usage of the word in text (the predicted properties). The curators tended to choose properties that reflected an objective social categorization of a concept. For example, *southern* is given the properties of "TimeSpc" (time and

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 15. Top 10 properties for *southern*, *birthday*, *marine*, and *german*

Concept:	southern	birthday	marine	german
AvgPrec:	0.833	0.475	0.750	0.525
Properties:	**TimeSpc**	Positiv	**PowTot**	HU
	Strong	**RspTot**	HU	Role
	Space	AffTot	Milit	**POLIT**
	POLIT	ABS	**PowAuPt**	**Name**
	Region	**Ritual**	Role	**Nation**
	IAV	**TIME**	Strong	**ECON**
	PowTot	Affil	POLIT	RcTot
	PLACE	**RspOth**	PowCon	PowPt
	Polit@	Kin@	COLL	Submit
	Pstv	Pleasur	Power	EMOT

space) and "Space." However, *southern* is also predicted to have the properties "Strong," "POLIT" (political), and "Region," which reflect more of a reference to a particular political area. Similarly, the gold properties for *birthday* reflect a ritual providing respect ("RspTot" and "Ritual"), but the predicted properties reflect the positive, affection-related feelings associated to birthdays (Positiv and AffTot). We also see how *marine* and *german* have gold properties associated with abstract position in society (marines are involved with power and Germans are a political entity), whereas the predicted properties reflect these concepts as personal attributes, for example, marines are humans in the military and Germans are humans and a social role. In essence, *linear SVM shifted* predicted properties that not only reflect the abstract role a concept plays in society, but also our day to day interaction with those attributes.

7.2.4. WordNet hypernyms

The final property dataset we will discuss is the WordNet Hypernyms dataset. The best-performing method under the MAP evaluation is *linear SVM shifted*. In Table 16, we present the top 10 properties for the nouns *pigeon* and *people* and in Table 17, we present the top 10 properties for the nouns *mathematics* and *theorist*. Unlike the other property datasets, the properties in this dataset form a strict hierarchy. This hierarchy plays a strong role in the accuracy of the predicted properties. Properties that are higher in the hierarchy (i.e., more general synsets) tend to do well as seen in the top properties of *pigeon*, *mathematics*, and *theorist*. However, properties that are lower in the hierarchy (i.e., more specific synsets) are predicted to be lower ranked, for example, for the word *pigeon*, the property "bird.n.01" is ranked 13th and the property "gallinaceous_bird.n.01" is ranked 22nd. Thus, this method excels at ranking correct general properties over incorrect general properties, but this method does not excel at ranking correct specific properties over incorrect specific properties.

An additional influence of hierarchy in the performance of the method on the WordNet Hypernyms property dataset is that the noun properties are strictly divided between abstract properties (descendants of the synset "abstraction.n.06") and physical properties (descendants of synset "physical_entity.n.01"). However, often times it is not clear whether a concept is abstract or physical, which causes difficulty in a method's ability to predict properties. For example, *people*

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

Table 16. Top 10 properties for *pigeon* and *people*

Concept:	pigeon	people
AvgPrec:	0.856	0.017
Properties:	**whole.n.02**	object.n.01
	physical_entity.n.01	organism.n.01
	object.n.01	living_thing.n.01
	entity.n.01	whole.n.02
	living_thing.n.01	causal_agent.n.01
	organism.n.01	person.n.01
	animal.n.01	physical_entity.n.01
	chordate.n.01	**entity.n.01**
	vertebrate.n.01	attribute.n.02
	matter.n.03	cognition.n.01

Table 17. Top 10 properties for *mathematics* and *theorist*

Concept:	mathematics	theorist
AvgPrec:	1.0	0.989
Properties:	**entity.n.01**	**entity.n.01**
	abstraction.n.06	**physical_entity.n.01**
	cognition.n.01	**object.n.01**
	knowledge_domain.n.01	**organism.n.01**
	discipline.n.01	**living_thing.n.01**
	psychological_feature.n.01	**person.n.01**
	content.n.05	**causal_agent.n.01**
	science.n.01	**whole.n.02**
	natural_science.n.01	abstraction.n.06
	attribute.n.02	**intellectual.n.01**

was predicted to be a physical entity, which makes sense as *people* refers to a collection of physical entities. However, groups are treated as abstract entities in the WordNet hierarchy. Given the strict separation between abstract and physical properties, a wrong guess about a concept causes the method to only predict properties in the wrong category. However, nouns (both concrete and abstract) that are associated with abstract concepts do not seem to have an issue with the abstract/physical divide. For example, the properties for the words *mathematics* and *theorist* were predicted accurately, even though both words are connected to abstract concepts (mathematics and theories).

