Learning to Negate Adjectives with Bilinear Models

Laura Rimell
University of Cambridge
laura.rimell@cl.cam.ac.uk

Amandla Mabona
University of Cambridge
amandla.mabona@cl.cam.ac.uk

Luana Bulat
University of Cambridge
ltf24@cam.ac.uk

Douwe Kiela
Facebook AI Research
dkiela@fb.com

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, because synonyms and antonyms are both distributionally similar (Grefenstette, 1992; Mohammad et al., 2008). Recent work on antonymy has learned specialized word embeddings using a lexical contrast objective to push antonyms further apart in the space (Pham et al., 2015; Ono et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2016), which has been shown to improve both antonym detection and the overall quality of the vectors for downstream tasks. In this paper we are interested in a related scenario: given an arbitrary word embedding model, with no assumptions about pre-training for lexical contrast, we address the task of **negation**, which we define as the prediction of a one-best antonym for an input word. For example, given the word *talkative*, the negation mapping should return a word from the set *quiet, taciturn, uncommunicative*, etc.

We focus on the negation of adjectives. The intuition behind our approach is to exploit a word's semantic neighborhood to help find its antonyms. Antonym pairs share a domain, or topic—e.g. *temperature*; but differ in their value, or polarity—e.g. *coldness* (Turney, 2012; Hermann et al., 2013). Negation must alter the polarity while retaining the domain information in the word embedding. We hypothesize that a successful mapping must be conditioned on the domain, since the relevant features for negating, say, a temperature adjective, differ from those for an emotion adjective. Inspired by Kruszewski et al. (2016), who find that nearest neighbors in a vector space are a good approximation for human judgements about negation, we represent an adjective's domain by the centroid of nearest neighbors in the embedding space or cohyponyms in WordNet.

We introduce a novel variant of a bilinear relational neural network architecture which has proven successful in identifying image transformations in computer vision (Memisevic, 2012; Rudy and Taylor, 2015), and which learns a negation mapping conditioned on a gate vector representing the semantic domain of an adjective. Our model outperforms several baselines on a multiple choice antonym selection task, and learns to predict a one-best antonym with high precision. In addition to the negation task, this model may be of interest for other NLP applications involving lexical or discourse relations.

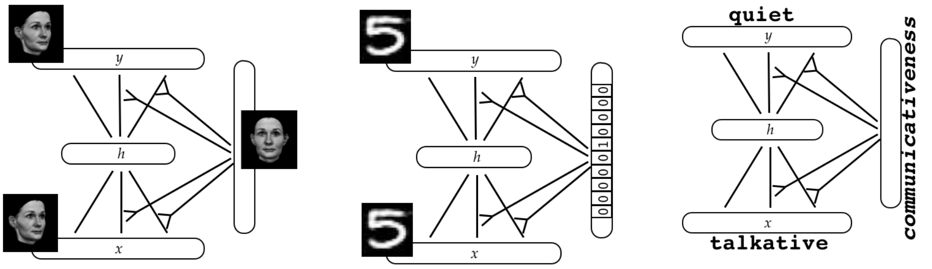
2 Relational Encoders

Our task is to map a word embedding vector x, e.g. *hot*, to an antonym vector y in the same space, e.g. *cold*, conditioned on the semantic domain, which is represented by a vector z (see Sec 3.2 for how this vector is obtained). We learn this mapping using a relational neural network, which we introduce in the following sections.

2.1 Relational Autoencoders: Background

Relational autoencoders (RAE), also known as gated autoencoders (GAE), have been used in

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 71–78,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics



(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 RE (this paper). Figures based on Memisevic (2013).

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) and 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 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.

$$\begin{aligned} h &= f(x, z) = \sigma((\overline{\mathrm{W}_\mathrm{e}}z)x) \\ y &= g(h, z) = \sigma((\overline{\mathrm{W}_\mathrm{d}}h)z) \end{aligned} \tag{2}$$

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 (Eq 2). 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). Analogous to the case of image transformation detection, we want the model to learn the changes relevant to negation without being distracted by cross-domain differences.

We approximate the semantic domain as the centroid of a set of related vectors (see Sec 3.2). This approach is inspired by Kruszewski et al. (2016), who investigate negation of nouns, which typically involves a set of alternatives rather than an antonym. It is natural to finish the statement *That's not a table, it's a ...* with *desk* or *chair*, but not *pickle*. Kruszewski et al. (2016) find that nearest neighbors in a vector space are a good approximation for human judgements about alternatives. We hypothesize that a set of alternatives can stand in for the semantic domain. Note that each word has its own domain, based on its WordNet or distributional neighbors; however, similar words will generally have similar gates.

