Robust Training under Linguistic Adversity

Yitong Li and **Trevor Cohn** and **Timothy Baldwin**
Department of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au, {tcohn,tbaldwin}@unimelb.edu.au

Abstract

Deep neural networks have achieved remarkable results across many language processing tasks, however they have been shown to be susceptible to overfitting and highly sensitive to noise, including adversarial attacks. In this work, we propose a linguistically-motivated approach for training robust models based on exposing the model to corrupted text examples at training time. We consider several flavours of linguistically plausible corruption, include lexical semantic and syntactic methods. Empirically, we evaluate our method with a convolutional neural model across a range of sentiment analysis datasets. Compared with a baseline and the dropout method, our method achieves better overall performance.

1 Introduction

Deep learning has achieved state-of-the-art results across a range of computer vision (Krizhevsky et al., 2012), speech recognition (Graves et al., 2013), and natural language processing tasks (Bahdanau et al., 2015; Kalchbrenner et al., 2014; Bitvai and Cohn, 2015). However, deep models tend to be overconfident in their predictions over noisy test instances, including adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). A range of methods have been proposed to train models to be more robust, such as injecting noise into the data and hidden layers (Jiang et al., 2009), dropout (Srivastava et al., 2014), and the incorporation of explicit regularization terms into the training objective (Ng, 2004; Li et al., 2016).

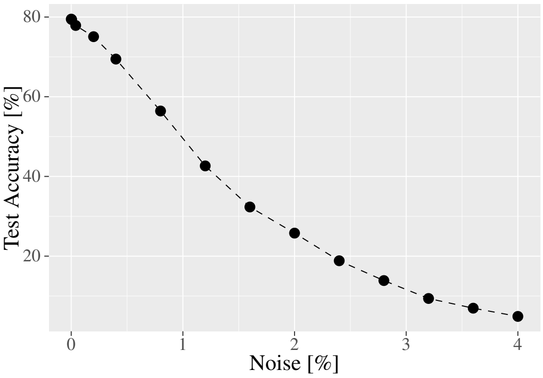

Figure 1: Accuracy (%) drops as we increase adversarial noise to word embeddings, as evaluated on binary classification dataset MR.

In this work, we propose a linguistically-motivated method customised to text applications, based on injecting different kinds of word- and sentence-level linguistic noise into the input text, inspired by adversarial examples (Goodfellow et al., 2015). Our method has its origins in computer vision, where it has been shown that small pixel perturbations indiscernible to humans can significantly distort the predictions of state-of-the-art deep models (Szegedy et al., 2014; Nguyen et al., 2015), an observation that has been harnessed in recent work on adversarial training (Goodfellow et al., 2015). This kind of noise is cheap to generate for images and is transferable between different models, but it is less clear how to generate analogous textual noise while preserving the fidelity of the training data, due to text being discrete and sequential in nature, with latent syntactic structure. Based on the same linguistic intuition, adversarial evaluation for natural language processing models was proposed by Smith (2012). Also, adversarial learning for text, such as perceptron learning (Søgaard, 2013) and unsupervised estimation methods (Smith and Eisner, 2005), have been studied in the language area.

Word embeddings learned from WORD2VEC

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 21–27,
Valencia, Spain, April 3-7, 2017.

(Mikolov et al., 2013) and GLOVE (Pennington et al., 2014) are now widely used as input to language processing models, however these representations are highly susceptible to noise. For example, Figure 1 shows that as we add adversarial noise $\eta = \epsilon \nabla_x Loss(x, y, \theta)$ to WORD2VEC representations, classification accuracy for a convolutional model (Kim, 2014) over a sentiment classification task (Pang and Lee, 2008) drops appreciably, such that with only 1% perturbations, a state-of-the-art model drops to the level of a random classifier.

Word embeddings are not an intuitive representation of human language, and it is not immediately clear how to generate adversarial noise over the raw text input without affecting the fidelity of the data. In human-to-human textual communication such as chat and microblogs, humans are remarkably resilient to "noise", in terms of typos, lexical and syntactic disfluencies, and the large variety of semantically-equivalent ways of expressing the same content (Han and Baldwin, 2011; Eisenstein, 2013; Baldwin et al., 2013; Pavlick and Callison-Burch, 2016). These observations are the inspiration for this work, in proposing a training strategy based on the explicit generation of linguistic corruption over the source training instances, to train robust text models. Empirically, we demonstrate the effectiveness of our method over a range of sentiment analysis datasets using a state-of-the-art convolutional neural network model (Kim, 2014). In this, we show that our method is superior to a baseline and dropout (Srivastava et al., 2014) using MAP training.[1]

2 Generating Text Noise

Our method involves the explicit generation of several kinds of linguistic corruption, to train more robust deep models. The first question is how to generate the linguistic noise, focusing on English for the purposes of this paper. We focus on the generation of two classes of text noise: (1) syntactic noise; and (2) semantic noise.[2]

Syntactic Noise The first class of linguistic noise is syntactic, focusing on the syntactic structure of the input, either through explicit parsing and generation using a deep linguistic parser, or sentence compression.

