A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification

Shiou Tian Hsu, Changsung Moon, Paul Jones and **Nagiza F. Samatova***
North Carolina State University, Raleigh, NC, USA
{shsu3, cmoon2, pjones}@ncsu.edu, samatova@csc.ncsu.edu

Abstract

The success of sentence classification often depends on understanding both the syntactic and semantic properties of word-phrases. Recent progress on this task has been based on exploiting the grammatical structure of sentences but often this structure is difficult to parse and noisy. In this paper, we propose a structure-independent 'Gated Representation Alignment' (GRA) model that blends a phrase-focused Convolutional Neural Network (CNN) approach with sequence-oriented Recurrent Neural Network (RNN). Our novel alignment mechanism allows the RNN to selectively include phrase information in a word-by-word sentence representation, and to do this without awareness of the syntactic structure. An empirical evaluation of GRA shows higher prediction accuracy (up to 4.6%) of fine-grained sentiment ratings, when compared to other structure-independent baselines. We also show comparable results to several structure-dependent methods. Finally, we analyzed the effect of our alignment mechanism and found that this is critical to the effectiveness of the CNN-RNN hybrid.

1 Introduction

Sentence classification is the task of modeling, representing and assigning sentences to classes, which are often based on structure or sentiment. This task is important for many applications requiring a degree of semantic comprehension. Recent advancements in sentence classification employ *distributed embedding models* (Mikolov et al., 2013), which discover semantic relations between words and represent words as real-valued vectors. State-of-the-art classification methods typically combine distributed embedding models with the following three strategies: n-gram models, sequential models and tree models. Of these, the best results have been obtained using tree models (Mou et al., 2015; Tai et al., 2015), which use sentence syntactic trees originating from grammar to help construct sentence embeddings. However, noisy text (such as found in online reviews) does not always contain much grammatical structure, which reduces the effectiveness of tree models. Hence it is important to study structure-independent models.

Much recent research into structure-independent n-gram CNN models (Kalchbrenner et al., 2014; Yu et al., 2014; Yin and Schütze, 2015; Kim, 2014; Zhang et al., 2016) attempts to build comprehensive sentence embeddings by identifying the most influential n-grams of different semantic aspects. However, while these methods are effective at exploring the regional syntax of words, they are unable to account for order-sensitive situations, where the order of words is critical to the meaning.

On the other hand, sequential models based on RNN (Graves, 2013; Sutskever et al., 2014; Palangi et al., 2016) build sentence embeddings using a *global cell* that reads one word at a time. The cell contains an update function that uses the most recent word to update sentence embeddings, while maintaining some memory of previously seen words. Recent extensions of RNN cells, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) (Cho et al., 2014), better enable the cell to memorize and forget information that is pertinent to the meaning of the sentence. However, it is not clear how much phrase-level information is captured since the RNN cells are optimized from a whole-sentence perspective.

* Corresponding author

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 443–449,
Valencia, Spain, April 3-7, 2017.

In this paper, we propose a hybrid CNN-RNN framework to model relationships between phrases and word sequences in each sentence. In the framework, we added a soft-aligning layer that provides an adaptive mechanism for RNN to 'peek' into relevant n-grams generated by a CNN and selectively include them. We call our model *Gated Representation Alignment* (GRA) since we implement soft-alignment using a group of Gated Recurrent Units. Similar to CNN and RNN approaches, GRA requires no explicit structural information about the sentence, making it adaptable to noisy text.

In our experiments, GRA outperforms an LSTM baseline by 4.6% when classifying fine-grained sentiment datasets. The other eight baseline models we tested improve on this baseline by up to 3.2%. Furthermore, GRA achieves comparable results to structure-dependent models. Further analysis against baselines shows the alignment mechanism in GRA is the key to combine the power of CNN and RNN approaches.

2 Methodology

Figure 1 depicts the GRA model, which consists of three stages: the first generates phrase vectors using CNN; the second combines the word and phrase vectors, and incorporates word order to generate sentence representations through a soft-aligned RNN; the third stage makes class predictions based on these sentence representations. The figure shows the processing flow for the i-th word, which is equivalent to the i-th time step.

2.1 Phrase Vector CNN

In the first stage of the GRA model, phrase vectors are derived from a set of CNNs that operate on the input sequence of words. Each phrase vector is a representation of between two and five words.

