Learning to Negate Adjectives with Bilinear Models

Anonymous EACL submission

Abstract

We learn a mapping that negates adjectives by predicting an adjective's antonym in an arbitrary word embedding model. We show that both linear models and neural networks improve on this task when they have access to a vector representing the semantic domain of the input word, e.g. a centroid of temperature words when predicting the antonym of 'cold'. We introduce a continuous class-conditional bilinear neural network which is able to negate adjectives with high precision.

1 Introduction

Identifying antonym pairs such as *hot* and *cold* in a vector space model is a challenging task. Previous work has explored learning or retrofitting specialised embeddings that push antonyms further apart than synonyms (Pham et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2016), or using unsupervised measures to distinguish antonym pairs from other binary lexical relations (Santus et al., 2015). However, these approaches do not offer a negation mapping which predicts an antonym for a given word, in an arbitrary word embedding model.

In this paper we learn such a mapping. We focus on negating adjectives, and treat negation as prediction of a one-best antonym. For example, given the expression *not talkative* and the vector $\overrightarrow{talkative}$, the mapping should return a word from the set *quiet, taciturn, uncommunicative*, etc. Antonym pairs share a domain—e.g. *temperature*, but differ in their value—e.g. *coldness* (Turney, 2012; Hermann et al., 2013). Negation must alter the value while retaining the domain.

In this paper we exploit the semantic neighbourhood of an adjective as a stand-in for the domain. We hypothesize that the relevant features for negating, say, a temperature adjective, differ from those for an emotion adjective. Therefore, our negation mappings make use of a vector at the centroid of words related to the input. This approach is inspired by Kruszewski et al. (2016), who find that nearest neighbours in a vector space are a good approximation for human judgements of possible alternatives to negated nouns.

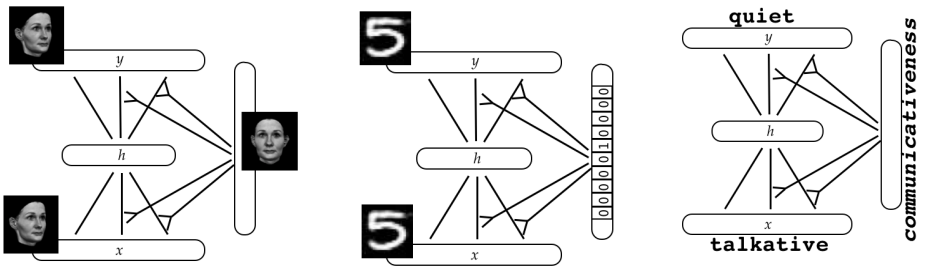
We also introduce a variant of a bilinear relational neural network architecture which has proven successful in identifying image transformations in computer vision. Our model outperforms several baselines on a multiple choice antonym selection task, and learns to produce a one-best antonym with high precision.

2 Relational Encoders

2.1 Relational Autoencoders: Background

Relational autoencoders (RAE), also known as gated autoencoders (GAE), have been used in computer vision to learn representations of transformations between images, such as rotation or translation (Memisevic and Hinton, 2007; Memisevic, 2012, 2013). RAEs are a type of *gated network*, which contains multiplicative connections between two related inputs. The "gating" of one image vector by another allows feature detectors to concentrate on the correspondences between the related images, rather than being distracted by the differences between untransformed images. See Figure 1(a). Multiplicative connections involve a weight for every pair of units in the input vector and gate vector. For an overview of RAEs see Memisevic (2013); Sigaud et al. (2015).

RAE gates perform a somewhat different function than LSTM gates (Hochreiter and Schmidhuber, 1997). Both architectures use a nonlinearity to modulate the contents of a product; in an RAE this is an outer (bilinear) product while in an LSTM it



(a) Relational Autoencoder (b) CCRAE (c) CCCRE

Figure 1: Neural network architectures and training signal for (a) RAE (Memisevic, 2013), (b) Class-Conditional RAE Rudy and Taylor (2015), and Continuous Class-Conditional RAE (this paper). Figures based on Memisevic (2013).

is a Hadamard (element-wise) product. However, LSTM memory gates represent an internal hidden state of the network, while RAE gates are part of the network input.

