Supporting Material for "Neural language representations predict outcomes of scientific research"

James P. Bagrow[1,2,*], Daniel Berenberg[3,2], and Joshua Bongard[3,2]

[1]Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
[2]Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
[3]Department of Computer Science, University of Vermont, Burlington, VT, United States
[*]Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

May 17, 2018

Contents

List of figures

List of tables

S1 Datasets

We used the metaBUS data release v2.08 for our corpus of correlate pairs. These data are available for download at http://www.frankbosco.com/data/CorrelationalEffectSizeBenchmarks.html. More up-to-date data are searchable using the metaBUS web interface: http://metabus.org. We also used the 300-dimension (English) word vector representations released by the ConceptNet project called ConceptNet Numberbatch, specifically version 17.06: https://github.com/commonsense/conceptnet-numberbatch.

The correlate texts have already been curated by the metaBUS team but we performed some further processing to connect the individual words (tokens) to terms in Numberbatch. Nonalphanumeric characters were removed, casing was removed, and text was tokenized on whitespace. Tokens were then mapped

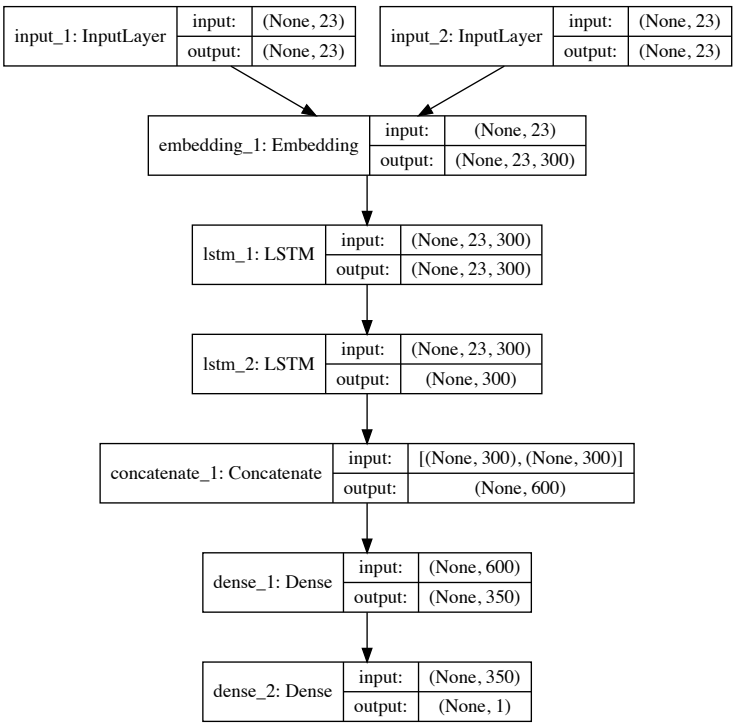


Figure S1: The neural network architecture, visualized using Keras.

to corresponding word vector indices in Numberbatch. A vector "index" of 0 was reserved for tokens in metaBUS not present in Numberbatch. The neural network is able to handle tokens outside the vocabulary of Numberbatch, although predictive performance is likely wose when many tokens are missing than when few or no tokens are missing.

S2 Predictive models

Here we describe the neural networks used in our study.

S2.1 Network Architecture

Our architecture takes two text sequences (the correlate pairs) and outputs their predicted correlation $\hat{r}$. The first layer consists of two inputs, one for each sequence, which lead into a static, untrainable embedding layer that translates each word in each sequence to its 300-dimensional word vector representation. Then, each word vector tensor passes through a stack of two Long Short-Term Memory (LSTM) layers of 300 units. The outputs of the second LSTM are then concatenated and sent though a dense layer of 350 units, which then feeds to a single output unit with a tanh activation. The tanh activation function constrains the network to predict $\hat{r} \in [-1, 1]$. We use ReLU activation functions In the two LSTM and single dense layers preceding the final unit.

S2.2 Training Procedure

Models were trained for a maximum of 200 epochs with a batch size of 1024 using the Adam optimization method [1] with an MSE objective function and a learning rate of 0.001. We randomly split the metaBUS corpus into 80% training and 20% testing, and reserved 10% of the training portion as validation data. The weights of the LSTM layers were initialized using the He normal method [2]. To monitor and avoid overfitting, we apply two different methods: dropout and early stopping. At each trainable layer of the model (all excluding the embedding layer) we apply a dropout rate of 0.15, meaning at each training step approximately 15% of any given layer does not contribute to the prediction at that step, putting more pressure on each individual unit to learn valuable information [3]. Meanwhile, the early stopping mechanism monitors the loss of the model on the validation data: if the validation loss does not improve after 8 consecutive training epochs, the training is terminated and the best model so far is saved, reducing wasted computational time and helping to prevent overfitting [4]. Each of the models was trained using the Keras 2.1.6 Python library with a Tensorflow backend on an NVIDIA Tesla K80 GPU.

