INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Attention-based Convolutional Neural Networks for Sentence Classification

Zhiwei Zhao, Youzheng Wu

SONY China Research Lab

`zhiweia.zhao@sony.com.cn, youzheng.wu@sony.com.cn`

Abstract

Sentence classification is one of the foundational tasks in spoken language understanding (SLU) and natural language processing (NLP). In this paper we propose a novel convolutional neural network (CNN) with attention mechanism to improve the performance of sentence classification. In traditional C-NN, it is not easy to encode long term contextual information and correlation between non-consecutive words effectively. In contrast, our attention-based CNN is able to capture these kinds of information for each word without any external features. We conducted experiments on various public and in-house datasets. The experimental results demonstrate that our proposed model significantly outperforms the traditional CNN model and achieves competitive performance with the ones that exploit rich syntactic features.

Index Terms: attention, convolutional neural network, SLU

1. Introduction

Spoken language understanding (SLU) typically involves identifying user's intent, which is to predict one label for each natural language sentence [1]. In natural language processing (NLP), sentence classification is also an essential component in many applications, such as question classification [2] and sentiment analysis [3]. In sentence classification, n-gram is one of the most used features. However n-gram cannot capture long term contextual information between non-consecutive words. Some studies introduced syntactic parsing information [4, 5] and showed that this kind of information plays an important role in modeling sentence.

Recently, neural network methods have achieved new state-of-the-art performance in a wide range of SLU and NLP tasks. In sentence classification, prior studies have presented recursive neural networks (Recursive NN) [6], recurrent neural networks (Recurrent NN) [7] and convolutional neural networks (CNN) [8]. To leverage syntactic parsing information, Recursive N-N [6] accepts a syntactic parse tree and generates the sentence representation along the branches in a bottom-up way, which has been proven to be efficient on capturing the semantics of sentence but its performance heavily relies on the performance of the syntactic tree construction. On the other hand, Recurrent NN [7] models a sentence word by word from left to right and stores the information of all previous contexts in a fixed-sized hidden layer, which could capture semantics of long sentence but has bias on later words. Therefore, some studies [9, 10] tried to combine the benefits of Recursive and Recurrent NNs such as feeding syntactic parse tree into LSTM which is an improved variant of Recurrent NN. CNN was originally proposed in computer vision, and recently it becomes popular in NLP tasks, such as sequence labeling [11, 12], machine translation [13], and sentence modeling [8, 14]. Different from Recursive and Recurrent NNs, CNN encodes n-grams by convolution operation and generates a fixed-sized high-level representation by pooling. Experimental results demonstrate that CNN is very efficient on capturing sentence information, which achieves new state-of-the-art performances on various classification tasks and doesn't have the bias problem.

However, there's a major limitation on basic CNN: it only considers sequential n-grams that are consecutive on the surface string, and thus overlooks some long distance correlation between non-consecutive words, while this kind of correlation plays an important role in many linguistic phenomena such as negation, subordination, and *wh*-extraction, all of which might dully affect the sentiment, subjectivity, or other categorization of the sentence. To address this problem, Ma et al. [15] proposed a dependency-based CNN which replaces the sequential contexts of word to word's parent, grandparent, great-grand-parent, and siblings on the dependency tree; Mou et al. [16] proposed tree-based CNN on either constituent trees or dependency trees of sentences to directly extract sentence structural features of sentence. Unfortunately, both of these two models require syntactic parsing which needs additional knowledge.

In parallel, a new direction of neural network research has emerged that make models learn to put different "attention" on different parts of input, which has been applied in machine translation [17], caption generation [18], handwriting synthesis [19], visual object classification [20], speech recognition [21, 22] and question answering [23]. In general, most of the existing studies on attention in NLP focus on modeling the correlations between different modalities, such as word alignment between source language and target language in machine translation and word similarity between question and answer in question answering. To our knowledge, there's few research on modeling the correlations between words within a sentence.

In this paper, we propose a novel attention based CNN (abbreviated to **ATT-CNN**), in which attention mechanism is used to capture long term contextual information and correlation between non-consecutive words automatically without any external syntactic information. We conduct experiments on several datasets and show that the proposed ATT-CNN model outperforms the basic CNN significantly and achieves competitive performance with the state-of-the-art models that exploit rich syntactic features.

