(2) The prediction in equation (1) averages out the posterior uncertainty $p(\theta|D)$. Such model averaging is used in the model proposed in the present paper. It avoids overfitting and often leads to more accurate predictions of performance compared to methods based on a point estimate $\hat{\theta}$.

There are two commonly used implementation methods for Bayesian learning. One is to design a Markov chain Monte Carlo (MCMC) method to draw samples from $p(\theta|D)$. The other is to design a simpler approximated distribution $q_\phi(\theta)$ parameterized by ϕ, and estimate ϕ by minimizing the Kullback-Leibler divergence $D_{\mathrm{KL}}(q_\phi(\theta)||p(\theta|D)) = \mathbb{E}_{q_\phi}[\log(q_\phi(\theta)/p(\theta|D))]$. The latter method, called variational Bayes, is used by the BART model.

3 BART model

The BART model can be defined as a multi-layer network of the following form, as illustrated in Figure 1 in the paper:

$$\mathbf{f}'_1 = (\mathbf{f}_1, \tau(\mathbf{f}_1)); \mathbf{f}'_2 = (\mathbf{f}_2, \tau(\mathbf{f}_2)) \tag{3}$$

$$\mathbf{f}_s = (s(\mathbf{f}'_1), s(\mathbf{f}'_2)) \tag{4}$$

$$\Pr(R_i = 1|\mathbf{f}_s, \mathbf{w}) = 1/[1 + \exp(-\mathbf{f}_s^\top \mathbf{w})]. \tag{5}$$

Below we explain the equations in detail.

(1) *Semantic feature vectors.* For a word a, we can represent it in a d-dimensional semantic space, e.g., $d = 300$, to obtain a feature vector f. In the BART model, the semantic feature vectors for individual words are adopted from an embedding matrix derived by the Word2vec model [13,14]. The embedding matrix V is a $d \times N$ matrix, where N is the number of words in the dictionary. The Word2vec model can be viewed as a two-layer network, including an encoder, $\mathbf{f} = Va$, which encodes the one-hot vector a of the input word into a vector $\mathbf{f}$, and a decoder $g = U^\top \mathbf{f}$, which decodes $\mathbf{f}$ into an N dimensional vector g, where U is another $d \times N$ matrix, to predict the probability distribution of the neighboring words in the text. The embedding matrices (V, U) are learned from big data in a large corpus of text (e.g., Google News corpus). Let $V = (V_1, ..., V_j, ..., V_N)$, where V_j is the j-th column vector of V. If a is the j-th word in the dictionary, its feature vector can be readily produced from the learned embedding matrix as $\mathbf{f} = V_j$.

(2) *Stage 1: Feature alignment (Eq.3)* . For a word a, its embedding $\mathbf{f}$ is a d dimensional vector, and each dimension of $\mathbf{f}$ encodes a different aspect of the semantic meaning of a. For a pair of words (a_1, a_2), let their embeddings be $\mathbf{f}_1 = (f_{11}, ..., f_{1k}, ..., f_{1d})$ and $\mathbf{f}_2 = (f_{21}, ..., f_{2k}, ..., f_{2d})$. For different pairs $(\mathbf{f}_1, \mathbf{f}_2)$, the relation R_i may be relevant to different semantic dimensions. For example, *love* and *hate* are related on a dimension of emotional attitude, but *rich* and *poor* are related on economic status. As a heuristic to align these dimensions, we re-order the dimensions of $(\mathbf{f}_1, \mathbf{f}_2)$ according to the rank of differences between the two vectors. Let $\Delta = (\Delta_1 = f_{11} - f_{21}, ..., \Delta_k = f_{1k} - f_{2k}, ..., \Delta_d = f_{1d} - f_{2d})$ be the element-wise differences. We can order the components of Δ so that $\Delta_{\tau(1)} \geq \Delta_{\tau(2)} \geq ... \geq \Delta_{\tau(d)}$, where $(\tau(i), i = 1, ..., d)$ is a permutation of $(1, ..., d)$. We define $\tau(\mathbf{f}_1) = (f_{1\tau(1)}, ..., f_{1\tau(k)}, ..., f_{1\tau(d)})$ and $\tau(\mathbf{f}_2) = (f_{2\tau(1)}, ..., f_{2\tau(k)}, ..., f_{2\tau(d)})$. We then concatenate $\mathbf{f}_1$ and $\tau(\mathbf{f}_1)$ to get the augmented vector $\mathbf{f}'_1 = (\mathbf{f}_1, \tau(\mathbf{f}_1))$ for the first word in a pair. Similarly we get $\mathbf{f}'_2 = (\mathbf{f}_2, \tau(\mathbf{f}_2))$ for the second word.