For the deep linguistic parser, we use the LinGO English Resource Grammar ("ERG": Copestake and Flickinger (2000)) with the ACE parser, based on pyDelphin.[3] The ERG supports both parsing and generation, via the semantic formalism of Minimal Recursion Semantics ("MRS": Copestake et al. (2005)). To generate paraphrases with the ERG, we simply parse a given input, select the preferred parse using a pretrained parse selection model (Oepen et al., 2002), and exhaustively generate from the resultant MRS. We then use uniform random sampling to select from the generator outputs, which potentially numbers in the thousands of variants. To handle unknown words during parsing and generation, we use POS mapping and introduce a unique relation for each unknown word, which we use to substitute the unknown word back in to the generation output. In practice, the primary sources of "noise" introduced by the ERG are due to topicalisation, adjective ordering, fronting of adverbial phrases, and relativisation of modifiers.

The second approach to syntactic noise is based on sentence compression ("COMP": Knight and Marcu (2000)), which aims to "trim" an input of peripheral content, while maintaining grammaticality, and also the syntax of the original as much as possible. While the state-of-the-art in sentence compression is based on deep learning methods such as recurrent neural networks (Filippova et al., 2015), we implement a simple parser-based model, due to the lack of large-scale annotated data for training and the fact that a relative lack of precision in the output may ultimately help our method. First, we parse the sentence using the Stanford CoreNLP constituency parser (Chen and Manning, 2014). Next, we model the conditional probability of deleting a sub-tree C with label S given its parent node with label R by $p(C|S, R) = \frac{p(C,S,R)}{\Sigma_C p(C,S,R)}$, trained on the sentence compression corpora of Clarke and Lapata (2006),[4] made up of a few hundred labelled instances.

Semantic Noise The second class of linguistic noise is semantic noise. Semantic noise is more subtle than syntactic noise, as we must be careful

[1] The implementation is freely available at https://github.com/lrank/Linguistic_adversity.

[2] We also experimented with a method which generates lexical noise, but for the purposes of our experiments here, as the vast majority of the generated candidates are OOV words, it is largely equivalent to word dropout, and omitted from this paper.

[3] https://github.com/delph-in/pydelphin

[4] http://jamesclarke.net/research/resources/

not to impact on the fidelity of the original labels, which can readily occur with full paraphrasing or abstractive summarisation. As such, we focus on lexical substitution of near-synonyms of words in the original text, and experiment with two methods for generating near-synonyms.

Our approach to generating semantic noise proceeds as follows. First, we apply filters to identify words which should not be candidates for lexical substitution, namely words which are parts of named entities or function words. As such, we use the Stanford CoreNLP POS tagger and named entity recogniser (Finkel et al., 2005; Chen and Manning, 2014), and identify "substitutable words" as those which are nouns, verbs, adjectives or adverbs, and not part of a named entity. For each substitutable word w, we generate the set of substitution candidates $s(w)$. For each candidate $w_i \in \{w\} \cup s(w)$ we allow the original word to be preserved with $p(w_i) = \alpha$, and share the remaining $1 - \alpha$ proportional to the language model score based on substituting w_i into the original text. For this, we use the pre-trained US English language model from the CMU Sphinx Speech Recognition toolkit.[5] Finally, we sample from the probability distribution $\{p(w_i) : w_i \in \{w\} \cup s(w)\}$ for each substitutable word w to generate a semantically-corrupted version of the original.

We experiment with two approaches to generating the substitution candidates. The first is based on Princeton WordNet ("WN": Miller et al. (1990)), over all synsets that a given substitutable word occurs in, using the NLTK API (Bird, 2006). The second is based on the "counter-fitting" method of Mrkšić et al. (2016) ("CFit"), whereby word embeddings from word2vec are projected based on a supervised objective function which penalises similarity between antonym pairs, and rewards similarity between synonym pairs, as trained on 10k English news sentences from WMT14 (Bojar et al., 2014).

Word Dropout As a standard approach to training robust models, we use word dropout (Srivastava et al., 2014; Pham et al., 2014). Dropout can be viewed as a method for zeroing out noise, and is first-order equivalent to an ℓ_2 regularizer applied after feature scaling (Wager et al., 2013).

