Rethinking Complex Neural Network Architectures for Document Classification

Ashutosh Adhikari,* **Achyudh Ram**,* **Raphael Tang**, and **Jimmy Lin**
David R. Cheriton School of Computer Science
University of Waterloo
{adadhika, arkeshav, r33tang, jimmylin}@uwaterloo.ca

Abstract

Neural network models for many NLP tasks have grown increasingly complex in recent years, making training and deployment more difficult. A number of recent papers have questioned the necessity of such architectures and found that well-executed, simpler models are quite effective. We show that this is also the case for document classification: in a large-scale reproducibility study of several recent neural models, we find that a simple BiLSTM architecture with appropriate regularization yields accuracy and F_1 that are either competitive or exceed the state of the art on four standard benchmark datasets. Surprisingly, our simple model is able to achieve these results *without* attention mechanisms. While these regularization techniques, borrowed from language modeling, are not novel, to our knowledge we are the first to apply them in this context. Our work provides an open-source platform and the foundation for future work in document classification.

1 Introduction

Recent developments in neural architectures for a wide range of NLP tasks can be characterized as a drive towards increasingly complex network components and modeling techniques. Worryingly, these new models are accompanied by smaller and smaller improvements in effectiveness on standard benchmark datasets, which leads us to wonder if observed improvements are "real". There is, however, ample evidence to the contrary. To provide a few examples: Melis et al. (2018) report that standard LSTM architectures outperform more recent models when properly tuned. Vaswani et al. (2017) show that sequence transduction using encoder–decoder networks with attention mechanisms work just as well with the attention module only, making most of the complex neural machinery unnecessary. Mohammed et al. (2018) show that simple RNN- and CNN-based models yield accuracies rivaling far more complex architectures in simple question answering over knowledge graphs.

Perhaps most damning are the indictments of Sculley et al. (2018), who lament the lack of empirical rigor in our field and cite even more examples where improvements can be attributed to far more mundane reasons (e.g., hyperparameter tuning) or are simply noise. Lipton and Steinhardt (2018) concur with these sentiments, adding that authors often use fancy mathematics to obfuscate or to impress (reviewers) rather than to clarify. Complex architectures are more difficult to train, more sensitive to hyperparameters, and brittle with respect to domains with different data characteristics—thus both exacerbating the "crisis of reproducibility" and making it difficult for practitioners to deploy networks that tackle real-world problems in production environments.

Like the papers cited above, we question the need for overly complex neural architectures, focusing on the problem of document classification. Starting with a large-scale reproducibility study of several recent neural models, we find that a simple bi-directional LSTM (BiLSTM) architecture with appropriate regularization yields accuracy and F_1 that are either competitive or exceed the state of the art on four standard benchmark datasets. As the closest comparison point, we find no benefit to the hierarchical modeling proposed by Yang et al. (2016) and we are able to achieve good classification results *without* attention mechanisms. While these regularization techniques, borrowed from language modeling, are not novel, we are to our knowledge the first to apply them in this context. Our work provides an open-source platform and the foundation for future work in document classification.

* Equal contribution.

Proceedings of NAACL-HLT 2019, pages 4046–4051
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

2 Background and Related Work

2.1 Document Classification

Over the last few years, deep neural networks have achieved the state of the art in document classification. One popular model, hierarchical attention network (HAN), uses word- and sentence-level attention in classifying documents (Yang et al., 2016). Although this model nicely captures the intuition that modeling word sequences in sentences should be handled separately from sentence-level discourse modeling, one wonders if such complex architectures are really necessary, especially given the size of training data available today.

An important variant of document classification is the multi-label, multi-class case. Liu et al. (2017) develop XML-CNNs for multi-label text classification, basing the architecture on Kim-CNN (Kim, 2014) with increased filter sizes and an additional fully-connected layer. They also incorporate dynamic adaptive max-pooling (Chen et al., 2015) instead of the vanilla max-pooling over time in KimCNN. The paper compares with CNN-based approaches for the multi-label task, but only reports precision and disregards recall. Yang et al. (2018) instead adopts encoder–decoder sequence generation models (SGMs) for generating multiple labels for each document. Similar to our critique of HAN, we opine against the high complexity of these multi-label approaches.

2.2 Regularizing RNNs

There have been attempts to extend dropout (Srivastava et al., 2014) from feedforward neural networks to recurrent ones. Unfortunately, direct application of dropout on the hidden units of an RNN empirically harms its ability to retain long-term information (Zaremba et al., 2014). Recently, however, Merity et al. (2018) successfully apply dropout-like techniques to regularize RNNs for language modeling, achieving competitive word-level perplexity on multiple datasets. Inspired by this development, we adopt two of their regularization techniques, embedding dropout and weight-dropped LSTMs, to our task of document classification.