If we had much larger training data (rather than just 20 word pairs per relation as in the current paper), the model could in principle learn the alignment re-ordering function τ by some form of attention model. For example, for each pair $(\mathbf{f}_1, \mathbf{f}_2)$, the model could evaluate the functional importance of semantic features for signaling relations between the two words, and pay different amounts of attention to specific dimensions according to their importance.

(3) *Stage 2: Feature selection (Eq.4).* The dimensionality of $(\mathbf{f}'_1, \mathbf{f}'_2)$ may be too high for the available training data (i.e., 20 positive examples per relation). It is preferable to reduce the dimensionality of $(\mathbf{f}'_1, \mathbf{f}'_2)$ in order to avoid overfitting. The dimensionality of $\mathbf{f}'_1 = (\mathbf{f}_1, \tau(\mathbf{f}_1))$ is $2d$, as is the dimensionality of $\mathbf{f}'_2$. Thus there are a total of $4d$ components in $(\mathbf{f}'_1, \mathbf{f}'_2)$ (i.e., 1200 dimensions in the present paper). Among these $4d$ components, we only select a small number of components for predicting a specific relation R_i. Such a variable selection step is accomplished by fitting a ℓ_1 regularized logistic regression on the augmented difference vector $\{(\mathbf{f}'_1 - \mathbf{f}'_2, R_i)\}$, which minimizes $\sum_D \log(1 + \exp(-R_i(\mathbf{f}'_1 - \mathbf{f}'_2)^\top \beta)) + (1 - \lambda)\|\beta\|^2_{\ell_2}/2 + \lambda\|\beta\|_{\ell_1}$, where $\sum_D$ denotes the summation over the training examples, $(\mathbf{f}'_1 - \mathbf{f}'_2)$ is the difference vector, and β is the coefficient vector of the same dimensionality as $(\mathbf{f}'_1 - \mathbf{f}'_2)$. $\|\beta\|_{\ell_1}$ is the ℓ_1 norm of β,(i.e., the sum of absolute values of β), and$\|\beta\|^2_{\ell_2}$ is the ℓ_2 norm of β (i.e., the sum of squared values of β). The ℓ_1 norm is known to induce sparsity of the learned β, so that only a small number of components of β are non-zero. λ is the tuning constant, set to .5 in the BART model. We then select those feature dimensions for which the corresponding components of β are non-zero, and form the vector $\mathbf{f}_s = (s(\mathbf{f}'_1), s(\mathbf{f}'_2))$, where $s()$ stands for feature selection.

(4) *Stage 3: Relational weight distribution through Bayesian learning (Eq.5).* After obtaining selected feature vectors $\mathbf{f}_s$, we then fit a Bayesian logistic regression on $\{(\mathbf{f}_s, R_i)\}$ using BART. The log posterior distribution is $\log p(\mathbf{w}|D) = -\sum_D \log(1 + \exp(-\mathbf{f}_s^T \mathbf{w})) + \log p(\mathbf{w})$, where $p(\mathbf{w})$ is the prior distribution of the weight parameter $\mathbf{w}$. The prior distribution $p(\mathbf{w})$ is defined as a multivariate normal distribution, $\mathrm{N}(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$, with a mean vector $\boldsymbol{\mu_0} = (\beta, -\beta)$, consisting of the weights β estimated in stage 2 for the dimensions linked with the first word, and the opposite weights $-\beta$ for the dimensions linked with the second word. The covariance matrix is defined by an Inverse-Gamma distribution with the two hyperparameters (a, b). The simulation sets the initial values of the two hyperparameters as $(a_0 = 1, b_0 = 5)$, as in earlier work [10].

