Evaluating the Effects of Bidirectional Inputs on Classification Tasks with LSTM

Luheng Wang
New York University
lw2534@nyu.edu

Yilun Kuang
New York University
yk2516@nyu.edu

Youqing Liang
New York University
jl8015@nyu.edu

Abstract

Many modern NLP systems perform bidirectional encoding of the input sentences for integrating contextual information. It remains unclear whether bidirectional inputs perform similarly as bidirectional models. This study investigates the role of bidirectional inputs by training the forward LSTM with unidirectional and bidirectional input sentences from CoLA, spam email, and COVID sentiment analysis datasets. By eliminating potential confounders, we establish that bidirectional inputs are causal factors for the improvement in classification performance for LSTM on certain tasks. We also present a qualitative analysis for the misclassified examples in the context of unidirectional and bidirectional inputs.

1 Introduction

For natural language processing applications, the general principle of bidirectionality underlies many aspects of model designs and training procedures. Early works in word embedding like word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) that capture individual word semantics have been extended to integrate contextual information via the introduction of context2vec that surpasses traditional word-embedding (Melamud et al., 2016). The model architecture of Bidirectional-LSTM based on Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) outperforms unidirectional models (Schuster and Paliwal, 1997). Recent advances in Transformer-based language model like BERT (Devlin et al., 2019) also integrate bidirectional encoding in the pretraining pipeline and is crucial to its SOTA performance in various natural language benchmarks (Vaswani et al., 2017).

The mysterious effectiveness of NLP models with bidirectional designs have largely been left with intuitive explanations of contextual importance. It remains unclear whether and if bidirectional inputs contribute to the modeling success. This study investigates the difference in LSTM model performance given unidirectional or bidirectional inputs. A sequence of words with its duplicate inverse is used as a bidirectional input compared to a sequence of words itself as a unidirectional input.

More specifically, consider a sequence of words ”I feel hell good.” The sequence with its duplicate inverse becomes ”I feel hell good good hell feel I.” Additional sequences like ”good hell feel I” and ”I feel hell good I feel hell good” are constructed. The latter two variations aim to eliminate the confounds of possible inherent advantages of right-to-left information and potential benefits of longer sentence for better model performances. This paper implements the single direction forward LSTM trained on a spam classification dataset, CoLA, and an India COVID sentiment analysis dataset with the above four different input conditions. The details of the experimental setup are described in the method section.

Overall, LSTM with bidirectional inputs achieves a 1 % increase in classification accuracy for the spam dataset and a 3.9% increase for the COVID dataset. The classification performance is boosted given the bidirectional inputs.

2 Data

We make use of three datasets. First, we experiment on The Corpus of Linguistic Acceptability dataset (Warstadt et al., 2018, CoLA). It is a dataset of 10657 texts annotated for their grammatical acceptability, which is a binary classification task (acceptable and not acceptable). Second, we use an Indian Covid sentiment analysis dataset [1] from

[1] https://www.kaggle.com/surajkum1198/twitterdata

Kaggle. It is a four-classes classification dataset, which contains approximately 3000 tweets from Indian users annotated for sentiment (anger, sad, joy, and fear). Lastly, we incorporate a spam classification task dataset [2] from Kaggle, which includes around 5500 data of email text.

Moreover, for each of the three datasets, we create three variation datasets. Variation dataset 1 (V_1) is the reverse of the original dataset (V_o). For example, suppose there exists a data text "this is terribly exciting" in V_o, then the corresponding data text in V_1 will be "exciting terribly is this". Variation dataset 2 (V_2) is the doubled version of V_o. For instance, suppose there exists a data text "this is terribly exciting" in V_o, then the corresponding data text in V_2 will be "this is terribly exciting this is terribly exciting". Lastly, variation dataset 3 (V_3) is the concatenation of V_1 and V_2. For instance, suppose there exists a data text "this is terribly exciting" in V_o, then the corresponding data text in V_3 will be "this is terribly exciting exciting terribly is this". The details of the three datasets are shown in Table 1. Note that consider all three datasets: Spam, Covid, and CoLA, each of them has three variation datasets. Thus, in total there exists 12 datasets - 3 original, 9 variations we constructed. The rationale of our design will be explained in the Method section.

3 Method

This study considers the LSTM performance on the above V_0, V_1, V_2 and V_3 datasets for each of the three datasets mentioned in Section 2.

3.1 LSTM

Consider a sequence T words $\mathcal{W} = \{w_t\}_{t=1,\ldots,T}$, the function LSTM : $\mathcal{W} \to \mathcal{H}$ computes a set of T hidden representations $\mathcal{H} = \{h_t\}_{t=1,\ldots,T}$, where $h_t = \overrightarrow{\text{LSTM}}(\{w_t\}_{t=1,\ldots,T})$ is computed by a forward LSTM (Conneau et al., 2018). A sentence is represented by the last hidden vector h_T (Conneau et al., 2018).