Method	Example
Original	The cat sat on the mat .
ERG	On the mat sat the cat .
Comp	The cat sat on $\diamond$ mat $\diamond$
WN	The kat sat on the flatness .
CFit	The pet stood onto the mat .

Table 1: Examples of generated sentences across four proposed methods. Modified words are marked by "underwave" and omitted words are denoted with a "$\diamond$".

Table 1 shows an example sentence and sample corrupted outputs after applying each type of linguistic noise. The ERG seldom changes words, and instead tends to reorder the words based on syntactic alternation. Comp performs like word dropout in that it tends to remove tokens with low semantic content and to generate complete sentences. WN and CFit both only modify the text at the word level, based on near-synonyms and words with similar semantic function, respectively.

3 Models and Training

We evaluate our methods on several sentence classification tasks, using a convolutional neural network ("CNN") model (Kim, 2014). Note that our method corrupts the input directly, and is thus easily transferrable to other classes of models (e.g., other deep learning or linear models).

Convolutional Neural Network The CNN operates at the sentence level by first embedding each word using a lookup table which is stacked into the sentence matrix $\mathbf{E}_{\text{S}}$. A 1d convolutional layer is then applied to $\mathbf{E}_{\text{S}}$, which applies a series of filters over each window of t words, with each filter employing a rectifier transform function. $\mathrm{MaxPooling}$ is applied over each set of filter outputs to result in a fixed-size sentence representation.[6] The sentence vector is fed into a final $\mathrm{Softmax}$ layer to generate a probability distribution over classification labels.

The model is trained to minimise the cross-entropy between the ground-truth and the model prediction, using the Adam Optimizer (Kingma and Ba, 2015) with learning rate 10^{-4} and a

[5] https://sourceforge.net/projects/cmusphinx/

[6] We use window widths of size $t \in \{3, 4, 5\}$, and 128 filters for each size. $\mathrm{MaxPooling}$ is applied to each of the three sizes separately, and the resulting vectors are concatenated to form the sentence representation.

batch size of 128. We initialise the embedding with dimension $m = 300$ Google pre-trained WORD2VEC word embeddings (Mikolov et al., 2013). Words not in the pre-trained vocabulary are initialized randomly using a uniform distribution $U([-0.25, 0.25)^m)$.

Injecting Noise during Training Our proposed method involves corrupting the training input with adversarial noise of various kinds. All the methods are non-deterministic, involving random sampling. They are applied afresh every epoch, such that each time an instance is processed, it will have a different input form.[7] The two semantic approaches (WN and CFIT) support configurable noise rates in terms of the proportion of substitutable words that are corrupted. Accordingly, we experiment with two thresholds on the random variable for substitution of each word: low ("lo"; $\alpha = 0.5$) and high ("hi"; $\alpha = 0$). Besides the above methods which employ a single type noise, we experiment with a combination (COMB) of the four different noise types (ERG + COMP + WN$_{\text{lo}}$ + CFIT$_{\text{lo}}$), by uniformly randomly choosing one of the four methods for noise generation each time we process a training instance.

Datasets We experiment on the following datasets:

- MR: sentence polarity dataset from movie reviews (Pang and Lee, 2008)[8]
- CR: customer review dataset (Hu and Liu, 2004)[9]
- Subj: subjectivity dataset (Pang and Lee, 2005)[8]
- SST: Stanford Sentiment Treebank, using the 2-class configuration (Socher et al., 2013)[10]

We evaluate using classification accuracy, based on both in-domain evaluation[11] and a cross-domain setting, in which we evaluate a model trained on MR and tested on CR, and vice versa. This last setting characterises a realistic application scenario, where robustness to vocabulary shift and other differences in the input is paramount.

4 Experimental Results and Analysis

Table 2 presents the results of training with different sources of linguistic corruption in the in-domain and cross-domain settings. In general, the proposed methods perform better than the baseline and dropout, and semantic noise using WN achieves consistent improvements across all settings. The COMB method uniformly outperforms the other methods for all in-domain evaluations, indicating that the improvements from training with different types of noise are orthogonal. Note that improvements are smaller on SST and MR than CR and Subj for all methods. Almost every method improves over word dropout, except counter-fitting at a high noise level. Also surprising is the fact that dropout shows no improvement over standard training, and is overall mildly detrimental.

Our intuition behind why WN consistently outperforms the baseline methods and other single sources of noise is it sometimes performs similarity to dropout, in replacing common words with rare ones, and sometimes substitutes frequent words for frequent words, leading to better generalisation in the word embeddings. To test this hypothesis, we computed nearest neighbours in the word embedding space for both the baseline method and the WN method. For example, the top-3 nearest neighbours for *superior* in CR are *exceptional*, *excellent* and *unmatched* for WN, while for the baseline, they are *inferior*, *exceptional* and *excellent*. That is, similar to the intuition behind counter-fitting, the methods appears to learn to differentiate between synonyms and antonyms, in a manner which is sensitised to the target domain.