S3 Mean-value baseline cannot explain neural network predictions

To determine if the neural network is simply memorizing the correlations present in the training data, we implemented a simple baseline or mean-value procedure: when predicting the correlation between pairs c_i and c_j, simply predict the mean of all previously reported correlations involving either c_i or c_j.

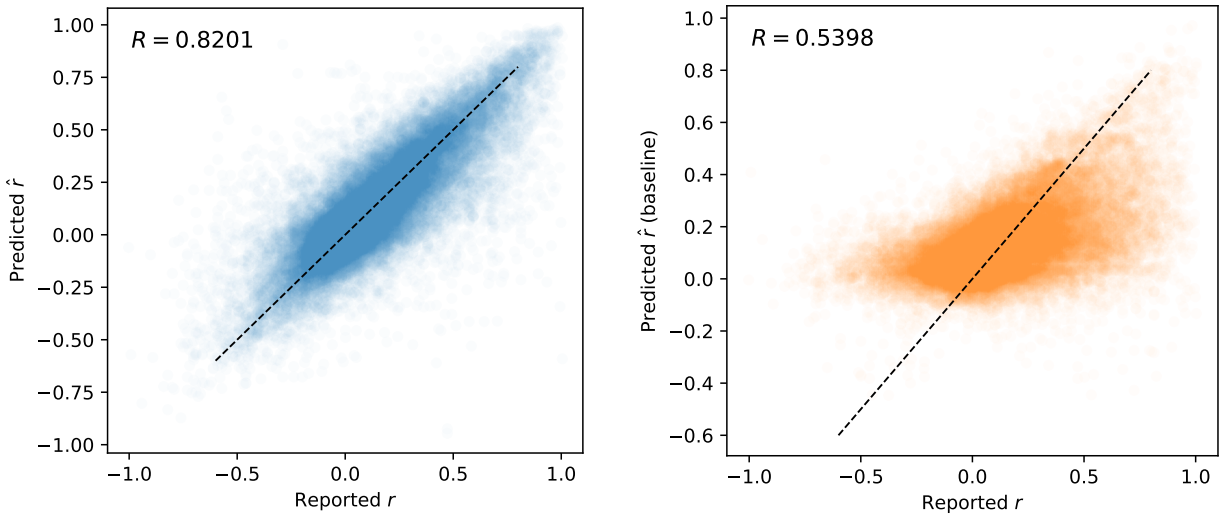
The predictive accuracy of the mean-value baseline ($R = 0.54$) is significantly lower than the neural network ($R = 0.82$) (Fig. S2). Further, the mean-value baseline is not as effective as the neural network at infilling the multi-paper correlation table shown in the main text. We compare their predictions in Figs. S3 and S4. Note that the mean-value infills are computed using the full training corpus, not the reported correlations shown in these 10 papers only. There is considerable information present in the difference between these infilled correlation tables, further illustrating the usefulness of the neural network above that of the mean-value baseline.

Of course, this mean value model can be improved by computing a weighted mean using the similarities of the correlates, for example by cosine distance between their word vector representations, but at that point it is probably more appropriate to use the neural network we applied in the main text.

S4 Using ensemble disagreement to select candidate correlate pairs

We trained an ensemble of $N = 50$ neural networks to build the "committee" used to select candidate correlate pairs in the main text. These models were similar to the main model shown in Fig. S1 but simpler. The primary difference is that they consisted of only one LSTM layer.

To maintain a diverse ensemble of models, a common goal for ensemble learning [5, 6], we took the following steps. Each model was trained on a bootstrap replicate of the complete metaBUS dataset, meaning that an approximately $1-1/e$ fraction of the data of will be out-of-bag for each member of the ensemble. Next before training each model we randomized some of its hyperparameters over a range of values. Specifically we chose for each member a random number of LSTM units between 150 and 250, and a random number of dense units between 100 and 200. We also gave the LSTM layer a random dropout rate between 0.1 and 0.2, and also the dense layer's dropout rate was randomly chosen between 0.1 and 0.2. A batch size of 512 was used for training and a 10% validation sample was reserved for early stopping (3 epochs as opposed to 8 for



(a) Neural network.