An Autoencoder (AE) can be defined as in Eq 1 (we omit bias terms for simplicity), where $\mathrm{W_e}$ are the encoder weights and $\mathrm{W_d}$ are the decoder weights. In autoencoders, weights are typically tied so that $\mathrm{W_d} = {\mathrm{W_e}}^T$.

$$\begin{aligned} h &= f(x) = \sigma(\mathrm{W_e}x) \\ y &= g(h) = \mathrm{W_d}h \end{aligned} \tag{1}$$

For an RAE, we have two inputs x and z. Instead of a weight matrix W we have a weight tensor $\overline{\mathrm{W}} \in R^{n_H \times n_X \times n_Z}$. The RAE is defined in Eq 2.

$$\begin{aligned} h &= f(x,z) = \sigma((\overline{\mathrm{W_e}}z)x) \\ y &= g(h,z) = \sigma((\overline{\mathrm{W_d}}h)z) \end{aligned} \tag{2}$$

Rudy and Taylor (2015) introduce a class-conditional gated autoencoder in which the gate is a one-hot class label, rather than a transformed version of the input image. For example, in the MNIST task the label represents the digit. Effectively, an autoencoder is trained per class, but with weight sharing across classes. See Figure 1(b).

2.2 Continuous Class-Conditional Relational Encoders

Our bilinear model is a continuous class-conditional relational encoder (CCCRE). The model architecture is the same as an RAE with untied encoder and decoder weights. However, the training signal differs from a classic RAE in two ways. First, it is not an autoencoder, but simply an encoder, because it is not trained to reproduce the input but rather to transform the input to its antonym. Second, the encoder is class-conditional in the sense of Rudy and Taylor (2015), since the gate represents the class. Unlike the one-hot gates of Rudy and Taylor (2015), our gates are real-valued, representing the semantic domain of the input vector. See Figure 1(c). The intuition behind our model is that it learns to negate a word in the context of that word's semantic domain.

3 Experiments

3.1 Models

We compare the CCCRE with several baselines. The simplest is cosine similarity in the original vector space. We train a linear model (**Linear**) which maps the input word to its antonym, an Untied Encoder (**UE**) with a bottleneck hidden layer, and a shallow feed-forward model (**FF**) with a wide hidden layer rather than a bottleneck. To test whether the semantic domain is helpful in learning negation, each of these models has a **Concat** version in which the input consists of the concatenated input word and gate vectors.

3.2 Experimental Settings

We use publicly-available[1] 300-dimensional embeddings trained on part of the Google News dataset using skip-gram with negative sampling (SGNS) (Mikolov et al., 2013). Antonym training data was obtained from WordNet (Miller, 1995) (hereafter WN), resulting in approximately 20K training pairs. We exclude any antonym pair with the input word in the test set.

Gate vectors were obtained under three conditions. In the **standard** condition we use all WN cohyponyms of an input word. If there are fewer than ten, we make up the difference with nearest

[1] https://code.google.com/archive/p/word2vec/

neighbours from the vector space. The gate vector is the vector centroid of the resulting word list.

In the **unsupervised** gate condition we do not use WN, but rather the ten nearest neighbours from the vector space. In the **restricted** condition, we use standard gates but remove all WN cohyponyms of test input words from the training data, e.g. *hot, cool, tepid* etc. if *cold* is a test word. The input word vector is never part of the centroid, and we use the same gate type at training and test time.

Hyperparameters were tuned on the GRE development set (Sec 3.3). All models were optimized using AdaDelta ($\rho = 0.95$) to minimize Mean Squared Error loss. The FF and CCCRE networks have hidden layers of 600 units, while UE has 150 and UE-Concat has 300. Minibatch size was 48 for CCCRE and 16 for all other networks. The linear models were trained for 100 epochs, FF networks for 400, UE for 300, and CCCRE for 200.

3.3 Evaluation

Experiment 1 uses the Graduate Record Examination (GRE) questions of Mohammad et al. (2013). The task, given an input word, is to pick the best antonym from five options. An example is shown in (3), where the input word is *piquant* and the correct answer is *bland*. We use only those questions where both input and target are adjectives.

piquant: (a) shocking (b) jovial (c) rigorous (d) merry (e) **bland** (3)

We evaluate a model by predicting an antonym vector for the input word, and choosing the multiple choice option with the smallest cosine distance to the predicted vector. We report accuracy, i.e. percentage of questions answered correctly.

Experiment 2 evaluates the precision of the models. A natural criterion for the success of a negation mapping is whether the model returns a good antonym at rank 1, or several good antonyms at rank 5, rather than returning any particular antonym as required by the GRE task.

We use two datasets: the GRE test set (**GRE**), and a set of 99 adjectives and their antonyms from a crowdsourced dataset collected by Lenci and Benotto acccording to the guidelines of Schulte im Walde and Köper (2013) (**LB**). For each input word we retrieve the five nearest neighbours of the model prediction and check them against a gold standard. Gold standard antonyms for a word include its antonyms from the test sets and WN. Following Gorman and Curran (2005), to minimise false negatives we improve the coverage of the gold standard by expanding it with antonyms from Roget's 21st Century Thesaurus, Third Edition.[2]

	Training Condition		
Method	**Stand.**	**Unsup.**	**Restr.**
Random	0.20	—	—
Cosine	0.50	—	—
Linear	0.56	0.56	0.53
Linear-Concat	0.66	0.59	0.63
UE	0.57	0.55	0.52
UE-Concat	0.63	0.58	0.61
FF	0.58	0.54	0.51
FF-Concat	0.65	0.56	0.63
CCCRE	**0.69**	**0.60**	**0.65**

Table 1: Accuracy on the 367 multiple-choice adjective questions in the GRE test set.