Let $X_i \in \mathbb{R}^k$ represent a k-dimensional embedding for the i-th word in the sentence. An input sentence of length N can thus be considered as a vertical concatenation of $X_{1:N}$. We apply a set of convolutional filters W_P^ℓ and bias terms b_P^ℓ to the sentence as per equation (1), in order to learn a representation for each phrase of length ℓ.

$$P_i^\ell = Relu(W_P^\ell \cdot [X_i, .., X_{i-\ell}] + b_P^\ell) \qquad (1)$$

We use $P_i^{L=\{2,3,..,\ell\}}$ to represent phrase vectors at time i, which includes all phrases ended with X_i.

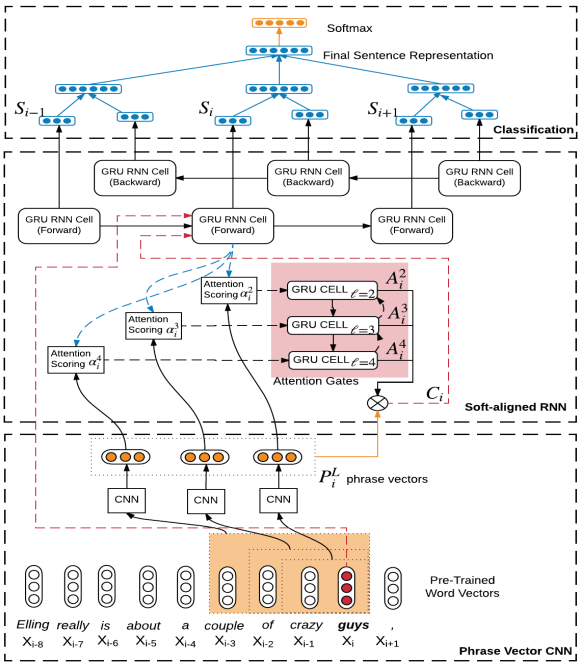

Figure 1: GRA Framework and Details at Step i

2.2 Soft-aligned RNN

The second stage generates sentence vector representations (or states) using a soft-aligned RNN. The state updated with the i-th word is represented as a d-dimensional vector S_i.

Our model was inspired by an attention GRU-RNN model introduced by Bahdanau et al. (2015), which was originally used for machine translation. The attention model provides an interface for a neural network to selectively include outputs from another model, which is ideal for our purpose of combining CNN and RNN.

For the i-th time step in GRU-RNN, the GRU cell forgets a portion of learned sentence information S_{i-1} using the update gate Z, and updates it through a reset gate R. In GRU cells, both gates are controlled by S_{i-1} and X_i. In GRA, another vector C_i combines the weight from the *Attention Gates* in Figure 1 with each phrase vector from CNN. This provides input to the GRU RNN cells, as shown in equation set (2).

An intuitive way to understand C_i is to consider that the model tries to determine which of the phrases generated by word X_i are more reasonable based on current sentence state S_i. In the example sentence shown in Figure 1, for the word *'guys'*, the weighting function determines weights for each of the phrase vectors representing *'couple of crazy guys'*, *'of crazy guys'* and *'crazy guys'*

based on their similarity to the sentence state.

To compute similarity, both the phrase vectors P_i^ℓ and the sentence state S_i^* are projected to a new vector space (after S_{i-1} is updated with X_i), and then similarity is evaluated by a dot product, represented as α_i^ℓ. We call this step **attention scoring** and formalize in equation set (3).

In the Bahdanau et al. (2015) attention framework, the underlying assumption was that one neural network always received the output of another. Applying softmax to the attention scores indicated that the receiving neural network must focus on a certain part of the input. However, this assumption might not hold in the GRA framework as phrase information is not always needed at each timestep of RNN training. For the example sentence *"Then one day, completely out of the blue, I had a letter from her."*, we clearly need to include phrase vectors for the word "blue" (which is only meaningful as part of a phrase) but not for other words such as "I". Accordingly, a loosely coupled framework that dynamically incorporates or omits phrase vectors is necessary.

The major challenge here is that the algorithm needs a reference to compute weights for the phrase vectors. For instance, in softmax, each input is simply weighted by its contribution to the sum. However, in GRA, the sum of similarity scores is not a good scaling factor since phrase vectors are sometimes omitted. Instead, we use a set of GRU cells that receive previous weights, other phrase's weights, and attention scores as inputs, and use these to compute the final weights for each phrase vector. The intuition is that GRA is trying to determine the weight for P^ℓ by concatenating attention scores, past weights and weights assigned to other phrase vectors. Using a RNN cell helps to store relevant past information and allows concurrent weights be easily added into the formula. To compute the weight for P_i^ℓ, a GRU cell receives the weight for $P_{i-1}^{\ell-1}$ if the weight for $P_i^{\ell-1}$ is not computed yet. We called this process **attention gating**, and the final output is the set of weights A_i^ℓ for the phrase vector P_i^ℓ, as formalized in equation set (4).