We approximate $p(\mathbf{w}|D)$ by a variational posterior distribution using the variational Bayes approach. The BART model could be implemented using other inference methods based on Markov chain Monte Carlo sampling, though these would take longer to converge. We employed the variational method developed by Jaakkola and Jordan [28] for Bayesian logistic regression to obtain a closed-form approximation to the posterior distribution for stage 3 in BART model. Variational methods are a family of methods that transform the problem of interest into an optimization problem by introducing an extra variational parameter, which is iteratively adjusted to successively improve approximations. The input to the learning model includes training data with K examples, composed of training data $\mathbf{x}_k$ and their corresponding relation labels R_k in which 1 indicates that the pair of words instantiates the relation (positive examples), and -1 indicates it does not (negative examples). The variational method iteratively updates the mean $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{V}$ for the multivariate normal distribution using the following updating equations during learning:

$$\mathbf{V}^{-1} = \mathbf{A} + 2\sum_k \lambda(\zeta_k)\mathbf{x}_k\mathbf{x}_k^T) \tag{6}$$

$$\boldsymbol{\mu} = \mathbf{V}(\mathbf{A}\boldsymbol{\mu}_0 + \sum_k R_k\mathbf{x}_k/2) \tag{7}$$

$$a = a_0 + 1/2 \tag{8}$$

$$b_i = b_0 + ((w_i - \mu_0 i)^2 + V_{ii})/2 \tag{9}$$

$$\zeta_k{}^2 = \mathbf{x}_k^T(\mathbf{V} + \boldsymbol{\mu}\boldsymbol{\mu}^T)\mathbf{x}_k, \tag{10}$$

where k indicates the kth training example, w_i is the ith element in the weight vector, V_{ii} is the ith diagonal element of covariance matrix, and $\mathbf{A}$ is a diagonal matrix with its ith diagonal

element given by a/b_i. The same updating rule was used by Lu et al. (2012).

The BART model can be viewed as including multiple computational components: semantic feature extraction, feature aligning, feature selection, and relation prediction. If we had a large set of training data or constrained relation domains (e.g., the family tree problem considered by Paccanaro & Hinton [11]), the whole model could in principle be trained end-to-end (i.e., training the word semantic representations and relation representations at the same time). However, when restricted to smaller training data for relations, we may separate the learning of the representational layers. Semantic word features can be learned from long-term experience or large text corpora, whereas the other components can be learned from small data consisting of dozens of word pairs instantiating a relation. The present paper takes the latter approach.

Supplemental Information

Pseudo-code for BART Learning Algorithm

$\boldsymbol{f}_1$, $\boldsymbol{f}_2$ are the input word vectors from Word2vec embeddings

```
for each relation i
        construct positive examples and negative examples
        do < feature augmentation >
                for each word pair j
                        compute difference vector f1 - f2
                        rank order difference features rank(f1 - f2)
                        concatenate difference vectors
                        concatenate raw vectors and ranked vectors
                end for

        do < feature selection >
                input concatenated difference vectors from the training data
                run lassoglm with Elastic-Net regularization
                output selected feature dimensions as s(.) and associated coefficients β
                construct selected feature vectors

        do < variational Bayesian learning >
                input selected feature vectors from the training data, regression coefficients
                compute empirical prior [β, - β]
                run variational Bayesian learning
                output posterior distribution of weights

end for
```

Overview of SemEval-2012 Task-2 Dataset

The norms *(22)* for the 79 relations were created by having an initial group of participants recruited via Amazon Mechanical Turk generate word pairs exemplifying each relation. A small number of paradigmatic examples were provided as seeds to guide participants. For example, the relation *reverse* was exemplified by the pairs *attack-defend*, *buy-sell* and *love-hate*. The set of word pairs generated by the first group was then used to construct a task for a second group of participants, who judged the typicality of examples. Specifically, the second group was assigned the task of choosing the "most illustrative" and "least illustrative" examples of a relation from sets of four alternatives. Approximately 50,000 responses were collected using Mechanical Turk.

