Increasing Learning Efficiency of Self-Attention Networks through Direct Position Interactions, Learnable Temperature, and Convoluted Attention

Philipp Dufter, Martin Schmitt, Hinrich Schütze
Center for Information and Language Processing (CIS), LMU Munich, Germany
{philipp,martin}@cis.lmu.de

Abstract

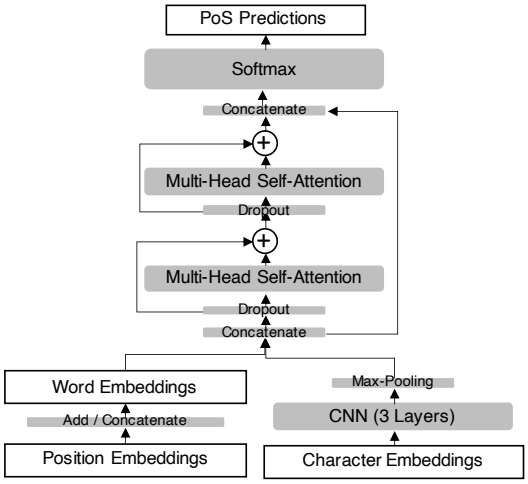
Self-Attention Networks (SANs) are an integral part of successful neural architectures such as Transformer (Vaswani et al., 2017), and thus of pretrained language models such as BERT (Devlin et al., 2019) or GPT-3 (Brown et al., 2020). Training SANs on a task or pretraining them on language modeling requires large amounts of data and compute resources. We are searching for modifications to SANs that enable faster learning, i.e., higher accuracies after fewer update steps. We investigate three modifications to SANs: direct position interactions, learnable temperature, and convoluted attention. When evaluating them on part-of-speech tagging, we find that direct position interactions are an alternative to position embeddings, and convoluted attention has the potential to speed up the learning process.

1 Introduction

Self-Attention mechanisms are at the core of successful neural network architectures in natural language processing (e.g., Transformer by Vaswani et al. (2017)). Compared to recurrent neural networks they do not have an inherent sequential bias, yet they allow the network to transfer knowledge across a sequence of length t in a constant number of steps. Typically, a self-attention layer consists of multiple attention heads and is itself part of a more sophisticated layer such as a Transformer Encoder Block.

We propose three minor modifications to the self-attention mechanism: (1) Position embeddings (Collobert et al., 2011; Vaswani et al., 2017) are used to inject positional information into SANs. We argue that learning position interactions can be modeled more directly than learning separate position embeddings and propose to replace embeddings with a **direct position interaction** matrix. (2) We hypothesize that spiky distributions generated by a softmax function within the attention head hinders the network from considering the broader sentence context effectively. Thus we introduce additional scalar parameters, a **learnable temperature**, that can support the network in using the context more effectively. (3) **Convoluted Attention:** attention matrices have been found to exhibit regular patterns (Clark et al., 2019; Kovaleva et al., 2019). A convolution which post-processes the attention matrix allows the network to detect attention patterns, and subsequently to reinforce or weaken attention scores.

We perform experiments on Part-of-Speech (PoS) tagging. We argue that a PoS model can only be successful for ambiguous and out-of-vocabulary tokens if it carefully considers and processes the context. Thus we consider PoS a suitable task to probe whether our modifications on the attention matrix enable more efficient learning. We perform experiments on the Penn Treebank (PTB) and on 47 languages of Universal Dependencies (UD). In short, our findings are: (i) Modeling absolute and relative position information through direct interaction matrices is a feasible alternative to position embeddings. (ii) Learnable temperature has almost no effect besides a small increase for out-of-vocabulary tokens. (iii) Convoluted attention achieves a higher accuracy after fewer epochs (more efficient learning) on PTB and has higher performance on UD. While results for convoluted attention are promising we are aware that only evaluating on PoS is a very restricted setting. Thus we plan to extend this study in future work.



(a) Model architecture. For simplicity only 2 Self-Attention layers are shown.