2.3 Classification Layer and Regularization

The penultimate layer of GRA, which outputs the final sentence vectors, averages sentence states from all time steps. Finally, classification is done using softmax to project the final sentence vector to K conditional probabilities, where K is the number of classes, and a class prediction is obtained from the **argmax** operation.

We implemented a bi-directional RNN with dropout for regularization (Pham et al., 2014). The RNN cells are shared for both forward and backward passes to limit the number of variables. This also helps to decrease over-fitting.

GRU RNN Cell[1,2,3]:

$$
\begin{aligned}
Z_i &= sigmoid(W_Z \cdot [X_i, S_{i-1}, C_i] + b_Z) \\
R_i &= sigmoid(W_R \cdot [X_i, S_{i-1}, C_i] + b_R) \\
H_i &= tanh(W_H \cdot [X_i, R_i \odot S_{i-1}, C_i] + b_H) \\
S_i &= (1 - Z_i) \odot S_{i-1} + Z_i \odot H_i
\end{aligned} \tag{2}
$$

Attention Scoring:

$$
\begin{aligned}
\alpha_i^l &= U_\alpha \cdot tanh((W_\alpha \cdot P_i^l) \odot S_i^*) + b_\alpha \\
S_i^* &= W_s \cdot [S_{i-1}, X_i] \\
\alpha_i^L &= [\alpha_i^2, ..., \alpha_i^\ell]
\end{aligned} \tag{3}
$$

Attention Gate[4,5]:

$$
\begin{aligned}
AZ_i^\ell &= tanh(W_{AZ}^\ell \cdot [\alpha_i^L, A_{latest}^{L-\ell}, A_{i-1}^\ell] + b_{AZ}) \\
AR_i^\ell &= tanh(W_{AR}^\ell \cdot [\alpha_i^L, A_{latest}^{L-\ell}, A_{i-1}^\ell] + b_{AR}) \\
AH_i^\ell &= tanh(\\
&\quad W_{AH}^\ell \cdot [\alpha_i^L, A_{latest}^{L-\ell}, AR_i^\ell \odot A_{i-1}^\ell] + b_{AH}) \\
A_i^\ell &= (1 - AZ_i^\ell) \odot A_{i-1}^\ell + AZ_i^\ell \odot AH_i^\ell \\
C_i &= [Ai^2 \odot Pi^2, ..., Ai^\ell \odot Pi^\ell]
\end{aligned} \tag{4}
$$

3 Datasets and Experimental Setup

We tested our model on datasets containing both 'clean' (i.e. well-structured) and 'noisy' text.

The clean datasets are obtained from **Stanford Sentiment Treebank (SST5)**, a 5-class movie review corpus (i.e. very negative, negative, neutral, positive, very positive) from Socher et al (2013). Labeling is done at both sentence and phrase level. Well-known sub-phrases (and individual words) are labelled separately for training, but are not used in testing. Dataset **SST2** is the same as SST5 but reduced to binary classes.

The noisy dataset is a 5-classes review dataset from **Yelp** (Tang et al., 2015). We parsed short reviews (less than 60 words) from the 200 most frequently reviewed restaurants. Also, we undersampled positive and very positive reviews as the reviews are skewed toward the positive end.

[1] $\odot$ represents element-wise multiplication

[2] [A,B]represents horizontal concatenation of A and B

[3] W represents weight matrix used for the corresponding parameter, and b as bias terms

[4] $A_i^{L-\ell}$ represents ℓ is excluded from L

[5] latest refers to i or i-1, depending if $A_i^{L-\ell}$ is computed

The accuracy results from the clean datasets were averaged over 5 runs using the train/test splits given in the datasets. The noisy dataset wasn't broken down in this way in advance, so we evaluated it using 10-fold cross validation.

In order to minimize parameter tuning, we used the *Adadelta* (Zeiler, 2012) optimizer to obviate the need to determine a learning rate. Dropout is set to 50% for each timestep in RNN, and we use no dropout in the penultimate layer.