(b) Mean-value baseline

Figure S2: The mean-value baseline model (b) performs significantly worse at prediction than the neural network (a) we used in the main text. It performs especially poorly in comparison for negative correlations.

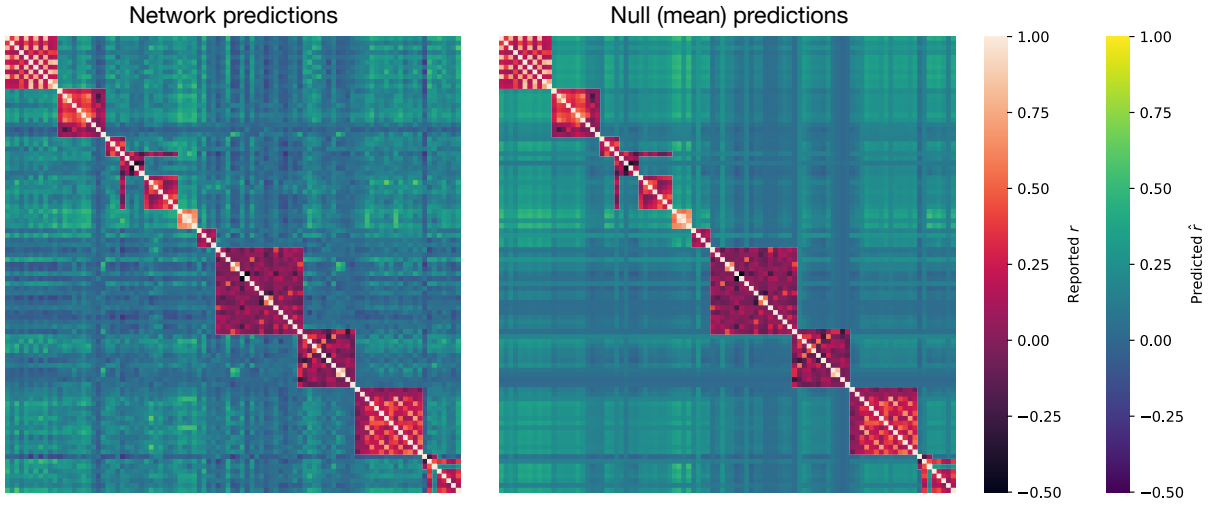


Figure S3: The mean value baseline provides less information to infill multi-paper correlation tables than the neural network.

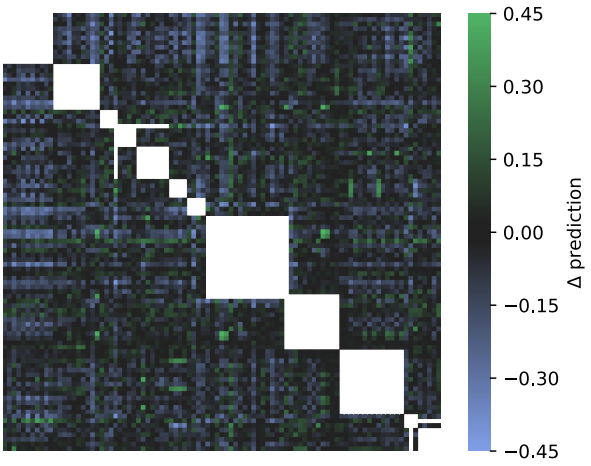


Figure S4: Difference in infilled predictions between mean-value baseline and neural network. This figure shows the difference between the matrices shown in Fig. S3. Positive values indicate the neural network predicts higher correlations than the baseline, negative values indicate the baseline predicts higher values than the neural network. We see distinct patterns in the difference, underscoring the presence of information in the neural network predictions not present in the mean-value baseline.

the main model). No activation function was used on the output unit, as opposed to the tanh function used for the main model.

Table S1 shows the correlates with the 1% most ensemble disagreement. We see a number of interesting pairs just from the small (n = 5000 pairs; main text) search.

Table S1: Candidate correlate pairs with ensemble disagreement in the Top 1%.