Hyperparameter	PTB	UD
# Layers	4	
Char CNN filter width	3	
# Char CNN filters	64	
# Attention Heads	4	
Embedding Dimension d	300	128
Position Emb. Dim.	300	128
Character Emb. Dim.	64	
Finetune Emb.	No	Yes
Use Pretrained Emb.	Yes	No
Finetune Position Emb.	Yes	
Finetune Character Emb.	Yes	
Vocabulary size	20000	50% of unique words
Max Sequence Length t	60	
Max Char Seq. Length	20	
Early Stopping	Yes, 3 epochs patience	
Activation Function	ReLU	
Dropout rate	0.1	
Optimizer	RMSprop with default parameters as in `keras.io/api/optimizers`	

(b) Model hyperparameters.

SAN	+PE[add]	+P	+R	+Temp	+Conv	+Conv2d
7,691,823	+18,000	+14,400	+480	+48	+173,760	+160

(c) Number of parameters for PTB when adding each extension to SAN. +R, +Temp and +Conv2d only add very few parameters to the model. SAN+P has less parameters than SAN+PE[ADD] when $t^2 \times n_{\text{heads}} < t \times d$.

Table 1: Model summary.

2 Methods

2.1 Model Architecture

To study our proposed modifications to self-attention we use a simple architectural setup; see Table 1a. Following embedding lookups for words and positions we deploy multiple layers of self-attention blocks and subsequently a softmax layer to get final PoS predictions. Our objective function is categorical cross-entropy. Character information is essential for PoS tagging (dos Santos and Zadrozny, 2014). To incorporate character information we follow (Yu et al., 2017) and use convolutional neural networks together with max-pooling to obtain a character level representation for words. We add/concatenate position embeddings to word embeddings and subsequently concatenate the character level word representation. We use a residual connection from the beginning to the end of the network and around each attention layer. See Table 1a and Table 1b for more details on the overall architecture and hyperparameters, and Table 1c for the number of parameters. We used common hyperparameters and did not tune them for higher performance.

2.2 Self-Attention

In this section we describe Self-Attention (Vaswani et al., 2017), for which we propose modifications in the following sections. We loosely follow the notation of Shaw et al. (2018) and define self attention as a function $att : \mathbb{R}^{t \times d} \rightarrow \mathbb{R}^{t \times d_h}$ where t is the sequence length, d the input dimension and d_h the output dimension. Consider an input $X \in \mathbb{R}^{t \times d}$ and weights $W_k, W_v, W_q \in \mathbb{R}^{d \times d_h}$. We denote the softmax function as σ. A scaled dot-product attention head is $Z := att(X) = \sigma(A)XW_v$ where $A = \sqrt{d_h}^{-1} XW_q(XW_k)^\intercal$ is the attention matrix and σ is applied along the horizontal axis. One self-attention layer consists of the concatenation of multiple attention heads. We call the model that adds (resp. concatenates) position embeddings to word embeddings *SAN+PE[add]* (resp. *SAN+PE[con]*).

2.3 Direct Position Interactions

It is well known that SANs are invariant with respect to reorderings of the input. To counteract this effect position embeddings, that are added or concatenated to the word embeddings, have been used (Collobert et al., 2011; Bahdanau et al., 2015). When adding position embeddings, parameters in form of a position embedding matrix $P \in \mathbb{R}^{t \times d}$ are added to the model. The corresponding position embeddings are then

added to token embeddings in the first layer. More specifically A is modified in the first layer to

$$A^{\text{PE[ADD]}} \sim (X+P)^\intercal W_q W_k^\intercal (X+P)^\intercal = \underbrace{XW_qW_k^\intercal X}_{\text{word-word} \sim A} + \underbrace{PW_qW_k^\intercal X^\intercal + XW_qW_k^\intercal P^\intercal}_{\text{word-position}} + \underbrace{PW_qW_k^\intercal P}_{\text{position-position}}.$$

We now propose to omit the word-position terms and replace the position-position term with the matrix $A^p \in \mathbb{R}^{t\times t}$. The values A^p_{ij} are learnable scalar values that directly model absolute positional interaction. Analogously we can introduce relative position embeddings by replacing position-position interaction with a matrix $A^r \in \mathbb{R}^{t\times t}$, where $A^r_{i,j} = a^r_{i-j+t}$ and $a^r \in \mathbb{R}^{2t}$ are the learnable parameters. We refer to these modifications as SAN+P and SAN+R, respectively. Absolute and relative position embeddings can then be easily combined by computing $A^{\text{P+R}} = A + A^p + A^r$, which we call SAN+P+R. Analogously to position embeddings that are only added in the first layer, we add A^p or A^r to the attention heads in the first layer. Note that the parameters A^p and A^r are not shared across attention heads.