2. Attention-based CNN

2.1. Basic CNN

CNN was adapted to various NLP tasks by Collobert et al. [11], and was extended to sentence classification task by Kalchbrenner et al. [8], Kim [14] and Ma et al. [15]. Take Kim's CNN [14] as an example. The model first replaces each word in a sentence with its vector representation and create sentence matrix $\mathbf{A} \in \mathbb{R}^{l \times d}$ where l is the (zero-padded) sentence length, and d

http://dx.doi.org/10.21437/Interspeech.2016-354

is the dimension of word embedding,

$$\mathbf{A} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_l]^\top, \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional word vector corresponding to the i-th word in the sentence. Secondly, the convolution operates on a sliding window of width n (n words). Concretely, the operation takes the dot product between a filter $\mathbf{W} \in \mathbb{R}^{nd \times h}$ and each n-gram in the sentence to obtain another sequence $\mathbf{c}$ which is composed of a series of h-dimensional vector $\mathbf{c}_i$ computed by:

$$\mathbf{c}_i = f\left(\mathbf{W} \cdot \mathbf{x}_{i:n} + \mathbf{b}\right), \tag{2}$$

where f is a non-linear activation function such as rectified linear unit (ReLU) or sigmoid function, and $\mathbf{x}_{i:n}$ denotes the concatenation of n word vectors: $\mathbf{x}_{i:n} = \mathbf{x}_i \oplus \mathbf{x}_{i+1} \oplus \cdots \oplus \mathbf{x}_{i+n-1}$. Finally, max pooling operation is applied on $\mathbf{c}$, which is followed by a fully connected softmax layer whose output is the probability distribution over labels.

Therefore, basic CNN cannot capture long term contextual information and correlation between non-consecutive words. This weakness is partially overcome in dependency-based CNN, but it always requires extra resources to get good dependency tree in practical use such as spoken language processing. In this paper, we propose a novel attention-based CNN to alleviate this problem.

2.2. ATT-CNN

Fig. 1 demonstrates the architecture of our proposed ATT-CNN model. As Fig. 1 shows, an attention layer is introduced between input layer and convolution layer. Concretely, the attention layer is to create a context vector for each word. The context vector is concatenated with the word vector as a new word representation which will be fed to the convolution layer. Intuitively, a pair of words that are far away from each other tends to be less connected. Therefore, we add distance decay to the attention mechanism.

The idea of the attention mechanism is to learn to focus the attention on specific significant words when deriving context vector $\mathbf{g}_i$ of $\mathbf{x}_i$. The red rectangles in Fig. 1 represents $\mathbf{g}_i$. The attention mechanism is an additional MLP which is jointly trained with all the other components of ATT-CNN. This mechanism determines which words should be put more attention on than other words over the sentence when predicting sentence class. The scored words are combined in a weighted sum:

$$\mathbf{g}_i = \sum_{j \neq i} \alpha_{i,j} \cdot \mathbf{x}_j, \tag{3}$$

where $\alpha_{i,j}$ are called attention weights and we require that $\alpha_{i,j} \geqslant 0$ and that $\sum_j \alpha_{i,j} = 1$ through softmax normalization. The equations describing the attention mechanism are [17, 24]:

$$\alpha_{i,j} = \frac{\exp\left(\mathrm{score}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)}{\sum_{j'} \exp\left(\mathrm{score}\left(\mathbf{x}_i, \mathbf{x}_{j'}\right)\right)}, \tag{4}$$

$$\mathrm{score}\left(\mathbf{x}_i, \mathbf{x}_j\right) = v_a^\top \tanh\left(W_a[\mathbf{x}_i \oplus \mathbf{x}_j]\right). \tag{5}$$

where score value is computed by the MLP mentioned above. Simply put, we use this MLP to model the correlation of words pair $(\mathbf{x}_i, \mathbf{x}_{j,j \neq i})$. And those $\mathbf{x}_{j,j \neq i}$ that have large score have more weights in context vector $\mathbf{g}_i$.

Take the following sentence in sentiment classification as an example:

There's not one decent performance from the cast and not one clever line of dialogue.

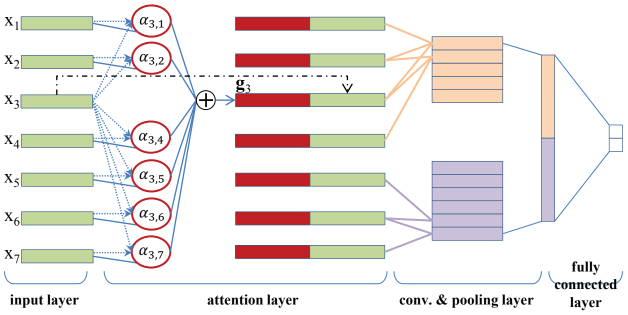


Figure 1: *ATT-CNN architecture. In attention layer, the dotted lines correspond to Eq. (4)-(5), the solid lines to Eq. (3), the dotted dash line is copy operation.*

When we learn the context vector for the word *performance* in this sentence, the attention mechanism focuses more attention on the words *not* and *decent* than other words. Particularly, score($performance$, not) and score($performance$, $decent$) is expected to be greater than other word pairs.