4 Results and Discussion

Table 1 shows the results of Experiment 1. A random baseline results in 0.20 accuracy. The cosine similarity baseline is already fairly strong at 0.50, suggesting that in general about two out of the five options are closely related to the input word.

Information about the semantic domain clearly provides useful information for this task, because the **Concat** versions of the Linear, UE, and FF models achieve several points higher than the models using only the input word. The linear model achieves a surprisingly high 0.66 accuracy under standard training conditions.

CCCRE achieves the highest accuracy across all training conditions, and is the only model that beats the linear baseline, suggesting that bilinear connections are useful for antonym prediction.

All the models show a notable loss of accuracy in the **unsupervised** condition, suggesting that the alternatives found in the vector neighbourhood are less useful than supervised gates. Even in this setting, however, CCCRE achieves a respectable 0.60. In the **restricted** condition, all non-Concat models perform near the cosine baseline, suggesting that in the standard setting they were memorising antonyms of semantically similar words. The Concat models and CCCRE retain a higher level of accuracy, indicating that they can generalise across different semantic classes.

Although CCCRE achieves the highest accuracy in Experiment 1, the GRE task does not really reflect our primary goal, namely to negate adjectives by generating a one-best antonym. CCCRE sometimes fails to choose the target GRE

[2] http://thesaurus.com

	GRE						LB					
	Stand.		Unsup.		Restr.		Stand.		Unsup.		Restr.	
Method	**P@1**	**P@5**	**P@1**	**P@5**	**P@1**	**P@5**	**P@1**	**P@5**	**P@1**	**P@5**	**P@1**	**P@5**
Cosine	0.05	0.07	—	—	—	—	0.13	0.10	—	—	—	—
Linear	0.36	0.29	0.34	0.29	0.32	0.28	0.29	0.25	0.30	0.24	0.29	0.23
Linear-Concat	0.39	0.33	0.43	0.34	0.36	0.31	0.33	0.28	0.31	0.27	0.32	0.27
UE	0.38	0.33	0.36	0.32	0.37	0.31	0.28	0.22	0.27	0.23	0.23	0.20
UE-Concat	0.38	0.33	0.43	0.38	0.27	0.31	0.33	0.28	0.34	0.27	0.28	0.25
FF	0.37	0.32	0.34	0.30	0.08	0.15	0.30	0.24	0.27	0.23	0.22	0.19
FF-Concat	0.36	0.30	0.46	0.40	0.37	0.34	0.34	0.26	0.28	0.26	**0.34**	0.27
CCCRE	**0.66**	**0.49**	**0.52**	**0.42**	**0.52**	**0.38**	**0.39**	**0.32**	**0.46**	**0.32**	**0.34**	**0.30**

Table 2: Precision at ranks 1 and 5 on the GRE and Lenci and Benotto datasets.

Method	**Top 5 Predictions**	
CCCRE	ornate:	**unadorned, inelegant, banal**, oversweet, **unembellished**
	ruthless:	**merciful, compassionate, gentle, righteous, meek**
FF-Concat	ornate:	**unadorned, unornamented**, overdecorated, **elegant**, sumptuousness
	ruthless:	merciless, heartless, **meek, merciful**, unfeeling

Table 3: Samples of top five nearest neighbours of predicted antonym vectors for CCCRE and FF-Concat.

antonym, but still makes a good overall prediction. For input word *doleful*, the model fails to choose the GRE target word *merry*, preferring instead *sociable*. However, the top three nearest neighbours for the predicted antonym of *doleful* are *joyful, joyous*, and *happy*, all very acceptable antonyms.

Table 2 shows the results of Experiment 2. On the GRE dataset, under standard training conditions, CCCRE achieves an impressive P@1 of 0.66, i.e. two thirds of the time it is able to produce an antonym of the input word as the nearest neighbour of the prediction. All of the other models score less than 0.40. In the **unsupervised** and **restricted** training conditions CCCRE still predicts a one-best antonym about half the time.