Weight-dropped LSTM. LSTMs comprise eight total input–hidden and hidden–hidden weight matrices; in weight dropping, Merity et al. (2018) regularize the four hidden–hidden matrices with DropConnect (Wan et al., 2013). The operation is applied only once per sequence, using the same dropout mask across multiple timesteps. Conveniently, this allows practitioners to use fast, out-of-the-box LSTM implementations without affecting the RNN formulation or training performance.

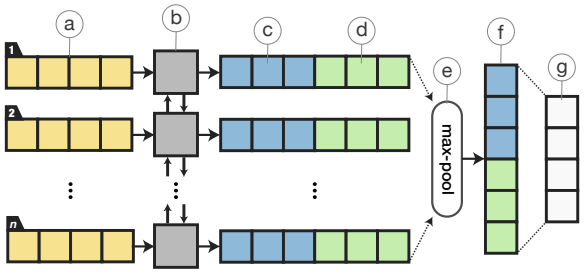

Figure 1: Illustration of the model architecture, where the labels are the following: (a) input word embeddings (b) BiLSTM (c, d) concatenated forward $h^f_{1:n}$ and backward $h^b_{1:n}$ hidden features (e) max-pooling over time (f) document feature vector (g) softmax or sigmoid output.

Embedding Dropout. Introduced in Gal and Ghahramani (2016) and successfully employed for neural language modeling (Merity et al., 2018), embedding dropout performs dropout on entire word embeddings, effectively removing some of the words at each training iteration. As a result, the technique conditions the model to be robust against missing input; for document classification, this discourages the model from relying on a small set of input words for prediction.

3 BiLSTM Model

We design our model to be minimalistic: First, we feed the word embeddings $\mathbf{w}_{1:n}$ of a document to a single-layer BiLSTM, extracting concatenated forward and backward word-level context vectors $\mathbf{h}_{1:n} = \mathbf{h}^f_{1:n} \oplus \mathbf{h}^b_{1:n}$. Subsequently, we max-pool $\mathbf{h}_{1:n}$ across time to yield document vector $\mathbf{d}$—see Figure 1, labels a–f. Finally, we feed $\mathbf{d}$ to a sigmoid or a softmax layer over the labels, depending on if the task type is multi-label or single-label classification (label g).

Contrary to prior art, our approach refrains from attention, hierarchical structure, and sequence generation, each of which increases model complexity. For one, hierarchical structure requires sentence-level tokenization and multiple RNNs. For another, sequence generation uses an encoder–decoder architecture, reducing computational parallelism. All three methods add depth to the model; our approach instead uses a single-layer BiLSTM with trivial max-pooling and concatena-

tion operations, which makes for both simple implementation and resource-efficient inference.

4 Experimental Setup

We conduct a large-scale reproducibility study involving HAN, XML-CNN, KimCNN, and SGM. These are compared to our proposed model, referred to as LSTM$_{reg}$, as well as an ablated variant without regularization, denoted LSTM$_{base}$. The implementation of our model as well as from-scratch reimplementations of all the comparison models (except for SGM) are provided in our toolkit called Hedwig, which we make publicly available to serve as the foundation for future work.[1] In addition, we compare the neural approaches to logistic regression (LR) and support vector machines (SVMs). The LR model is trained using a one-vs-rest multi-label objective, while the SVM is trained with a linear kernel. Both of these methods use word-level tf–idf vectors of the documents as features.

All of our experiments are performed on Nvidia GTX 1080 and RTX 2080 Ti GPUs, with PyTorch 0.4.1 as the backend framework. We use Scikit-learn 0.19.2 for computing the tf–idf vectors and implementing LR and SVMs.

4.1 Datasets

We evaluate our models on the following four datasets: Reuters-21578, arXiv Abstract Paper dataset (AAPD), IMDB, and Yelp 2014. Reuters and AAPD are multi-label datasets, whereas IMDB and Yelp are single-label ones. For IMDB and Yelp, we use random sampling to split the dataset such that 80% is used for training, 10% for validation, and 10% for test. We use the standard ModApte splits (Apté et al., 1994) for the Reuters dataset, and author-defined splits for AAPD (Yang et al., 2018). We summarize the statistics of these datasets in Table 1.

Unfortunately, there is little consensus within the natural language processing community for choosing the splits of IMDB and Yelp 2014. Furthermore, they are often unreported in modeling papers, hence preventing direct comparison with past results. We are not able to find the exact splits Yang et al. (2016) use; for consistency, we use the same proportion the authors report, but of course this yields different samples in each split.