Furthermore, we introduce a decay factor $\lambda \in [0, 1)$ as a penalty to the output of score function to reduce the impact of noise information which would be produced as the sentence length grows.

$$\mathrm{score}\left(\mathbf{x}_i, \mathbf{x}_j\right) = (1-\lambda)^k \cdot \mathrm{score}\left(\mathbf{x}_i, \mathbf{x}_j\right), \tag{6}$$

where $k = |j - i| - 1$. As we set $\lambda \to 0$, contexts in a wider range would be taken into consideration; while if we increase λ to make it close to 1, only the contexts in a local range would be considered.

Finally, we define the extended word vector $\mathbf{x}'_i$ for $\mathbf{x}_i$ as the concatenation of $\mathbf{x}_i$ and the its context vector, and $\mathbf{x}'_i$ is then used to update the sentence matrix A, which will be fed into the CNN as described in Section 2.1.

$$\mathbf{x}'_i = \mathbf{x}_i \oplus \mathbf{g}_i, \tag{7}$$

3. Experiments

3.1. Datasets

We conducted experiments on various public datasets as well as an in-house dataset.

TREC [2]: A question classification dataset containing 5952 sentences, which are classified into 6 coarse-grained classes, namely, abbreviation, entity, description, human, location and numeric.

TREC2 [2]: It contains the same sentences as **TREC**, but with 50 fine-grained classes annotated, such as numeric:temperature, numeric:distance, and entity:vehicle.

SST (Stanford Sentiment Treebank) [25]: A movie review sentiment corpus contains 11,855 sentences annotated with 5 labels, namely, very negative, negative, neutral, positive, and very positive.

SLU-UI: This is our in-house dataset for user intent classification in SLU. This corpus is collected through crowdsourcing, which makes sure the collected utterances are similar with the spoken language. This dataset contains 13,211 sentences with 6 domains (i.e. sms, phone, alarm, clock, etc.) and 25 user intent classes including sms-sent, phone-call, alarm-on, clock-

Model	SST	TREC	TREC2
Basic CNN [14]	48.0[*]	93.4[*]	86.4[†]
Basic CNN [14]($word2vec$, baseline)[‡]	48.8	94.0	87.0
Basic CNN [14]($GloVe$, baseline)[‡]	44.5	94.6	86.8
DCNN [15]	49.5	95.4	88.8
TBCNN [16]	**51.4**	**96.0**	–
RNN [6]	43.2	–	–
RNTN [25]	45.7	–	–
DRNN [26]	49.8	–	–
S-LSTM [10]	48.0	–	–
Tree-LSTM [9]	51.0	–	–
Paragraph-Vec [27]	48.7	–	–
ATT-CNN ($word2vec$)	50.4	95.4	88.6
ATT-CNN ($GloVe$)	49.2	**96.0**	**89.8**

[*] Reported by Kim [14].
[†] Reported by Ma et al. [15] using Kim's code (https://github.com/yoonkim/).
[‡] Re-duplicate by Kim's code.

Table 1: *Results of ATT-CNN against other models on public data.*

Model	Acc.
Basic CNN ($word2vec$)	93.6
DCNN [15]($word2vec$)	94.1[*]
ATT-CNN ($word2vec$)	**94.7**
Basic CNN ($GloVe$)	93.8
DCNN [15]($GloVe$)	**94.8**[*]
ATT-CNN ($GloVe$)	94.0

[*] Evaluate by using Ma's code (https://github.com/cosmmb/DCNN).

Table 2: *Results of ATT-CNN on in-house data.*

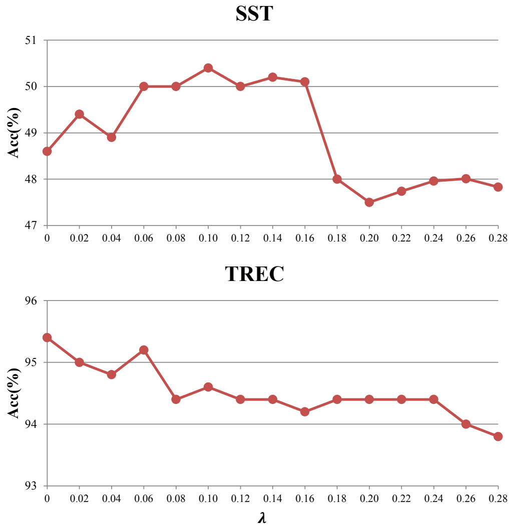


Figure 2: *Distance decay.*

check_time, etc. Training set consists of 11,875 sentences, and test set has 1,336 sentences.