Correlate 1	Correlate 2	E[$\hat{r}$]	$\sigma(\hat{r})$
Aggression-supervisor	Video characteristics: Number of concrete statements	0.033091	0.240711
actor trust	Mean rating (T2) Fairness condition Middle performer	0.324968	0.225283
Novice completion time (Task 1, in seconds)	Organization embeddedness	-0.040126	0.220755
Experimental 3- item loading assumption	Physical (compensation scale)	0.167164	0.202105
Obstructionism	Meeting format	-0.083101	0.199643
Counterfactual thinking regarding treatment toward self	Value of surveys	-0.104113	0.197323
Job analysis: Detail of descriptor	Mean rating (T2) Motivating condition Middle performer	0.062453	0.185485
Interpersonal affect	Global job satisfaction (1-5)	0.306790	0.184210
Counterfactual thinking regarding treatment toward self	Preview information	-0.098595	0.180608
Anticipated regret about persistence (-10 to + 10)	Bases of power: Referent (field training)	0.120924	0.180003
Social forces	Desire (discretion)	0.032081	0.178413
Neuroticism	Instructor characteristics: Teaching skill	-0.184349	0.176696
Behavior Commission	Instructor characteristics: Completion time on task (in minutes)	-0.095671	0.174231
G coefficient variance component (z score)	Job Satisfaction	-0.082509	0.173581
Behavioral intentions	Recommendations made	-0.009320	0.173255
Manipulation check PA	Hardworking	0.451182	0.172144
Correlational accuracy	Satisfaction	-0.041887	0.171874
Modem Racism Scale (MRS)	Exhaustion	-0.059882	0.171266
Trust (T1)	Military invasive	-0.008274	0.170604
Organizational tenure	On-time retirement	0.105206	0.167623
Gross domestic product	Time 2 variables: Job search behavior (unemployed group)	-0.376718	0.167058
Mean rating (T1) Fairness condition Middle performer	Performance (Time 2) (T2)	0.236572	0.166747
Work satisfaction	Simple Nonstrategic Actions (P)	-0.237742	0.166404
Competition	GRO X Gender	0.124175	0.166231
course-content variables: Instructor experience	Number of basic statements	-0.076577	0.166137
Noncompliant behavior	Diary-keeping condition: Task condition	-0.167971	0.166024
Sum of PcVc	Skill transfer	-0.039455	0.165213
Intent to leave	Unit (Participants = fixed hourly pay system)	-0.032186	0.163733
Recommendations made	Job analysis: Procedures for developing	0.050139	0.162057
Direct contact	Justification Question l	0.140842	0.161316
Intensity	Verdict ratings	-0.025061	0.161030

Continued on next page

Correlate 1	Correlate 2	$E[\hat{r}]$	$\sigma(\hat{r})$
Accuracy measure: Borman's differential accuracy	Biodata inventory (Difference Score)	0.143893	0.160265
G coefficient variance component (z score)	Research implementation	-0.028003	0.159969
Work intensity	Instructor characteristics: Expertise	0.182853	0.159930
Outcome expectancy	Emotional demands (T2)	-0.026845	0.159219
Competency modeling: Ranking descriptor	Socialization (CPI)	0.276968	0.158227
Conscientiousness	Cognitive X Physical	-0.040164	0.157618
Radiotherapy assistant	Prosocial Motivation X Manager Integrity X Dispositional Trust Propensity	-0.105287	0.157335
Standard Occupational Classification (SOC) major group	Perceived interactional justice	-0.430987	0.157139
Reaction to peer ratings for evaluation (wage) purposes	True/false Scoially Desirable Responding Score (Honest condition)	0.137170	0.157074
Average monitoring intensity	Average exam performance	-0.052568	0.156633
Direct commitment (easy goal condition)	Skilled Trades	0.508497	0.156310
Organizational attract. T3	Grip-Left (Female Sample)	0.217855	0.156064
Overall handshake	Feels superior	0.344592	0.155691
DCS: W (Time 1)	Ratings of others' contribution (RO)	-0.091074	0.155611
Expected utility of withdrawal	Organizational Commitment Questionnaire (OCQ) Item 14	-0.342942	0.154830
HS/T3	Team commitment	0.083982	0.154235
Performance	Supervisor: L Perf/L Inc	-0.135945	0.153173
NEO Personality Inventory-Revised: Angry Hostility Scale (A-H)	Role perceptions Help-S (helping aimed at the supervisor)	0.053814	0.153112
NEO Personality Inventory-Revised: Angry Hostility Scale (A-H)	Flexibility (In-basket)	0.147563	0.152931

References

[1] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. S3

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015. S3

[3] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014. S3

[4] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Advances in neural information processing systems*, pp. 402–408, 2001. S3

[5] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," in *Advances in neural information processing systems*, pp. 190–196, 1996. S3

[6] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000. S3