Participants' responses were then submitted to an algorithm (MaxDiff, also termed Best-Worst Scaling; see (*25*)) for computing similarities between examples to create a set of word pairs ordered by prototypicality for each of the 79 relations in the dataset. For example, the list for the *taxonomic* relation includes 41 word pairs, ranging from *weapon:spear* and *tree:oak* with high typicality ratings, to *school:university* and *politician:senator* with low typicality ratings. The *contradictory* relation includes *hot:cold* and *rich:poor* with high ratings, and *cold:warm* and *gaseous:solid* with low ratings.

Supplemental Table S1

Examples of relation pairs from norms *(22)*: five most prototypical examples of one specific relation (italics) for each of ten relation types (bold).

Relation type	Examples				
Class inclusion *(Taxonomic)*	weapon:spear	tree:oak	animal:pig	bird:robin	vegetable:carrot
Part-whole *(Object:Component)*	hand:finger	egg:yolk	foot:toe	tree:branch	house:room
Similar *(Synonymity)*	house:home	kid:child	raise:elevate	teach:instruct	big:large
Contrast *(Contradictory)*	hot:cold	full:empty	rich:poor	young:old	dry:wet
Attribute *(Item:Attribute)*	fire:hot	villain:evil	water:wet	tycoon:wealthy	snow:cold
Notattribute *(Item:Nonattribute)*	fire:cold	rainbow: monochromatic	darkness:light	ecstasy:sad	liar:honest
Case relation *(Agent:Object)*	spinner:yarn	weaver:cloth	baker:bread	architect: building	writer:pen
Cause-purpose *(Cause:Effect)*	loss:grief	injury:pain	disease: sickness	explosion: damage	accident:damage
Space-time *(Item:Location)*	library:books	forest:trees	school:student	zoo:animals	garden:flower
Reference *(Sign:Significant)*	crown:royalty	badge:authority	smoke:fire	smile:happiness	license:permission

Supplemental Table S2

UCLA Verbal Analogy Test Problems

Relation: Categorical

A	B	C	D	C	D'
vegetable	cabbage	insect	beetle	insect	frog
insect	bee	fish	halibut	fish	water
flower	rose	bird	pigeon	bird	nest
bird	wren	insect	mosquito	insect	sting
vegetable	lettuce	bird	sparrow	bird	cat
sport	soccer	vehicle	bus	vehicle	engine
weapon	pistol	clothing	shoes	clothing	cotton
sport	tennis	weapon	gun	weapon	murder
furniture	sofa	sport	golf	sport	coat
tool	pliers	vehicle	van	vehicle	fuel
clothing	trousers	fish	cod	fish	net
fruit	banana	furniture	dresser	furniture	house
fish	shark	sport	baseball	sport	team
fruit	plum	clothing	coat	clothing	silk
flower	carnation	tool	hacksaw	tool	carpenter
bird	crow	sport	football	sport	stadium
weapon	sword	flower	daffodil	flower	vase
clothing	jacket	bird	pigeon	bird	dog
furniture	table	fruit	pear	fruit	tree
clothing	jeans	vegetable	potato	vegetable	apple

Relation: Function

A	B	C	D	C	D'
fly	bird	hop	rabbit	hop	leg
build	house	dig	hole	dig	shovel
sing	song	ride	horse	ride	rider
hear	ear	wear	clothes	wear	woman
drive	car	burn	wood	burn	fire
open	door	touch	hands	touch	doctor
squeeze	juice	shoot	gun	shoot	miss
cut	knife	hit	hammer	hit	nail
throw	ball	open	envelope	open	close
read	magazine	play	football	play	kids
carry	suitcase	sit	chair	sit	job
drink	glass	cook	pan	cook	chef
burn	fire	blow	wind	blow	down
ski	snow	swim	water	swim	swimmer

tell	story	fight	battle	fight	soldier
run	horse	pull	tractor	pull	muscle
write	poem	carve	statue	carve	knife
ride	bicycle	drink	cup	drink	water
throw	ball	jump	parachute	jump	walk
ride	elevator	sail	boat	sail	wind