[1] http://hedwig.ca

Dataset	C	N	W	S
Reuters	90	10,789	144.3	6.6
AAPD	54	55,840	167.3	1.0
IMDB	10	135,669	393.8	14.4
Yelp 2014	5	1,125,386	148.8	9.1

Table 1: Summary of the datasets. C denotes the number of classes in the dataset, N the number of samples, and W and S the average number of words and sentences per document, respectively.

For the multi-label datasets, we report the well-known micro-averaged F_1 score, which is the class-weighted harmonic mean between recall and precision. For the single-label datasets, we compare the models using accuracy.

4.2 Training and Hyperparameters

To ensure a fair comparison, we tune the hyperparameters for all baseline models. For HAN, we use a batch size of 32 across all the datasets, with a learning rate of 0.01 for Reuters and 0.001 for the rest. To train XML-CNN, we select a dynamic pooling window length of eight, a learning rate of 0.001, and 128 output channels, with batch sizes of 32 and 64 for single-label and multi-label datasets, respectively. For KimCNN, we use a batch size of 64 with a learning rate of 0.01. For training SGM on Reuters, we use the source code provided by the authors[2] and follow the same hyperparameters in their paper (Yang et al., 2018). For the LR and SVM models, we use the default set of hyperparameters in Scikit-learn.

For LSTM$_{reg}$ and LSTM$_{base}$, we use the Adam optimizer with a learning rate of 0.01 on Reuters and 0.001 on the rest of the datasets, using batch sizes of 32 and 64 for multi-label and single-label tasks, respectively. For LSTM$_{reg}$, we also apply temporal averaging (TA): as shown in Kingma and Ba (2014), TA reduces both generalization error and stochastic noise in recent parameter estimates from stochastic approximation. We set the default TA exponential smoothing coefficient of β_{EMA} to 0.99. We choose 512 hidden units for the Bi-LSTM models, whose max-pooled output is regularized using a dropout rate of 0.5. We also regularize the input–hidden and hidden–hidden Bi-LSTM connections using embedding dropout and weight dropping, respectively, with dropout rates of 0.1 and 0.2.

[2] https://github.com/lancopku/SGM

#	Model	Reuters		AAPD		IMDB		Yelp '14	
		Val. F_1	Test F_1	Val. F_1	Test F_1	Val. Acc.	Test Acc.	Val. Acc.	Test Acc.
1	LR	77.0	74.8	67.1	64.9	43.1	43.4	61.1	60.9
2	SVM	89.1	86.1	71.1	69.1	42.5	42.4	59.7	59.6
3	KimCNN Repl.	83.5 ±0.4	80.8 ±0.3	54.5 ±1.4	51.4 ±1.3	42.9 ±0.3	42.7 ±0.4	66.5 ±0.1	66.1 ±0.6
4	KimCNN Orig.	–	–	–	–	–	37.6††	–	61.0††
5	XML-CNN Repl.	88.8 ±0.5	86.2 ±0.3	70.2 ±0.7	68.7 ±0.4	–	–	–	–
6	HAN Repl.	87.6 ±0.5	85.2 ±0.6	70.2 ±0.2	68.0 ±0.6	51.8 ±0.3	51.2 ±0.3	68.2 ±0.1	67.9 ±0.1
7	HAN Orig.	–	–	–	–	–	49.4‡	–	**70.5**‡
8	SGM Orig.	82.5 ±0.4	78.8 ±0.9	–	**71.0**†	–	–	–	–
9	LSTM_{base}	87.6 ±0.2	84.9 ±0.3	72.1 ±0.4	69.6 ±0.4	52.5 ±0.2	52.1 ±0.3	68.6 ±0.1	68.4 ±0.1
10	LSTM_{reg}	**89.1** ±0.8	**87.0** ±0.5	**73.1** ±0.4	70.5 ±0.5	**53.4** ±0.2	**52.8** ±0.3	**69.0** ±0.1	68.7 ±0.1

Table 2: Results for each model on the validation and test sets; best values are bolded in blue. *Repl.* reports mean ± SD of five runs from our reimplementations; *Orig.* refers to point estimates from †Yang et al. (2018), ‡Yang et al. (2016), and ††Tang et al. (2015).

For our optimization objective, we use cross-entropy and binary cross-entropy loss for single-label and multi-label tasks, respectively. On all datasets and models, we use 300-dimensional word vectors (Mikolov et al., 2013) pre-trained on Google News. We train all neural models for 30 epochs with five random seeds, reporting the mean validation set scores and their corresponding test set results.