https://doi.org/10.1017/S1351324921000267 Published online by Cambridge University Press

8. Conclusion

In this paper, we have studied supervised and semi-supervised models that infer properties from distributional vectors. We have proposed the use of label propagation (specifically, modified adsorption) for the task, and we have proposed two modifications that apply to several models, namely the use of a cosine kernel and the shifting of zero property values to negative values to achieve better separation between properties that do and that do not apply.

We find that modified adsorption, in particular shifted modified adsorption based on a single nearest-neighbor graph, is best at predicting a ranking of property values (based on a Spearman's ρ evaluation). Under a MAP evaluation, which focuses on the distinction of properties that do apply from those that do not, the differences between property datasets become more apparent. In terms of MAP, modified adsorption (in particular shifted modified adsorption with an overlay of multiple nearest-neighbor graphs with exponentially decaying weights) has the best performance on feature norm datasets, where it is necessary to smooth over property values that should be positive but are zero because of missing information. SVM, in particular with shifting and with a linear kernel, works best on the categorization-based datasets that we derived from the GI and from WordNet. We further find that using a cosine kernel instead of a linear kernel mostly improves performance. Shifting zero values to negative values in property datasets is also helpful, in particular for datasets that contain small positive values.

A future direction for property inference research is improving the construction of property datasets. Getting good property sets is hard. We have chosen two that have quite nice property lists, but they are only a small fraction of the things we know about a concept. One direction for future work is to explore semi-automatic approaches to property dataset creation that will allow researchers to produce larger and higher quality property datasets.

Some concepts are harder to describe in terms of features than others, and verbs are clearly harder than nouns, as can also be seen in the Vinson and Vigliocco feature norms. One direction of future work is to explore the property representation of verbs. One approach may be trajectories for events, an idea proposed by Gärdenfors (2014).

Supplementary material

To view supplementary material for this article, please visit https://doi.org/10.1017/S1351324921000267

References

Agirre E., **Alfonseca E.**, **Hall K.**, **Kravalova J.**, **Paşca M.** and **Soroa A.** (2009). A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado. Association for Computational Linguistics, pp. 19–27.

Almuhareb A. and **Poesio M.** (2004). Attribute-based and value-based clustering: an evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain. Association for Computational Linguistics, pp. 158–165.

Baroni M., **Bernardini S.**, **Ferraresi A.** and **Zanchetta E.** (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation* **43**(3), 209–226.

Baroni M. and **Lenci A.** (2010). Distributional memory: a general framework for corpus-based semantics. *Computational Linguistics* **36**(4), 673–721.

Baroni M. and **Lenci A.** (2011). How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, Edinburgh, UK. Association for Computational Linguistics, pp. 1–10.

Bernier-Colborne G. and **Barrière C.** (2018). CRIM at SemEval-2018 task 9: a hybrid approach to hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, New Orleans, Louisiana. Association for Computational Linguistics, pp. 725–731.

Bollacker K.D., **Evans C.**, **Paritosh P.**, **Sturge T.** and **Taylor J.** (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In Wang J.T. (ed), *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10–12, 2008*. ACM, pp. 1247–1250.

Bruni E., **Boleda G.**, **Baroni M.** and **Tran N.-K.** (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jeju Island, Korea. Association for Computational Linguistics, pp. 136–145.

Clark S. (2015). Vector space models of lexical meaning. In *The Handbook of Contemporary Semantic Theory*, Chapter 16. John Wiley & Sons, Ltd., pp. 493–522.

Derby S., **Miller P.** and **Devereux B.** (2019). Feature2Vec: distributional semantic modelling of human property knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 5853–5859.

Devereux B., **Pilkington N.**, **Poibeau T.** and **Korhonen A.** (2009). Towards unrestricted, large-scale acquisition of feature-based conceptual representations from corpus data. *Research on Language and Computation* **7**(2–4), 137–170.

Devereux B.J., **Tyler L.K.**, **Geertzen J.** and **Randall B.** (2014). The Centre for Speech, Language and the Brain (CSLB) concept property norms. *Behavior Research Methods* **46**(4), 1119–1127.