2.4 Learnable Temperature

We propose to multiply each W_i with a trainable scalar weight γ_i for $i \in \{k, v, q\}$. We refer to this modification as learnable temperature, as $\gamma_k \times \gamma_q$ can be interpreted as a temperature of the softmax function used in attention. While it is related to normalization techniques, such as batch-, layer- or weight-normalization (Lei Ba et al., 2016; Salimans and Kingma, 2016; Ioffe and Szegedy, 2015), we only add a single learnable parameter per weight matrix and do not perform normalization. Normalization often involves complicating the objective function. We hypothesize that adding a learnable scalar value γ_i to scale weight matrices helps the network learn faster.

2.5 Convoluted Attention

We propose to process the matrix $\sigma(A)$ in convolutional layers, i.e., we create the matrix $A' = conv(\sigma(A))$. We experimented with having the convolution before taking the softmax, but this resulted in worse results. Note that after the convolution the attention scores are not normalized anymore. We apply both one and two dimensional convolutions (see Figure 1). This allows attention to reinforce neighborhood patterns, that have been identified e.g., by Clark et al. (2019; Kovaleva et al. (2019). Consider a sequence w_1, w_2, w_3 and assume attention weights are high for w_1, w_3 and low for w_2; then a convolutional filter can learn such a pattern and increase the attention weight for w_2 if this is beneficial for performance. For 1d convolution we use t convolutional filters per attention head to preserve the shape of the matrix. For 2d convolution we have one filter per attention head. This can be interpreted as a some sort of smoothing over the attention matrix. We use filter-width 3 in both cases.

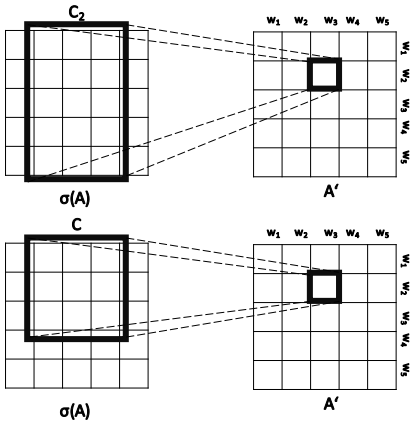


Figure 1: Applying 1d (top) and 2d (bottom) convolution on the attention matrix $\sigma(A)$ to get a postprocessed matrix A'. For 1d convolution we use one filter per row.

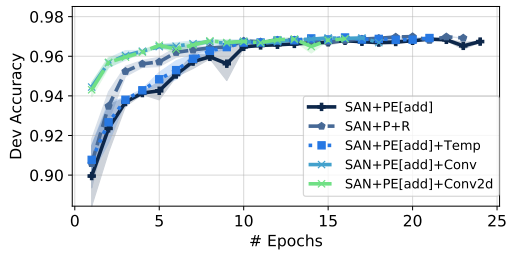
2.6 Data

PTB. We work on the WSJ section of the Penn-Treebank (PTB) (Marcus et al., 1993) with the usual data split (train: 0-18, dev: 19-21, tst: 22-24). We report accuracy across all words, out of vocabulary (OOV), and ambiguous words. We consider a word ambiguous if it has more than one unique PoS tag in the training data. We report mean and standard deviation (in subscript) across three random seeds. For pretrained word embeddings we use fastttext subword embeddings (Bojanowski et al., 2017).[1]

UD. We use version 2.2 of the Universal Dependencies as used in the CoNLL 2018 shared task (Zeman et al., 2018). We consider treebanks that have train, development, and test data and where results are reported in (Smith et al., 2018). This results in 47 treebanks.

[1] https://fasttext.cc/docs/en/english-vectors.html

3 Results



(a) Development accuracy on PTB during training.