The LB dataset is more challenging, because it contains a number of words which lack obvious antonyms, e.g. *taxonomic, quarterly, morphological*, and *fiscal*. However, CCCRE still achieves the highest precision on this dataset. Interestingly, precision does not suffer as much in the less supervised trianing conditions, and P@1 even improves with the **unsupervised** nearest neighbour gates. We speculate that nearest distributional neighbours correspond better than the WN ontology to the crowdsourced antonyms in this dataset.

Table 3 shows sample predictions for the CCCRE and FF-Concat models. It can be seen that CCCRE has more antonyms at the highest ranks.

5 Related Work

Pham et al. (2015); Nguyen et al. (2016); Mrkšić et al. (2016) use WN and other ontologies to retrofit embeddings, or learn embeddings with a secondary objective of pushing antonyms further apart than synonyms. Santus et al. (2014a,b, 2015) perform unsupervised synonym-antonym discrimination using the relative salience of shared features. Mohammad et al. (2013) use a supervised thesaurus-based method on the GRE task. Pham et al. (2015) learn negation as a linear map, and find that it is more effective at predicting a one-best antonym when vectors have been specially trained for lexical contrast.

RAEs and related architectures have been used in computer vision for a number of applications including recognising transformed images (Memisevic and Hinton, 2007), recognising actions (Taylor et al., 2010), learning invariant features from images and videos (Grimes and Rao, 2005; Zou et al., 2012), and reconstructing MNIST digits and facial images (Rudy and Taylor, 2015). Wang et al. (2015) use RAEs for tag recommendation, but RAEs have not been widely used in NLP.

6 Conclusion

We have shown that a representation of the semantic domain improves antonym prediction in linear and non-linear models, and that the multiplicative connections in a bilinear model are effective at learning to negate adjectives with high precision.

Our implementation uses a full tensor for the bilinear layer, and takes several hours to train on a GPU. Future work will exploit options for reducing the number of parameters to be learned (Alain and Olivier, 2013) and address negation of nouns and verbs, where negation is more likely to involve a set of alternatives than a one-best antonym.

References

Droniou Alain and Sigaud Olivier. 2013. Gated autoencoders with tied input weights. In *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA.

James Gorman and James R. Curran. 2005. Approximate searching for distributional similarity. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 97–104, Ann Arbor.

D. Grimes and R. Rao. 2005. Bilinear sparse coding for invariant vision. *Neural Computation*, 17(1):4773.

Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. 2013. "Not not bad" is not "bad": A distributional account of negation. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 74–82, Sofia, Bulgaria.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Germán Kruszewski, Denis Paperno, Raffaella Bernardi, and Marco Baroni. 2016. There is no logical negation here, but there are alternatives: Modeling conversational negation with distributional semantics. *Computational Linguistics*, 42.

Roland Memisevic. 2012. On multi-view feature learning. In *Proceedings of ICML*.

Roland Memisevic. 2013. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1829–1846.

Roland Memisevic and Geoffrey Hinton. 2007. Unsupervised learning of image transformations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing lexical contrast. *Computational Linguistics*, 39 (3):555–590.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL*, pages 142–148, San Diego.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonymsynonym distinction. In *Proceedings of ACL*, pages 454–459, Berlin.

N. Pham, A. Lazaridou, and M. Baroni. 2015. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of ACL*, pages 21–26.

Jan Rudy and Graham Taylor. 2015. Generative class-conditional denoising autoencoders. In *Proceedings of ICLR Workshop*.

Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. 2014a. Unsupervised antonym-synonym discrimination in vector space. In *Atti della Conferenza di Linguistica Computazionale Italiana (CLIC-IT 2014)*, Pisa, Italy.

Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. 2014b. Taking antonymy mask off in vector space. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC 2014)*, Phuket, Thailand.

Enrico Santus, Alessandro Lenci, Qin Lu, and Chu-Ren Huang. 2015. When similarity becomes opposition: Synonyms and antonyms discrimination in DSMs. *Italian Journal of Computational Linguistics*, 1(1):41–54.

Sabine Schulte im Walde and Maximilian Köper. Pattern-based distinction of paradigmatic relations for German nouns, verbs, adjectives. In *Language Processing and Knowledge in the Web*, pages 184–198. 2013.

Olivier Sigaud, Clément Masson, David Filliat, and Freek Stulp. 2015. Gated networks: an inventory. arXiv:1512.03201 [cs.LG].

G. Taylor, R. Fergus, Y. LeCun, and C. Bregler. 2010. Convolutional learning of spatio-temporal features. In *Proceedings of*

the European Conference on Computer Vision (ECCV10).

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2015. Relational stacked denoising autoencoder for tag recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*.

W. Y. Zou, S. Zhu, A. Y. Ng, and K. Yu. 2012. Deep learning of invariant features via tracked video sequences. In *Advances in Neural Information Processing Systems 25*.