Devlin J.T., **Gonnerman L.M.**, **Andersen E.S.** and **Seidenberg M.S.** (1998). Category-specific semantic deficits in focal and widespread brain damage: a computational account. *Journal of Cognitive Neuroscience* 10(1), 77–94.

Drucker, **H.**, **Burges**, **C.J.C.**, **Kaufman L.**, **Smola A.J.** and **Vapnik V.** (1996). Support vector regression machines. In Mozer M., Jordan M.I. and Petsche, T. (eds), *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2–5, 1996*. MIT Press, pp. 155–161.

Erk K. (2012). Vector space models of word meaning and phrase meaning: a survey. *Language and Linguistics Compass* **6**(10), 635–653.

Fagarasan L., **Vecchi E.M.** and **Clark S.** (2015). 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. Association for Computational Linguistics, pp. 52–57.

Fellbaum C. (ed) (1998). *WordNet: An Electronic Lexical Database*. *Language, Speech, and Communication*. Cambridge, MA: MIT Press.

Feng Y. and **Lapata M.** (2010). Visual information in semantic representation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California. Association for Computational Linguistics, pp. 91–99.

Gärdenfors P. (2014). *The Geometry of Meaning: Semantics Based on Conceptual Spaces*. Cambridge, MA: MIT Press.

Garrard P., **Lambon Ralph M.A.**, **Hodges J.R.** and **Patterson K.** (2001). Prototypicality, distinctiveness, and intercorrelation: analyses of the semantic attributes of living and nonliving concepts. *Cognitive Neuropsychology* **18**(2), 125–174.

Graff D., **Kong J.**, **Chen K.** and **Maeda K.** (2003). English gigaword. *Linguistic Data Consortium, Philadelphia* **4**(1), 34.

Gupta A., **Boleda G.**, **Baroni M.** and **Padó S.** (2015). Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 12–21.

Herbelot A. (2013). What is in a text, what isn't, and what this has to do with lexical semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, Potsdam, Germany. Association for Computational Linguistics, pp. 321–327.

Herbelot A. and **Vecchi E.M.** (2015). Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 22–32.

Herbelot A. and **Vecchi E.M.** (2016b). Many speakers, many worlds: interannotator variations in the quantification of feature norms. In *Linguistic Issues in Language Technology, Volume 13, 2016*. CSLI Publications.

Hintzman D. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological Review* **93**(4), 411–428.

Hintzman D.L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review* **95**(4), 528.

Hsu C.-W., **Chang C.-C.** and **Lin C.-J.** (2003). A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University.

Johns B.T. and **Jones M.N.** (2012). Perceptual inference through global lexical similarity. *Topics in Cognitive Science* **4**(1), 103–120.

Jurafsky D. and **Martin J.H.** (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd Edn. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Pearson Education International.

Kilgarriff A. (1997). Putting frequencies in the dictionary. *International Journal of Lexicography* **10**(2), 135–155.

Langone H., **Haskell B.R.** and **Miller G.A.** (2004). Annotating WordNet. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, Boston, Massachusetts, USA. Association for Computational Linguistics, pp. 63–69.

Lazaridou A., **Bruni E.** and **Baroni M.** (2014). Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland. Association for Computational Linguistics, pp. 1403–1414.

Levy O. and **Goldberg Y.** (2014). Neural word embedding as implicit matrix factorization. In Ghahramani Z., Welling M., Cortes C., Lawrence N.D. and Weinberger K.Q. (eds), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada*, pp. 2177–2185.

McRae K., **Cree G.S.**, **Seidenberg M.S.** and **McNorgan C.** (2005). Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods* **37**(4), 547–559.

McRae K., **Cree G.S.**, **Westmacott R.** and **Sa V.R.D.** (1999). Further evidence for feature correlations in semantic memory. *Canadian Journal of Experimental Psychology = Revue canadienne de psychologie expérimentale* **53**(4), 360.

McRae K., **De Sa V.R.** and **Seidenberg M.S.** (1997). On the nature and scope of featural representations of word meaning. *Journal of Experimental Psychology: General* **126**(2), 99.

Miller G.A., **Leacock C.**, **Tengi R.** and **Bunker R.T.** (1993). A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21–24, 1993*.

Montefinese M., **Zannino G.D.** and **Ambrosini E.** (2015). Semantic similarity between old and new items produces false alarms in recognition memory. *Psychological Research* **79**(5), 785–794.