During experiments, we set the dimension of word vectors to 300, and the CNN filter length to [2,3,4]. Each CNN filter has 150/50 dimensions in SST5,SST2/Yelp. Bi-directional RNN state size is set to 450/150 for SST5, SST2/Yelp for each direction. Each experiment lasts 10 epochs, with mini-batch size of 200. Similar to most benchmark models, GRA uses pre-trained word vectors[6] (trained on **GoogleNews**) to initialize the words embeddings. Words not present in the corpus are initialized randomly.

4 Results and Discussion

The classification accuracy of GRA and baseline methods are shown in Table 1. Results for baseline methods running against the SST5 / SST2 datasets are mostly taken directly from the corresponding papers[7] [8]. For baseline algorithms we reimplemented, we used the parameter settings specified in the original papers. It was only possible to run some of the baseline algorithms on the Yelp dataset due to availability of source code and parameter configurations.

It can be seen from Table 1 that GRA outperforms the baselines on the fine-grained datasets (SST5 / Yelp), and is also comparable with the binary case (SST2).

Next, we further investigated the effect of soft-alignment, and compared GRA with structure dependent models for a more extensive analysis.

4.1 Effect of Soft-alignment

We first empirically evaluate the effect of soft-alignment by comparing GRA with/without soft-alignment on the **SST5** dataset. In the latter case, the last formula in formula set (4) becomes $C_i = [P_i^2, ..., P_i^\ell]$, which can be seen as simply chaining together the two models. We added two more CNN and RNN hybrid models here for comprehensive comparison. Both hybrids combined CNN and RNN at the penultimate layer, but the first one combined models by taking the average of the softmax scores; the second combined models by concatenating the sentence vectors generated by CNN and RNN. These two hybrid models can be seen as ensemble approaches since CNN and RNN are not interacting while generating the sentence vector. We show the results in Table 2.

Methods	SST5	SST2	Yelp
LSTM (baseline)	46.4	85.9†	56.5
Bi-Directional LSTM	49.5	86.1†	57.8
DCNN (Kalchbrenner et al., 2014)	48.5	86.8	-
Paragraph-Vec (Le and Mikolov, 2014)	48.7	87.8	-
CNN non-static (Kim, 2014)*	48.0	87.2	55.5
CNN multi-channel (Kim, 2014)*	47.4	88.1	56.0
MG-CNN(w2v+Glv) (Zhang, 2016)*	48.2	87.9	55.8
MGNC-CNN(w2v+Syn+Glv) (Zhang, 2016)	48.6	88.3	-
MVCNN (Yin and Schutze, 2016)	49.6	**89.4**	-
GRA	**51.0**	87.9 †	**58.1**

Table 1: Accuracy of GRA and benchmarks. † denotes models that are trained on SST5 but sum the result of the softmax layer to obtain binary predictions; as stated in Mou et al. (2015), it is more difficult to obtain good results with this approach.

Methods	SST5
Average of softmax of CNN and RNN	50.2
Concatenate CNN and RNN	50.6
GRA not-aligned	48.8
GRA	**51.0**

Table 2: Accuracy of GRA and other hybrids.

It can be seen from Table 2 that even very simple ensemble methods can yield good results when compared to standalone models. On the other hand, for GRA without alignment the result became worse when compared to RNN without phrase vectors (i.e. Bi-Directional LSTM in Table 1). We suppose that the drop of accuracy in the not-aligned version is a result of phrase vectors being over-counted with large weights, and thus reducing the effectiveness of the sequence learning ability in RNN. However, with soft-alignment, GRA can incorporate CNN phrase vectors into an RNN without impacting the sequence learning effectiveness.

We further qualitatively tested our assumption

[6] https://code.google.com/p/word2vec

[7] * denotes that we reimplemented the algorithm, but reported SST5/SST2 results based on the results shown in their publications.

[8] Models without citation are implemented following parameter settings in section 3.

that GRA preserves more phrase level information without compromising the RNN. We evaluate this by quantifying the union of correct cases from GRA (both with and without soft-alignment) against the CNN/LSTM baselines. If soft-alignment helps to bridge the two models, then the predictions from GRA should be closer to those from CNN/LSTM with soft-alignment enabled than the not-aligned case. We show the results of this evaluation in Figure 2 using the test set from SST5. Each point shows the size of the union of correct cases for a variety of sentence lengths, and only for sentences that are predicted correctly more than 3 times in the 5 runs. When compared to LSTM and CNN/LSTM models, GRA with alignment produces a consistently larger union of correct cases (typically by 5-10%) than GRA without alignment. These results support our intuition that soft-alignment make an important difference.