Toward Robust Baselines. Recently, reproducibility is becoming a growing concern for the NLP community (Crane, 2018). Indeed, very few of the papers that we consider in this study report validation set results, let alone run on multiple seeds. In order to address these issues, we report scores on both validation and test sets for our reimplementations; doing so is good practice, since it reinforces the validity of the experimental results and claims. We also provide the standard deviation of the scores across different seeds to demonstrate the stability of our results. This is in line with previous papers (Zhang and Wallace, 2017; Reimers and Gurevych, 2017; Crane, 2018) that emphasize reporting variance for robustness against potentially spurious conclusions.

5 Results and Discussion

We report the mean and standard deviation (SD) of the F_1 scores and accuracy for all five runs in Table 2. For HAN and KimCNN, we include results from the original papers to validate our reimplementation. We fail to replicate the reported results of SGM on AAPD using the authors' codebase and data splits.[3] As a result, we simply copy the value reported in Yang et al. (2018) in Table 2, row 8, which represents their maximum F_1 score. To verify the correctness of our HAN and KimCNN reimplementations, we compare the differences in F_1 and accuracy on the appropriate datasets. We attribute the small differences to using different dataset splits (see Section 4.1) and reporting mean values.

Baseline Comparison. We see that our simple LSTM_{reg} model achieves state of the art on Reuters and IMDB (see Table 2, rows 9 and 10), establishing mean scores of 87.0 and 52.8 for F_1 score and accuracy on the test sets of Reuters and IMDB, respectively. This highlights the efficacy of proper regularization and optimization techniques for the task. We observe that LSTM_{reg} consistently improves upon the performance of LSTM_{base} across all of the tasks—see rows 9 and 10, where, on average, regularization yields increases of 1.5 and 0.5 points for F_1 score and accuracy, respectively.

A few of our LSTM_{reg} runs attain state-of-the-art test F_1 scores on AAPD. However, in the interest of robustness, we report the mean value, as mentioned in Section 4.2. We also find the accuracy of LSTM_{reg} and our reimplemented version of HAN on Yelp 2014 to be almost two points lower than the copied result of HAN (rows 6, 7, and 10) from Yang et al. (2016). On the other hand, both of the models surpass the original result by nearly two points for the IMDB dataset. We cannot rule out that these disparities are caused

[3] The authors did not answer our e-mails seeking assistance.

by the absence of any widely-accepted splits for evaluation on Yelp 2014 and IMDB (as opposed to model or implementation differences).

Interestingly, the non-neural LR and SVM baselines perform remarkably well. On Reuters, for example, the SVM beats many neural baselines, including our non-regularized LSTM$_{base}$ (rows 2–9). On AAPD, the SVM either ties or beats the other models, losing only to SGM (rows 2–8). Compared to the SVM, the LR baseline appears better suited for the single-label datasets IMDB and Yelp 2014, where it achieves better accuracy than the SVM does.

6 Conclusions and Future Work

In this paper, we question the complexity of existing neural network architectures for document classification. To demonstrate the effectiveness of proper regularization and optimization, we apply embedding dropout, weight dropping, and temporal averaging when training a simple BiLSTM model, establishing either competitive or state-of-the-art results on multiple datasets.

One potential extension of this work is to conduct a comprehensive ablation study to determine the relative contribution of each of the regularization and optimization techniques. Furthermore, it would be interesting to compare these techniques to the recent line of research in deep language representation models, such as Embeddings from Language Models (ELMo; Peters et al., 2018) and pre-trained transformers (Devlin et al., 2018; Radford, 2018). Finally, the examined regularization and optimization methods deserve exploration in other NLP tasks as well.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. We thank Nabiha Asghar for providing us with the Yelp 2014 dataset. We also thank the anonymous reviewers for their valuable comments.

References

Chidanand Apté, Fred Damerau, and Sholom M Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural 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)*, volume 1, pages 167–176.

Matt Crane. 2018. Questionable answers in question answering research: reproducibility and variability of published results. *Transactions of the Association for Computational Linguistics*, 6:241–252.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

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.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Zachary C. Lipton and Jacob Steinhardt. 2018. Troubling trends in machine learning scholarship. *arXiv:1807.03341v2*.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

Gbor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.

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.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. 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 291–296.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348.

D. Sculley, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. 2018. Winner's curse? On pace, progress, and empirical rigor. In *Proceedings of the 6th International Conference on Learning Representations, Workshop Track (ICLR 2018)*.

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(1):1929–1958.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv:1409.2329*.

Ye Zhang and Byron Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 253–263.