Murphy G. (2004). *The Big Book of Concepts*. A Bradford Book. Cambridge, MA: MIT Press.

Ng K.S. (2013). A simple explanation of partial least squares. Technical report, The Australian National University.

Nickel M. and **Kiela D.** (2017). Poincaré embeddings for learning hierarchical representations. In Guyon I., von Luxburg U., Bengio S., Wallach H.M., Fergus R., Vishwanathan S.V.N. and Garnett R. (eds), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, pp. 6338–6347.

Noraset T., **Liang C.**, **Birnbaum L.** and **Downey D.** (2017). Definition modeling: learning to define word embeddings in natural language. In Singh S.P. and Markovitch S. (eds), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA*. AAAI Press, pp. 3259–3266.

Pinter Y. and **Eisenstein J.** (2018). Predicting semantic relations using global graph properties. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 1741–1751.

Randall B., **Moss H.E.**, **Rodd J.M.**, **Greer M.** and **Tyler L.K.** (2004). Distinctiveness and correlation in conceptual structure: behavioral and computational studies. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **30**(2), 393.

Roller S., **Erk K.** and **Boleda G.** (2014). Inclusive yet selective: supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland. Dublin City University and Association for Computational Linguistics, pp. 1025–1036.

Rosipal R. and **Trejo L.J.** (2001). Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research* **2**, 97–123.

Rothe S. and **Schütze H.** (2015). AutoExtend: extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China. Association for Computational Linguistics, pp. 1793–1803.

Rubinstein D., **Levi E.**, **Schwartz R.** and **Rappoport A.** (2015). 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 (Volume 2: Short Papers)*, Beijing, China. Association for Computational Linguistics, pp. 726–730.

Semin G.R. and **Fiedler K.** (1988). The cognitive functions of linguistic categories in describing persons: social cognition and language. *Journal of Personality and Social Psychology* **54**(4), 558.

Stone P. (1997). Thematic text analysis: new agendas for analyzing text content. In Roberts C. (ed), *Text Analysis for the Social Sciences*. Mahwah, NJ: Lawerence Erlbaum Associates.

Talukdar P.P. and **Crammer K.** (2009). New regularized algorithms for transductive learning. In Buntine W.L., Grobelnik M., Mladenic D. and Shawe-Taylor J. (eds), *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7–11, 2009, Proceedings, Part II*, vol. 5782. Lecture Notes in Computer Science. Springer, pp. 442–457.

The British National Corpus, Version 3 (BNC XML Edition). (2007). Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium.

Turney, **P.D.** and **Pantel P.** (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* **37**(1), 141–188.

Turton J., **Vinson D.** and **Smith R.** (2020). Extrapolating binder style word embeddings to new words. In *Proceedings of the Second Workshop on Linguistic and Neurocognitive Resources*, Marseille, France. European Language Resources Association, pp. 1–8.

Tyler L.K., **Moss H.E.**, **Durrant-Peatfield M.** and **Levy J.** (2000). Conceptual structure and the structure of concepts: a distributed account of category-specific deficits. *Brain and Language* **75**(2), 195–231.

Ustalov D., **Arefyev N.**, **Biemann C.** and **Panchenko A.** (2017). Negative sampling improves hypernymy extraction based on projection learning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain. Association for Computational Linguistics, pp. 543–550.

Vieth H.E., **McMahon K.L.** and **de Zubicaray G.I.** (2014). The roles of shared vs. distinctive conceptual features in lexical access. *Frontiers in Psychology* **5**, 1014.

Vinson D. and **Vigliocco G.** (2002). A semantic analysis of noun-verb dissociation in aphasia. *Journal of Neurolinguistics* **15**, 317–351.

Vinson D.P. and **Vigliocco G.** (2008). Semantic feature production norms for a large set of objects and events. *Behavior Research Methods* **40**(1), 183–190.

Vinson D.P., **Vigliocco G.**, **Cappa S.** and **Siri S.** (2003). The breakdown of semantic knowledge: insights from a statistical model of meaning representation. *Brain and Language* **86**(3), 347–365.

Vulić I. and **Mrkšić N.** (2018). Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, pp. 1134–1145.

Wing B. and **Baldridge J.** (2011). Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA. Association for Computational Linguistics, pp. 955–964.

Cite this article: Rosenfeld A and Erk K (2023). An analysis of property inference methods. *Natural Language Engineering* **29**, 201–227. https://doi.org/10.1017/S1351324921000267