		All	OOV	Ambig.
Baselines	SAN	94.62 $_{0.15}$	80.37 $_{0.72}$	91.36 $_{0.24}$
	SAN+PE[con]	96.81 $_{0.02}$	85.45 $_{0.58}$	95.16 $_{0.01}$
	SAN+PE[add]	96.82 $_{0.07}$	85.81 $_{0.26}$	95.19 $_{0.12}$
Methods	SAN+P	96.84 $_{0.10}$	85.69 $_{0.41}$	95.23 $_{0.17}$
	SAN+R	96.33 $_{0.09}$	83.70 $_{0.85}$	94.41 $_{0.12}$
	SAN+P+R	96.88 $_{0.05}$	85.71 $_{0.52}$	95.33 $_{0.05}$
	SAN+PE[add]+Temp	**96.93** $_{0.04}$	86.15 $_{0.26}$	95.37 $_{0.10}$
	SAN+PE[add]+Conv	**96.93** $_{0.07}$	86.41 $_{0.35}$	**95.38** $_{0.10}$
	SAN+PE[add]+Conv2d	96.88 $_{0.04}$	**86.48** $_{0.31}$	95.30 $_{0.08}$
	Meta-Bi-LSTM	97.96		
	Tag-Dictionary	92.70		

(b) Results for PTB. Bold: best result across the proposed methods. Meta-Bi-LSTM is by Bohnet et al. (2018), Tag-Dictionary by Huang et al. (2015).

	Model	All	OOV	Ambig.
Bl.	SAN+PE[add]	92.40 $_{0.31}$	77.89 $_{1.10}$	88.85 $_{0.35}$
Methods	SAN+P+R	92.82 $_{0.27}$	78.77 $_{0.97}$	89.44 $_{0.39}$
	SAN+PE[add]+Temp	92.49 $_{0.24}$	78.33 $_{1.01}$	88.85 $_{0.34}$
	SAN+PE[add]+Conv	94.21 $_{0.17}$	82.40 $_{0.65}$	92.20 $_{0.26}$
	SAN+PE[add]+Conv2d	**94.40** $_{0.17}$	**82.87** $_{0.69}$	**92.47** $_{0.29}$
	Uppsala	95.62		

(c) Results for UD. We report the macro mean across 47 treebanks. Subscript is average standard deviation. For reference we denote results by Smith et al. (2018) (Uppsala) in the last row.

UD Treebank	Baseline	Methods				
	SAN + PE[add]	SAN + P + R	SAN + PE[add] + Temp	SAN + PE[add] +Conv	SAN + PE[add] + Conv2d	Uppsala
Afrikaans-AfriBooms	92.11	92.02	92.06	94.50	**94.75**	96.28
Ancient,Greek-PROIEL	94.79	95.01	95.03	95.81	**95.99**	97.05
Ancient,Greek-Perseus	**89.50**	89.09	89.34	89.12	89.11	92.40
Arabic-PADT	94.21	94.05	94.14	95.22	**95.36**	90.70
Basque-BDT	90.86	91.09	90.68	92.38	**92.79**	96.05
Bulgarian-BTB	96.32	96.74	96.15	97.15	**97.37**	98.85
Chinese-GSD	88.54	88.45	88.36	**91.44**	91.42	89.15
Croatian-SET	95.65	95.81	95.70	96.23	**96.42**	97.93
Czech-CAC	97.66	97.79	97.80	98.10	**98.25**	99.17
Czech-FicTree	96.41	96.44	96.46	96.88	**97.07**	98.42
Danish-DDT	90.25	90.78	90.45	93.82	**94.50**	97.14
Dutch-Alpino	92.14	92.61	92.08	93.83	**93.88**	95.78
Dutch-LassySmall	92.17	92.18	92.28	**93.45**	93.22	96.18
English-GUM	89.42	88.93	89.17	**92.38**	92.35	94.67
English-LinES	89.38	89.53	88.69	93.34	**93.56**	96.47
Estonian-EDT	94.24	94.32	94.23	95.19	**95.30**	97.16
Finnish-FTB	90.12	89.89	90.31	91.72	**91.79**	96.30
Finnish-TDT	93.43	93.78	93.63	**94.01**	93.83	97.06
French-GSD	93.99	95.48	94.05	96.29	**96.33**	96.86
French-Sequoia	94.45	94.88	94.54	96.00	**96.37**	97.92
French-Spoken	86.50	88.27	87.93	91.17	**92.20**	95.51
German-GSD	90.57	91.32	90.72	92.26	**92.48**	94.02
Gothic-PROIEL	92.84	92.24	92.87	93.11	**93.79**	93.43
Greek-GDT	92.94	93.22	93.57	**95.02**	94.99	97.26
Hebrew-HTB	92.62	92.59	92.67	94.64	**94.90**	80.26
Hindi-HDTB	93.20	94.96	93.18	95.80	**95.86**	97.44
Hungarian-Szeged	87.38	88.90	88.76	89.47	**89.97**	94.60
Korean-Kaist	92.59	92.81	92.70	93.92	**93.95**	95.21
Latin-ITTB	95.94	96.40	95.75	97.04	**97.09**	98.34
Latin-PROIEL	93.83	93.74	93.89	94.12	**94.46**	96.21
Norwegian-Bokmaal	93.22	95.57	93.40	96.17	**96.30**	98.04
Norwegian-Nynorsk	92.33	94.64	92.52	**95.89**	95.85	97.57
Old,Church,Slavonic-PROIEL	92.37	91.99	92.64	93.36	**93.84**	95.76
Old,French-SRCMF	88.65	92.71	88.84	93.32	**93.71**	95.48
Persian-Seraji	94.53	94.22	94.57	95.69	**95.95**	96.79
Polish-LFG	96.29	96.20	96.21	96.49	**96.85**	98.57
Polish-SZ	94.57	94.11	94.49	95.17	**95.31**	97.95
Portuguese-Bosque	92.55	94.73	92.59	**96.05**	95.80	95.90
Serbian-SET	95.33	95.23	95.39	96.27	**96.56**	97.61
Slovak-SNK	92.49	91.39	92.68	93.42	**93.58**	96.57
Slovenian-SSJ	94.72	94.95	94.81	95.81	**96.06**	97.99
Spanish-AnCora	95.14	96.92	95.09	97.61	**97.65**	98.69
Swedish-LinES	88.94	88.95	88.70	92.72	**93.00**	96.64
Swedish-Talbanken	90.38	90.47	90.01	94.06	**94.55**	97.45
Ukrainian-IU	93.21	93.14	93.34	93.86	**94.41**	96.89
Urdu-UDTB	89.62	89.34	89.67	**92.31**	91.53	93.66
Vietnamese-VTB	84.42	84.77	84.78	86.29	**86.52**	78.89