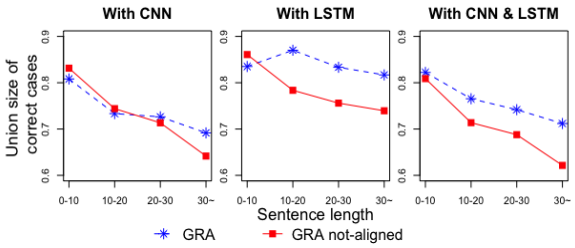

Figure 2: Coverage of CNN and LSTM correct cases between GRA and GRA-without-alignment.

We also evaluated how sentimentally-sensitive the model is with soft-alignment by slightly modifying some of the sentences. We demonstrate in Figure 3 how predicted sentiments can be changed using a sample sentence. In Figure 3, we change the sentiment of sentence with minimal interruption, i.e. *"good"* to *"not good"* or *"bad"*. While all models reacted to the change significantly, GRA predicts a major sentiment shift and is the only one that changes the overall output prediction to negative. We believe the abrupt change in sentiment observed by GRA is caused by the model capturing phrase level changes.

4.2 Structure-dependent Models

In Table 3, we compare GRA with state-of-the-art structure-dependent models. Although we were only able to run one baseline against the noisy Yelp dataset (due to both availability of re-implementation and the lack of a good sentence-grammar tree), GRA shows comparable results to these models, and does no worse than second place for SST5 and SST2.

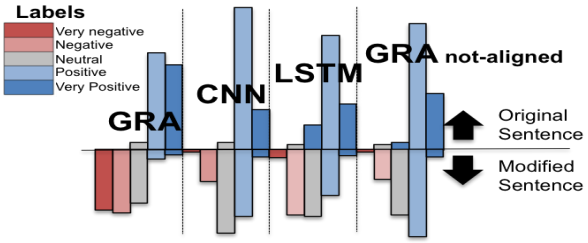

Figure 3: Change of sentiment distribution when sentiment of sentence is manually reversed. Sentiment distribution is obtained by feeding the derived sentence vectors to the softmax layer. The sample sentence was a **positive** sentence: *"If you sometimes like to go to the movies to have fun, this movie is a **good** place to start"*. We replaced *"a good"* with *"not a good"* to reverse the sentiment of the sentence.

Methods	SST5	SST2	Yelp
MV-RNN (Socher et al., 2012)	44.4	82.9	-
RNTN (Socher et al., 2013)	45.7	85.4	-
DRNN (Irsoy and Cardie., 2014)	49.8	86.6 †	-
Dependency Tree-LSTM (Tai et al., 2015)	48.4	85.7	55.2
Constituency Tree-LSTM (Tai et al., 2015)	51.0	**88.0**	-
c-TBNN (Mou et al., 2015)	50.4	86.8 †	-
d-TBNN (Mou et al., 2015)	**51.4**	87.9 †	-
GRA	51.0	87.9 †	**58.1**

Table 3: Accuracy of GRA against structure dependent methods.† has same meaning as Table 1.

5 Conclusion

We propose a novel structure-free method for combining RNN with CNN to improve sentence modeling. While CNN captures phrase-level information by convoluting sub-sentences, RNN preserves global sentence information. Our soft-alignment mechanism helps to combine the two. Empirical results show that our hybrid model outperforms the baseline structure-free models, and performs similarly to structure-dependent models.

Acknowledgments

This material is based upon work supported in whole or in part with funding from LAS. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*, San Diego, California.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104, Montreal, Canada.

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, Maryland, June. Association for Computational Linguistics.

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

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, Beijing, China.

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*, pages 3111–3119, Lake Tahoe, Nevada.

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2315–2325, Lisbon, Portugal, September. Association for Computational Linguistics.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):694–707.

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

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. 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, Washington, October. Association for Computational Linguistics.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Proceedings of INTERSPEECH*, pages 194–197, Portland, Oregon.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, Montreal, Canada.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal, September. Association for Computational Linguistics.

Wenpeng Yin and Hinrich Schütze. 2015. Multichannel variable-size convolution for sentence classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 204–214, Beijing, China, July. Association for Computational Linguistics.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer

sentence selection. In *NIPS Deep Learning Workshop*, Montreal, Canada.

Matthew D. Zeiler. 2012. Adadelta: an adaptive learning rate method. *CoRR, abs/1212.5701*.

Ye Zhang, Stephen Roller, and Byron C. Wallace. 2016. Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1522–1527, San Diego, California, June. Association for Computational Linguistics.