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 (Eq 3),

$$y = \mathrm{W}x \tag{3}$$

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 (both as in Eq 1 with different hidden layer sizes). 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 $x||z$, rather than x.

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. Training data always excludes antonym pairs where the input word is an input word the test set. Exclusion of pairs where the target word is a target in the test set depends on the training condition.

Gate vectors were obtained under two conditions. In the **standard** condition we begin with all WN cohyponyms of an input word. If there are fewer than ten, we make up the difference with nearest neighbors from the vector space. The gate vector is the vector centroid of the resulting word list. In the standard training condition, we do not exclude antonym pairs with the target word in the test set, since we hypothesize it is important for the model to see other words with a similar semantic domain in order to learn the subtle changes necessary for negation. For example, if the pair (*hot, cold*) is in the test set, we exclude (*hot, cold*), (*hot, freezing*), etc. from training; but we do not exclude (*icy, hot*) or (*burning, cold*) from training.

In the **unsupervised** gate condition we do not use WN, but rather the ten nearest neighbors from the vector space. Note that it is only the gates which are unsupervised, not the word pairs: the training targets are still supervised.

We also use a **restricted** training condition, to test whether it is important for the model to have training examples from a similar semantic domain to the test examples. E.g. if (*hot, cold*) is in the test set, is it important for the model to have other temperature terms in the training data? We remove all WN cohyponyms of test input words from the training data, e.g. *hot, cool, tepid* etc. if *cold* is a test input word. Although we do not explicitly remove training examples with the target word in the test set, these are effectively removed by the nature of the semantic relations. We use standard (supervised) gates in this condition.

In all conditions, the input word vector is never part of the gate 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 (4), 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** (4)

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

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

	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.

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 neighbors 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 minimize false negatives we improve the coverage of the gold standard by expanding it with antonyms from Roget's 21st Century Thesaurus, Third Edition.[2]

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-Concat 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 neighborhood 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 memorizing antonyms of semantically similar words. The Concat models and CCCRE retain a higher level of accuracy, indicating that they can generalize across different semantic classes.

We are unable to compare directly with previous results on the GRE dataset, since our evaluation is restricted to adjectives. As an indicative comparison, Mohammad et al. (2013) report an F-score of 0.69 on the full test dataset with a thesaurus-based method, while Zhang et al. (2014) report an F-score of 0.62 using a vector space induced from WN and distributional vectors, and 0.82 with a larger thesaurus. (Previous work reported F-score rather than accuracy due to out-of-coverage terms.)

Although CCCRE achieves the highest accuracy in Experiment 1, the GRE task does not reflect our primary goal, namely to negate adjectives by generating a one-best antonym. CCCRE sometimes fails to choose the target GRE 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 neighbors 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 neighbor 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, psychiatric*, and *biblical*. However, CCCRE still achieves the highest precision on this dataset. Interestingly, precision does not suffer as much in the less supervised training conditions, and P@1 even improves with the **unsupervised** nearest neighbor gates. We speculate that nearest distributional neighbors correspond better than the WN ontology to the crowdsourced antonyms in this dataset. LB antonyms for

[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 neighbors of predicted antonym vectors for CCCRE and FF-Concat.

psychiatric include *normal, well, sane,* and *balanced*. The **unsupervised** model predicts *sane* as the top neighbor, while **standard** predicts *psychiatrists*. The sense in which *sane* is an antonym of *psychiatric* is an extended sense, of a form unlikely to be found in WN training data.

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

Previous work on negation has focused on pattern-based extraction of antonym pairs (Lin et al., 2003; Lobanova, 2012). Such bootstrapped lexical resources are useful for the negation task when the input words are covered. Turney (2008); Schulte im Walde and Köper (2013); Santus et al. (2014, 2015) use pattern-based and distributional features to distinguish synonym and antonym pairs.

Schwartz et al. (2015) build a vector space using pattern-based word co-occurrence, which can be tuned to reduce the cosine similarity of antonyms. Yih et al. (2012); Chang et al. (2013) use LSA to induce antonymy-sensitive vector spaces from a thesaurus, while Zhang et al. (2014) use tensor decomposition to induce a space combining thesaurus information with neural embeddings. Pham et al. (2015); Ono et al. (2015); Nguyen et al. (2016) learn embeddings with an objective that increases the distance between antonyms, while Nguyen et al. (2016); Mrkšić et al. (2016) reweight or retrofit embeddings to fine-tune them for antonymy. Our approach differs in that we learn a negation mapping in a standard embedding space.