(d) Results per treebank. Accuracy computed across all words.

Table 2: Results.

The plot in Table 2a shows development accuracy on PTB over training time. Convoluted attention yields a much steeper learning curve. Accuracy goes up quicker and the model seems to be converged after just 5 epochs. Learnable temperature does not have any visible effect on the learning curve and SAN+P+R seems to have a slightly steeper learning curve. After training for more than 10 epochs they converge to a similar performance. This is expected as our modifications do not make the model more expressive, they target more efficient learning.

Table 2b shows test results for PTB. All our modifications achieve comparable performance to SAN+PE[add]. Convoluted attention even achieves a slight performance improvement. Similarly learnable temperature has slightly higher performance for OOV. Replacing position embeddings both with absolute and relative direct position interactions is feasible and yields similar performance. Using only relative position interactions is slightly worse. It is surprising that +R reaches almost the same performance as +PE[add] with far less parameters. The combination, SAN+P+R, does not work better than just using SAN+P. Overall we reach a reasonable performance of almost 97%.

Table 2c, 3d show results across 47 treebanks. The overall conclusions are similar: Learnable temperature does not have any effect. SAN+P+R performs as well as SAN+PE[add] and indicates that this is a possible alternative to position embeddings. Surprisingly, convoluted attention yields much better results on UD with a 2 percentage point increase over SAN+PE[add]. Investigating the reason for this is

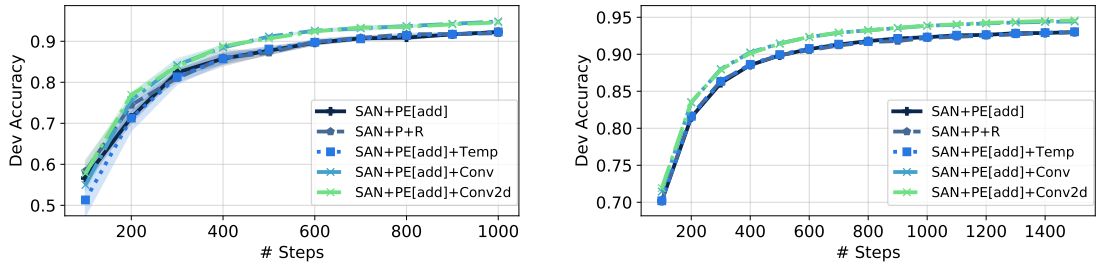


Figure 2: Development accuracy during training for PTB (left) and UD (right). For UD we report the average across languages. Compared to Table 2a this is a more detailed view on the first training steps.