Relation: Opposite

A	**B**	**C**	**D**	**C**	**D'**
artificial	natural	friend	enemy	friend	relative
love	hate	rich	poor	rich	wealthy
alive	dead	succeed	fail	succeed	conquer
ugliness	beauty	joy	sorrow	joy	emotion
huge	tiny	arrive	depart	arrive	come
somber	cheerful	lawyer	client	lawyer	doctor
late	early	win	lose	win	capture
quick	slow	grow	wither	grow	plant
antonym	synonym	selfish	generous	selfish	egocentric
quiet	noisy	last	first	last	final
start	stop	trivial	important	trivial	famous
laugh	cry	cheap	expensive	cheap	inexpensive
accept	reject	dark	bright	dark	dim
abundant	scarce	agree	disagree	agree	concur
hero	coward	strong	weak	strong	muscular
teacher	student	calm	stormy	calm	serene
give	take	allow	forbid	allow	permit
clean	dirty	awake	alseep	awake	alert
remember	forget	increase	decrease	increase	lengthen
crazy	sane	bent	straight	bent	crooked

Relation: Synonym

A	**B**	**C**	**D**	**C**	**D'**
easy	simple	sad	unhappy	sad	happy
hurry	rush	harm	injure	harm	help
rob	steal	cry	weep	cry	laugh
polite	courteous	angry	furious	angry	happy
beginner	novice	doctor	physician	doctor	heal
baby	infant	woman	lady	woman	girl
brook	stream	courage	bravery	courage	cowardice
huge	enormous	wealthy	rich	wealthy	poor
cheerful	happy	awful	horrible	awful	wonderful
remain	stay	begin	start	begin	continue

unusual	strange	famous	renowned	famous	unknown
sick	ill	little	small	little	big
legal	lawful	usual	normal	usual	strange
collect	gather	leave	abandon	leave	stay
clean	neat	fortunate	lucky	fortunate	miserable
immense	colossal	precise	exact	precise	approximate
stone	rock	garbage	trash	garbage	bag
help	aid	raise	lift	raise	lower
rug	carpet	bucket	pail	bucket	milk

Supplemental Figure S1

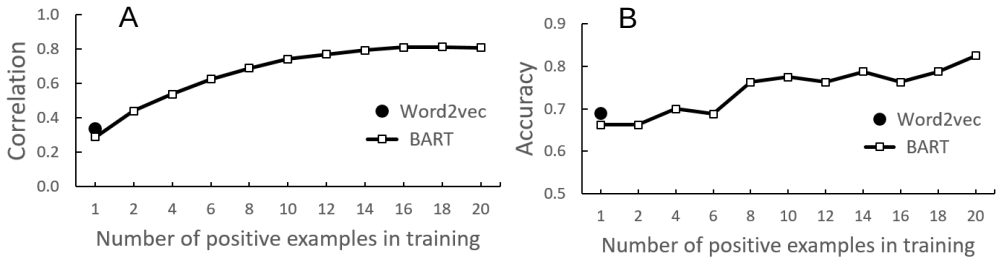

Figure S1. Learning curves for BART. A: Correlation between human typicality ratings and BART predictions as a function of number of positive examples used in training. B: Overall proportion of VAT problems solved correctly by BART as a function of number of positive examples used in relation training. On both graphs Word2vec performance is plotted for comparison.

Comparisons of BART Model with Control Simulations

A number of model variants were implemented and tested in order to isolate the impact of individual components of the BART model on analogy performance. These simulation results are summarized in Figure S2, which presents the overall performance of each model on the VAT set of verbal analogies. In addition to the baseline Word2vec model, we tested a feedforward neural network with a hidden layer. This feedforward neural network included an input layer with 600 units representing semantic features taken from Word2vec for the two words in a pair, a hidden layer with 10 units, and an output layer indicating whether the input word pair instantiates a specific relation. All relations shared the same network architecture, but were trained separately (using Matlab default function of feedforwardnet), with the same inputs as used to train BART. The MATLAB implementation of the feedforward neural network training is shown in the text box below.

```
hidunits = 10;
net = feedforwardnet(hidunits);
net = train(net,X,T);
pred_prob_pos = net(test_data);
```

The feedforward neural network for each relation predicts the probability that the relation is instantiated by an input word pair. These predicted probabilities were then used to create a distributed relation vector (analogous to that formed using BART). To solve each analogy problem, relational similarity was then calculated in the same way as for BART. Analogy performance based on this feedforward network was extremely poor (51%), less than achieved by Word2vec itself.