3.2 LSTM with Varying Inputs

Given the forward LSTM with the original dataset V_0

$$\overrightarrow{h_{t;\,V_0}} = \overrightarrow{\text{LSTM}}\Big(\{w_t\}_{t=1,\ldots,T}\Big), \tag{1}$$

[2]https://www.kaggle.com/uciml/sms-spam-collection-dataset/data

the forward LSTM with the reverse dataset V_1 is defined as

$$\overrightarrow{h_{t;\,V_1}} = \overrightarrow{\text{LSTM}}\Big(\text{flip}(\{w_t\}_{t=1,\ldots,T})\Big), \tag{2}$$

for some flip() function that flips word tokens of a given sequence. Now define the double dataset V_2 to be $\mathcal{W}_{\text{double}} = \Big\{\{w_t\}_{t=1,\ldots,T}, \{w_t\}_{t=1,\ldots,T}\Big\}$, then we have

$$\overrightarrow{h_{t;\,V_2}} = \overrightarrow{\text{LSTM}}\Big(\mathcal{W}_{\text{double}}\Big). \tag{3}$$

For the concatenated dataset V_3 with $\mathcal{W}_{\text{concat}} = \Big\{\{w_t\}_{t=1,\ldots,T}, \text{flip}(\{w_t\}_{t=1,\ldots,T})\Big\}$, we have

$$\overrightarrow{h_{t;\,V_3}} = \overrightarrow{\text{LSTM}}\Big(\mathcal{W}_{\text{concat}}\Big). \tag{4}$$

.

3.3 Model Comparison

To establish the importance of bidirectional inputs in LSTM classification performance, we need to have an improvement of classification accuracy based on $\overrightarrow{h_{t;\,V_3}}$ over $\overrightarrow{h_{t;\,V_0}}$. There are several confounders that could potentially contribute to the better performance of LSTM with the dataset V_3 than the dataset V_1.

First, there might be an inherent advantage of right-to-left inputs for natural language processings than left-to-right inputs. Consider the sentence "I am terribly exciting". The most informative word "exciting" for sentiment analysis is situated in the last position of the sentence. It is possible that A language model trained on the reverse version of the original sentence is inherently better. To eliminate this confounding factor, we design V_1 as a reverse version of V_0. If there is no substantial improvement of LSTM trained on V_1 over V_0, we consider the reverse order in itself is not crucial for model performance.

Second, it is possible that LSTM trained on V_3 performs better than LSTM trained on V_0 due to more information presented in V_3 as sentence concatenation provides more tokens. Consider again the sentence "I love the food here, it is terribly good". The doubled version "I love the food here, it is terribly good I love the food here, it is terribly good" increases the numbers of positive words in the sentence two folds. The model performance

Variation	V_o	V_1	V_2	V_3
Pattern	original	inverse	original+original	original+inverse
Example	this is terribly exciting	exciting terribly is this	this is terribly exciting this is terribly exciting	this is terribly exciting exciting terribly is this

Table 1: Examples of texts in each of the four datasets (1 original dataset, 3 variation datasets)

could be boosted due to inherently more information. To eliminate this confound, we compare models trained on V_2 and V_0. If there is no substantial improvement of LSTM trained on V_2 over V_0, we consider the double information in itself is not crucial for model performance.

By eliminating the above two confounds, we can conclude that bidirectional inputs does improve model performance if there is an actual increase in the classification metrics. The detailed comparison standard is given in the experiment section below.

4 Experiment Design and Baselines

We have three baselines, which are the testing performance of LSTM trained on each of the three original datasets V_o mentioned in 2. LSTM is configured to have 1 hidden layer of hidden size of 32, batch maximum sentence length 128, and dropout probability 0.3.

Our experiment first records the three baseline results as plain accuracy. Note that in the leaderboard for CoLA, the metric recommended for CoLA is Matthews Correlation Coefficient (MCC). However, since our goal is to compare the testing results of models trained on different training datasets instead of trying to achieve leaderboard-level performance, we choose to use simple plain accuracy for straight-forward comparison. Second, we train LSTM with the same configuration on V_1 and record the difference Δ_{1o} between the testing performances of LSTM trained on V_1 and V_o. Third, we repeat the process for V_2 with the same training, validation, and testing split, and obtain Δ_{2o} which is the difference between the testing performances of LSTM trained on V_2 and V_o. Lastly, we follow the same procedure and train LSTM on V_3 to get a testing performance, as well as Δ_{3o} which is the difference between the testing performances of LSTM trained on V_3 and V_o. We define a desired improvement to be

$$\Delta_{3o} > \Delta_{1o} + \Delta_{2o}$$

In other words, if the above inequality is satisfied, then we conclude the bidirectionality of preprocessed data is indeed beneficial for testing performance.