Although similar in function to WN, the counter-fitting based method performs unexpectedly poorly. This appears to be a consequence of the training of these embeddings, namely that the corpus was much smaller than that used for the WORD2VEC training, and consequently coverage on our corpora was substantially lower, leading to the approach making inappropriate substitutions and not aiding model robustness.

Sentence compression was found to be highly effective. To illustrate by example, the sentence *Player has a problem with dual-layer dvd's such*

[7] Using a single application of noise is less effective, but still yields improvements over baseline methods including dropout.

[8] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[9] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[10] http://nlp.stanford.edu/sentiment/

[11] Where there is no pre-defined training/test split for a given dataset, we use 10-fold cross validation. See Kim (2014) for more details on the datasets and evaluation settings.

Method	In domain				Cross domain	
	MR	CR	Subj	SST	MR/CR	CR/MR
baseline	80.4	82.6	92.4	84.5	67.0	67.2
dropout	80.1	82.4	92.6	84.5	67.7	67.4
ERG	80.0	82.8	92.9	84.4	68.1	67.3
COMP	79.5	83.1	93.2	84.3	68.1	**67.5**
WN_{lo}	80.9	83.2	93.1	84.3	68.5	67.3
WN_{hi}	81.2	83.8	92.9	84.6	67.9	**67.5**
CFIT_{lo}	79.8	82.7	92.6	84.1	**68.9**	67.3
CFIT_{hi}	76.2	78.9	91.0	80.3	67.4	64.2
COMB	**81.4**	**84.3**	**93.6**	**84.8**	68.4	67.4

Table 2: Accuracy (%) of the CNN, in four in-domain settings, and two cross-domain settings, with word dropout ("dropout"), or linguistic corruption based on different sources of syntactic and semantic corruption. The best result for each dataset is indicated in **bold**.

as Alias seasons 1 and season 2 is compressed into *has a problem with dual-layer dvd* which preserves the key information that we expect to be useful for model learning. This allows the model to better learn the components of the input that are predictive of sentiment.

Syntactic paraphrasing (ERG) tends to primarily corrupt the word order, with fewer lexical substitutions. Thus, the model is less prone to overfitting to local n-gram features, and focuses on learning words and phrases that are genuinely predictive of sentiment.

5 Conclusions

In this paper, we present a training method that corrupts training examples with linguistic noise, in order to learn more robust models. Based on evaluation over several sentiment analysis datasets with convolutional neural networks, we show that this method outperforms standard training and dropout, both for in-domain and out-of-domain application. Our approach has wide-spread potential to also benefit other types of discriminative model and in a range of other language processing tasks.

Acknowledgments

We are grateful to the anonymous reviewers for their helpful feedback and suggestions, and to Ned Letcher for assistance in running the ERG. This research was supported in part by the Australian Research Council.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364, Nagoya, Japan.

Steven Bird. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. 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 180–185, Beijing, China.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, USA.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750, Doha, Qatar.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 377–384, Sydney, Australia.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage english grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, USA.

Katja Filippova, Enrique Alfonseca, A. Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal.

Rose Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, Ann Arbor, USA.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, Vancouver, Canada.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, USA.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, USA.

Yulei Jiang, Richard M. Zur, Lorenzo L. Pesce, and Karen Drukker. 2009. A study of the effect of noise injection on the training of artificial neural networks. In *International Joint Conference on Neural Networks*, pages 1428–1432, Atlanta, USA.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, USA.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, San Diego, USA.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. In *Proceedings of the 18th Annual Conference on Artificial Intelligence*, pages 703–710, Austin, USA.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, Lake Tahoe, USA.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2016. Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1979–1985, Austin, USA.

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, Lake Tahoe, USA.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, USA.

Andrew Y. Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Canada.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, Boston, USA.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1253–1257, Taipei, Taiwan.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, USA.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.

Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *14th International Conference on Frontiers in Handwriting Recognition*, pages 285–290, Crete, Greece.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–362, Ann Arbor, USA.

Noah A. Smith. 2012. Adversarial evaluation for models of natural language. *arXiv preprint arXiv:1207.0245*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, USA.

Anders Søgaard. 2013. Part-of-speech tagging with antagonistic adversaries. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 640–644, Sofia, Bulgaria.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, Banff, Canada.

Stefan Wager, Sida Wang, and Percy S. Liang. 2013. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359, Lake Tahoe, USA.