We also tested five variants of BART. Each was designed to assess the importance of a specific component of the BART model, by altering or removing a single component while keeping the rest of the model identical to full BART. The first two controls targeted the importance of using Bayesian inference with contrast-based priors in the final phase of learning (stage 3). Control 1 replaced BART's stage 3 by regularized regression (i.e., estimating mean weights only, rather than full weight distributions, without priors). Specifically, we removed the computation indicated in the BART pseudo-code by lines 17-21, replacing it with the Matlab function of "lassoglm" to implement the control model. Control 2 used Bayesian regression in stage 3 but with an uninformative prior (zeros as mean values in the prior distribution). Specifically, we used a zero vector to replace line 19 in the BART pseudo-code, while maintaining all the other computations. Controls 1 and 2 achieved correlations with human typicality judgments comparable to full BART, but performance on the verbal analogy test dropped by about 20%.

Controls 3 and 4 respectively assessed the importance of using raw and ranked dimensions in stage 1, in which BART remaps features to address the feature alignment problem. In Control 3, we removed raw vectors and only kept ranked vectors in line 8 of the BART pseudo-code, thereby simulating an otherwise-identical model lacking raw dimensions

(i.e., using ranked dimensions only). The Control 3 simulation showed significantly reduced performance on the verbal analogy test (a decline of 24%). In Control 4, designed to be complementary to the Control 3 simulation, we removed ranked feature dimensions (i.e., using raw dimensions only) in line 8 of the BART pseudo-code. In Control 4 performance on the verbal analogy test was reduced by 9%. These results show that raw dimensions contain information that is very important for solving analogies (Control 3), but that ranked dimensions are also important (Control 4). The BART model is able to provide a partial solution to the feature alignment problem by combining information captured in the raw semantic vectors created by Word2vec with information captured in ranked dimensions.

Control 5 assessed whether the specific semantic space provided by the Word2vec model is important for success in analogical reasoning with abstract relations. We applied a randomly-generated orthogonal transformation to all the Word2vec input vectors (a change that does not affect the performance of Word2vec itself). The BART model was then trained and tested using these transformed feature vectors as inputs, in a manner otherwise identical to the BART simulation using the original Word2vec inputs. Using these transformed inputs caused BART's performance on the VAT to drop by 10%. The results from Control 5 imply that the original Word2vec word space is especially effective in providing semantic knowledge that enables relation learning.

Supplemental Figure S2

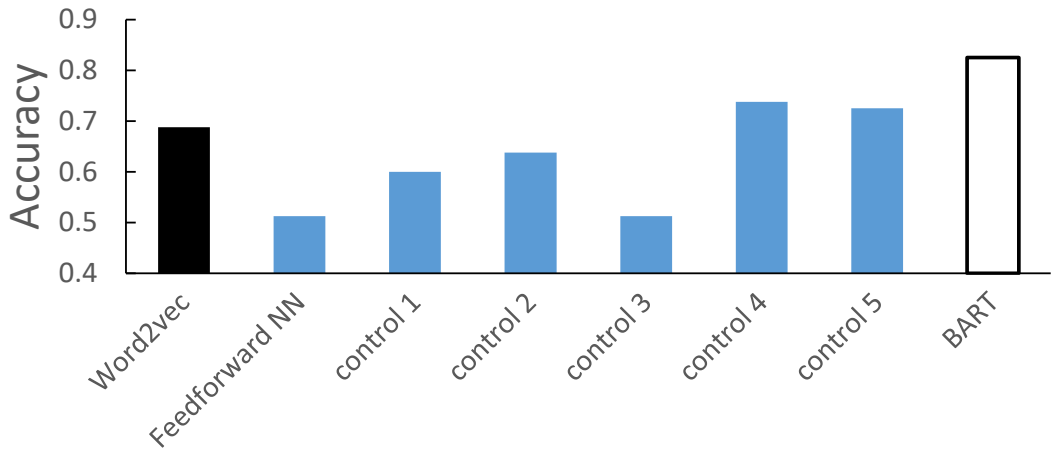

Figure S2. Proportion of correct solutions to UCLA Verbal Analogy Test problems achieved by BART and by several simulations based on control models (see description above).