5 Results and Analysis

5.1 Results

The results of the three baselines are presented in Table 2. The testing accuracy of models trained on V_1 dataset is shown in Table 3. The testing accuracy of models trained on V_2 dataset is shown in Table 4. The testing accuracy of models trained on V_3 dataset is shown in Table 5.

Dataset	Spam	CoLA	COVID
Accuracy	0.982	0.700	0.629

Table 2: Baseline results of plain accuracy

Dataset	Spam V_1	CoLA V_1	COVID V_1
Accuracy	0.982	0.702	0.637

Table 3: Testing Accuracy of LSTM trained on V_1 of three original datasets

Dataset	Spam V_2	CoLA V_2	COVID V_2
Accuracy	0.983	0.705	0.631

Table 4: Testing Accuracy of LSTM trained on V_2 of three original datasets

Dataset	Spam V_3	CoLA V_3	COVID V_3
Accuracy	0.990	0.702	0.670

Table 5: Testing Accuracy of LSTM trained on V_3 of three original datasets

From the above tables, we can calculate the corresponding Δ's. The results are shown in Table 6

	Spam	CoLA	COVID
$\Delta_1 o$	0	0.002	0.008
$\Delta_2 o$	0.001	0.005	0.002
$\Delta_3 o$	0.008	0.002	0.041

Table 6: Δ's for each of the three datasets

Notice that for Spam Classification dataset and Covid dataset, $\Delta_3 o$ is larger than the sum of $\Delta_1 o$ and $\Delta_2 o$. Therefore, we conclude for these two datasets, our bidirectional pre-processing technique indeed improves the performance of LSTM model. However, for CoLA dataset, there is no evidence of improvement according to our definition.

5.2 Analysis

First, it is reasonable that bidirectional preprocessing does not help with the performance of LSTM trained on CoLA. Since CoLA's label determines the grammatical acceptability of a sentence, reversing the order of words would sabotage the sentence structure and hence the grammar. Therefore, grammatically speaking, V_3 contains a new type of language that does not share the same set of syntax with ordinary English. In other words, the model is performing grammatical classification on a language that is not English, which would produce substantial variance because the labels used are still for English. Therefore, it is logical that the performance of LSTM is not improved with bidirectional inputs in V_3.

Second, for Covid and Spam dataset, we examine more closely and explicitly what kind of text would cause bidirectional input to have advantage. We take out the following example from the Spam dataset, which is classified correctly by LSTM trained on V_3 but incorrectly by LSTM trained on V_o.

> *"Yay! Finally lol. I missed our cinema trip last week :-("*

This sentence has true label "ham", i.e., it is not a spam email. However, the LSTM which is trained on V_o classifies it as spam. Our inference is that, on one hand, it contains "!", "finally", "miss", "cinema", and "last" which are all common words that appear in spam advertising emails. More importantly, when these words are fed into the model as embeddings, the model does not have information of words that come after them. For instance, when the model sees "cinema", it is not aware of the next word "trip", which is a dependency of "cinema". This causes the cinema to process with only "cinema" itself as well as the words before it, such as "finally". This may mislead the model to learn this text as a "cinema membership discount advertising" email, or other connotation that is common in a spam email. However, when the model is trained on V_3, it has the information from both directions. Then, seeing "trip" together with "cinema" would tune the model's understanding towards a personal email. The significance of bidirectionality is shown in this manner.

6 Conclusion

In this study, we have shown that the single direction forward LSTM achieves a 1% improvement in the spam classification task and a 3.9% improvement in the COVID sentiment analysis dataset. The drop in model performance in the CoLA dataset can be explained by the inherent nature of the CoLA dataset as grammatical acceptability. We have also shown qualitatively how a single sentence can be classified correctly given bidirectional inputs instead of the unidirectional inputs. We conclude that bidirectional information is a contributing factor to the improved classification performance of the forward LSTM.

Despite of the improvement in classification accuracy due to bidirectional information, it is unclear how bidirectional information helps model performance. Potential factors could include absolute position of certain tokens, the relative position of negation operator, or assignment of named entities. We leave it to future work for investigating the mechanism bidirectional information via behavior testing by Checklist (Ribeiro et al., 2020). Checklist is a comprehensive software that tests different aspects of natural language understanding like semantic role labeling, negation, name entity recognition, logic etc (Ribeiro et al., 2020). By testing on different aspects of natural language understanding for LSTM trained on undirectional and bidirectional inputs, we can decompose the misclassification rates into axes of linguistic abilities not captured by models with unidirectional input. With more varied dataset choices covering grammar, sentiment, natural language inference in the future, we can extend our analysis to state-of-the-art Transformer models like BERT to further interrogate and compare the effectiveness of bidirectional inputs and encoding in model performance.

7 Github

All code files for this current project can be found here:
https://github.com/wangluheng328/BA-Significance

References

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2018. Supervised learning of universal sentence representations from natural language inference data.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Association for Computational Linguistics (ACL)*.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

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 *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.