Rethinking Complex Neural Network Architectures for

Document Classification

1 Introduction

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 F1 that are either competitive or exceed the state of the art on four standard benchmark datasets.

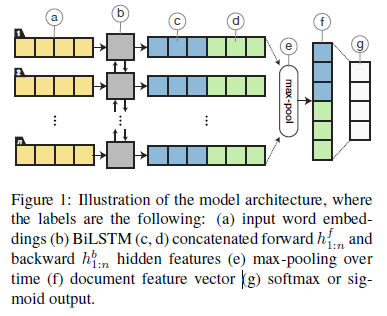
2 Background and Related Work

2.1 Document Classification

2.2 Regularizing RNNs

Weight-dropped LSTM

Embedding Dropout



3 BiLSTM Model

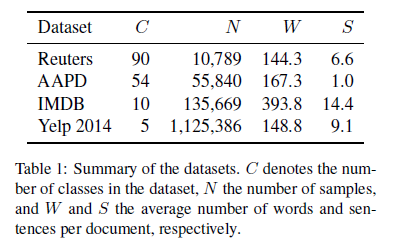
Contrary to prior art, our approach refrains from attention, hierarchical structure, and sequence generation, each of which increases model complexity. Contrary to prior art, our approach refrains from attention, hierarchical structure, and sequence generation, each of which increases model complexity. 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 concatenation operations, which makes for both simple implementation and resource-efficient inference.

4 Experimental Setup

4.1 Datasets

For the multi-label datasets, we report the well-known micro-averaged F1 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  
5 Results and Discussion

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.

Finally, the examined regularization and optimization methods deserve exploration in other NLP tasks as well.