We compared our model with the **basic CNN** [14] that only exploits consecutive word embedding, a variant that replace consecutive n-gram convolution with dependency n-gram convolution **DCNN** [15] and **TBCNN** [16]. In addition, we compare our model with three recursive-based methods and two recurrent-based models: The recursive-based models are **RNN** [6], **RNTN** [25] and **DRNN** [26]. The recurrent-based models are **Tree-LSTM** [9] and **S-LSTM** [10]. On the SLU-UI dataset, we compared our model with the **basic CNN** [14] and **DCNN** [15]. The Stanford dependency parser [28] is used for evaluating on DCNN. Due to the time limit, we didn't test Recursive NN and Recurrent NN.

We use the publicly available $word2vec$ and $GloVe$ word embedding, both of which have dimensionality of 300. The $word2vec$ was trained on 100B words from Google News using the continuous bag-of-words architecture [29]. The $GloVe$ was trained on the Common Crawl with 840B tokens [30]. Words not present in word embeddings are initialized randomly. For regularization, we adopt the following two ways: (1) employ random dropout on the fully connected layer [31]. (2) apply L2-norm penalty. Training is done through stochastic gradient descent over shuffled mini-batches with the Adadelta update rule [32].

3.2. Results and Discussions

The results on the public datasets is shown in Table 1 and we can observe that: (1) Our ATT-CNN achieves competitive performance with the Recursive NN, Recurrent NN and DCNN on the SST dataset and performs as well as DCNN and TBCNN on the TREC and TREC2 dataset, though our model does not use any external syntactic information which are necessary for the others. (2) Our ATT-CNN significantly outperforms the basic CNN on all of the datasets thanks to the attention mechanism. (3) Currently the best results we are aware of on the TREC and SST are 96.0% and 51.4% which is obtained by TBCNN [16]. However their model needs to parse sentence into syntactic tree as input and has a very complicated structure. Our model achieves the competitive performance with the state-of-the-art results with far less complexity.

Comparison between the proposed ATT-CNN with the basic CNN and DCNN on the SLU-UI dataset is shown in Table 2.

As we expected, our model achieves better performance comparing with basic CNN. But comparing with DCNN, ATT-CNN wins when using $word2vec$, but fails when using $GloVe$. The result illustrates: (1) both of DCNN and ATT-CNN outperform baseline indicates the information of non-consecutive words is also important in SLU. (2) DCNN performs quite well since our SLU-UI corpus does not contain many non-fluent utterances which are difficult for dependency parsing. (3) ATT-CNN is still competitive comparing with DCNN since ATT-CNN does not require additional information.

3.2.1. Effect of distance decay

This section presents the effect of applying distance decay on the TREC and SST datasets in Fig. 2. It can be easily found that distance decay λ performs differently on these two datasets. We think the reason why performance on the TREC drops as long as λ grows is that words in the sentence of TREC have closer relationship with each other due to the average sentence length is only 10 words, and setting λ to small values could make the model consider the context in a wider range. But on the SST dataset, the longer sentence length (19 words in average) indicates the relationship among words in one sentence is not as close as it is in TREC, so narrow down the context range by increasing λ a little bit could improve the performance. But we can see that keep on increasing λ also leads to performance degradation because only local context is considered.

3.2.2. Visualization of attention

Fig. 3 and 4 show the attention visualization of SST and TREC. Each row of the matrix denotes one α vector, and the darker of a cell the more attention is put on the word of the corresponding column. As Fig. 3 shows, the negative sentiment patterns *not ... decent* and *not ... clever* are quite significant in the first and second part of the sentence. In Fig. 4(a), a typical DESC pattern *what ... do* is top-scored by the attention model. In Fig. 4(b), *flower* getting more attention is a strong ENTY indication and in Fig. 4(c) a typical LOC word *habitat* get more attention. From Fig. 3 and 4, we can conclude that the correlation of a pair of words $(\mathbf{x}_i, \mathbf{x}_{j,j\neq i})$ can be considered as how much does $\mathbf{x}_i$ depend on $\mathbf{x}_{j,j\neq i}$ to indicate the corresponding sentence class.