Mohammad et al. (2013) use a supervised thesaurus-based method on the GRE task. Pham et al. (2015) learn negation as a linear map, finding it more accurate at predicting a one-best antonym when using vectors trained for lexical contrast.

RAEs and related architectures have been used in computer vision for a number of applications including recognizing transformed images (Memisevic and Hinton, 2007), recognizing 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 to our knowledge RAEs have not been previously 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.

One direction for future improvement is to make the model more efficient to train, by reducing the number of parameters to be learned in the relational network (Alain and Olivier, 2013). Future work will address negation of nouns and verbs, especially the cases requiring prediction of a set of alternatives rather than a true antonym (e.g. *desk, chair*, etc. for *table*). Bilinear models may also be useful for NLP tasks involving other lexical and discourse relations that would benefit from being conditioned on a domain or topic.

References

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

Kai-Wei Chang, Wen-tau Yih, and Christopher Meek. October 2013. Multi-relational latent semantic analysis. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1602–1612, Seattle, Washington. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D13-1167`.

James Gorman and James Curran. June 2005. Approximate searching for distributional similarity. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 97–104, Ann Arbor, Michigan. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W05/W05-1011`.

Gregory Grefenstette. Finding semantic similarity in raw text: the Deese antonyms. In Robert Goldman, Peter Norvig, Eugene Charniak, and Bill Gale, editors, *Working Notes of the AAAI Full Symposium on Probabilistic Approaches to Natural Language*, pages 61–65. Menlo Park, California, 1992.

David B. Grimes and Rajesh P. N. Rao. 2005. Bilinear sparse coding for invariant vision. *Neural Computation*, 17(1):47–73.

Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. August 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. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W13-3209`.

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.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne.

Anna Lobanova. *The Anatomy of Antonymy: a Corpus-driven Approach*. PhD thesis, University of Groningen, 2012.

Roland Memisevic. 2012. On multi-view feature learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland.

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*, Minneapolis, Minnesota.

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 (NIPS) 26*, pages 3111–3119, Lake Tahoe.

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

Saif Mohammad, Bonnie Dorr, and Graeme Hirst. October 2008. Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 982–991, Honolulu, Hawaii. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D08-1103`.

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 M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. June 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 142–148, San Diego, California. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1018.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. August 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany. Association for Computational Linguistics. URL http://anthology.aclweb.org/P16-2074.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. May–June 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 984–989, Denver, Colorado. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1100.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. July 2015. A multitask objective to inject lexical contrast into distributional semantics. 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)*, pages 21–26, Beijing, China. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-2004.

Jan Rudy and Graham Taylor. 2015. Generative class-conditional denoising autoencoders. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR) Workshop*, San Diego, California.

Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. December 2014. Taking antonymy mask off in vector space. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 135–144, Phuket,Thailand. Department of Linguistics, Chulalongkorn University. URL http://www.aclweb.org/anthology/Y14-1018.

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. 2013. Pattern-based distinction of paradigmatic relations for German nouns, verbs, adjectives. In *Proceedings of the 25th International Conference of the German Society for Computational Linguistics and Language Technology*, pages 184–198, Darmstadt, Germany.

Roy Schwartz, Roi Reichart, and Ari Rappoport. July 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 258–267, Beijing, China. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K15-1026.

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 (ECCV)*, Crete, Greece.

Peter Turney. August 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 905–912, Manchester, UK. Coling 2008 Organizing Committee. URL http://www.aclweb.org/anthology/C08-1114.

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*, Austin, Texas.

Wen-tau Yih, Geoffrey Zweig, and John Platt.

July 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1212–1222, Jeju Island, Korea. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D12-1111.

Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. October 2014. Word semantic representations using Bayesian probabilistic tensor factorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1522–1531, Doha, Qatar. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1161.

Will Y. Zou, Shenghuo Zhu, Andrew Y. Ng, and Kai Yu. 2012. Deep learning of invariant features via simulated fixations in video. In *Neural Information Processing Systems (NIPS 25)*, Lake Tahoe. URL http://ai.stanford.edu/~wzou/nips_ZouZhuNgYu12.pdf.