part of future work. Both 1d and 2d convolution perform similar. +Conv2d yields performance improvements with only 10 additional parameters per attention head added in the model (see Table 1c). This indicates that our hypothesis that convolutions are suitable to reinforce patterns in the attention matrix is reasonable.

Figure 2 shows the learning curves for PTB and UD for the first training steps (around 1 epoch for PTB). One can see that convoluted attention exhibits a somewhat steeper learning curve from the very beginning, but the overall effect is more visible in Table 2a.

4 Related Work

Many variants of positional embeddings have been explored: Vaswani et al. (2017) reported on sinusoidal and learned position embeddings, Shaw et al. (2018) explored relative position embeddings and Shen et al. (2018) introduced directional self-attention. We propose to replace traditional position embeddings by a direct position interaction matrix. Recently Raffel et al. (2020) proposed to model relative positions with scalar values, an idea also investigated by Schmitt et al. (2020). This approach is similar to our SAN+R. Contemporary to this submission, Ke et al. (2020) proposed TUPE, which is similar to SAN+P. They also find it to be a feasible alternative to position embeddings and report slight performance increases. In contrast to weight normalization (Salimans and Kingma, 2016), a related method to learnable temperature, we do not normalize the weight matrices. Instead we only add a learnable scalar parameter and observed that normalizing the weights actually harms performance. Lin et al. (2018) introduced a self-adaptive temperature. However, they focused on parametrizing the temperature of timestep t using the activations from timestep $t-1$. Contemporary to this work, Henry et al. (2020) proposed query-key normalization in Transformers. There is range of work trying to combine attention with convolution (Yin and Schütze, 2018; Yu et al., 2018). We are not aware of any work that applies convolution directly to attention weights.

5 Conclusion

We conclude that position embeddings can be replaced with direct position interactions.[2] Learnable temperature has almost no effect. Convoluted attention speeds up learning on PTB and yields better results on UD. We are aware that this paper is a small study with limited validity as it considers only one task. Given that convoluted attention yielded promising results, we plan to extend this line of experiments to additional tasks and architectures in future work. Our code is available.[3]

Acknowledgements

We gratefully acknowledge funding through a Zentrum Digitalisierung.Bayern fellowship awarded to the first author. This work was supported by the European Research Council (# 740516). We thank the anonymous reviewers for valuable comments.

[2] An original draft of this paper has been created early 2019 and was first submitted to a conference in 2019. Since then some of its results have also been explored in recent work such as (Raffel et al., 2020; Ke et al., 2020).

[3] https://github.com/pdufter/convatt

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia, July. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *ICML*, pages 1818–1826.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. 2020. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul. PMLR.

Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking the positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China, November. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Junyang Lin, Xu Sun, Xuancheng Ren, Muyu Li, and Qi Su. 2018. Learning when to concentrate or divert attention: Self-adaptive attention temperature for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2985–2990, Brussels, Belgium, October-November. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tim Salimans and Durk P Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 901–909. Curran Associates, Inc.

Martin Schmitt, Leonardo FR Ribeiro, Philipp Dufter, Iryna Gurevych, and Hinrich Schütze. 2020. Modeling graph structure via relative position for better text generation from knowledge graphs. *arXiv preprint arXiv:2006.09242*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June. Association for Computational Linguistics.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium, October. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wenpeng Yin and Hinrich Schütze. 2018. Attentive convolution: Equipping CNNs with RNN-style attention mechanisms. *Transactions of the Association for Computational Linguistics*, 6:687–702.

Xiang Yu, Agnieszka Falenska, and Ngoc Thang Vu. 2017. A general-purpose tagger with convolutional neural networks. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 124–129, Copenhagen, Denmark, September. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *International Conference on Learning Representations (ICLR)*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium, October. Association for Computational Linguistics.