4. Conclusion

We introduced attention mechanism to basic CNN to model the words correlations within sentence and proposed the ATT-CNN model. Our proposed model is able to capture long term contextual information and correlation between non-consecutive words without any syntactic information. The experiments demonstrate that our model significantly outperforms basic CNN, and achieves competitive performance with Recursive and Recurrent NNs on various datasets.

Currently we only consider the attention between word pairs, we will consider rich information into attention mechanism in the future work.

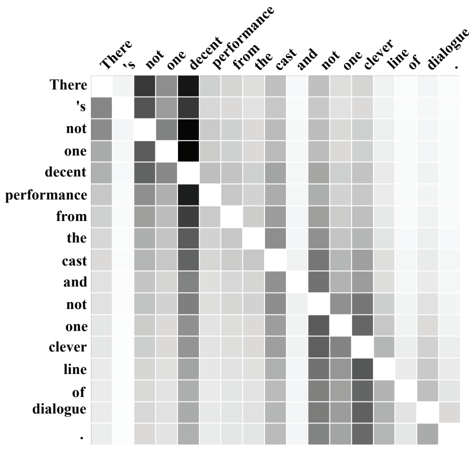
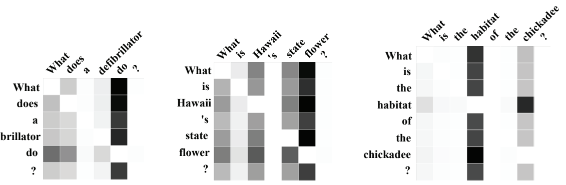


Figure 3: *Attention visualization of SST.*



(a) DESC* (b) ENTY† (c) LOC‡

* Question about description and abstract concepts.
† Question about entities.
‡ Question about locations.

Figure 4: *Attention visualization of TREC.*

5. References

[1] G. Tur, R. D. Mori, and Eds, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. New York, NY: John Wiley and Sons, 2011.

[2] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of COLING 2002*.

[3] P. Bo, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of EMNLP 2002*.

[4] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using crfs with hidden variables," in *Proceedings of NAACL 2010*, pp. 786–794.

[5] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, 2011.

[6] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of EMNLP 2011*.

[7] S. Lawrence, C. L. Giles, and S. Fong, "Natural language grammatical inference with recurrent neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 1, pp. 126–140, 2000.

[8] P. Blunsom, E. Grefenstette, and N. Kalchbrenner, "A convolutional neural network for modelling sentences," in *Proceedings of ACL 2014*.

[9] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of ACL 2015*.

[10] X. Zhu, P. Sobhani, and H. Guo, "Long short-term memory over tree structures," in *Proceedings of ICML 2015*.

[11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[12] Y. Shen, X. he, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of WWW 2014*.

[13] F. Meng, Z. Lu, M. Wang, H. Li, W. Jiang, and Q. Liu, "Encoding source language with convolutional neural network for machine translation," in *Proceedings of ACL 2015*.

[14] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of EMNLP 2014*.

[15] M. Ma, L. Huang, B. Xiang, and B. Zhou, “Dependency-based convolutional neural networks for sentence embedding,” in *Proceedings of ACL 2015*.

[16] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin, “Discriminative neural sentence modeling by tree-based convolution,” in *Proceedings of EMNLP 2015*.

[17] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proceedings of ICLR 2015*.

[18] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proceedings of ICML 2015*.

[19] A. Graves, “Generating sequences with recurrent neural networks,” 2013.

[20] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” in *Proceedings of NIPS 2014*.

[21] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, “End-to-end continuous speech recognition using attention-based recurrent nn: First results,” 2014.

[22] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Proceedings of NIPS 2015*.

[23] W. Yin, H. Schtüze, B. Xiang, and B. Zhou, “Abcnn: Attention-based convolutional neural network for modeling sentence pairs,” 2015, http://arxiv.org/abs/1512.05193.

[24] M. Luong, H. P. Christopher, and D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proceedings of EMNLP 2015*.

[25] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of EMNLP 2013*.

[26] O. Irsoy and C. Cardie, “Deep recursive neural networks for compositionality in language,” *Advances in Neural Information Processing Systems*, pp. 2096–2104, 2014.

[27] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of ICML 2014*.

[28] D. Chen and C. D. Manning, “A fast and accurate dependency parser using neural networks,” in *Proceedings of EMNLP 2014*.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of NIPS 2013*.

[30] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of EMNLP 2014*.

[31] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *CoRR*, vol. abs/1207.0580, 2014.

[32] M. Zeiler, “Adadelta: An adaptive learning rate method,” *CoRR*, vol. abs/1212.5701, 2012.