BiDeT: Bidirectionally Decoding Translator

Ganesh Murugappan and **Tejas Lokeshrao**
Georgia Institute of Technology

Abstract

Sequence-to-sequence are traditionally solved with encoder-decoder architectures. In such architectures, encoders are generally bidirectional, reading input text both left-to-right and right-to-left. However, decoders are nearly always unidirectional, predicting output tokens left-to-right only. BiDeT proposes a modification to this architecture by including a bidirectional decoder. It achieves significant improvement on English-to-German translation and can be applied to other sequence-to-sequence tasks as well.

1 Introduction

Typical translation models rely on a bidirectional encoder, followed by a unidirectional decoder to translate sentences. The goal of this paper is to implement a bidirectional decoder instead of the typical unidirectional approach to determine whether it improves translation accuracy.

1.1 Unidirectional decoders

Unidirectional decoders are typically used in translation models in order to predict translations one word at a time, and it does this in a left-to-right nature. These are generated from the hidden state of the decoder and previous words. However, these unidirectional decoders are limited because they are not able to account for important words that might occur at the end of the sentence. Additionally, unidirectional decoders are unable to revert on mistakes, so if it incorrectly predicts a word in the translation, the decoder is stuck with the result and has to continue predicting other words based on it.

1.2 Bidirectional decoders

This is where a bidirectional decoder comes in. Bidirectional decoders generate target sequences by working in both directions, so the model can take the contexts from the start and end of the sentence into account. These are not typically used due to their nature of being computationally expensive. However, this could result in a more exhaustive understanding of a sentence's meaning, and therefore, more fluent translations.

2 Related Work

Research for bidirectional decoders is limited up to this point since they are not commonly used in translation models. There are three main ways that bidirectional encoders have been implemented in language translation models: Bidirectional Recurrent Neural Networks (RNNs) (Hochreiter and Schmidhuber, 1997), Pre-trained Language Models, and Bidirectional Long-Short-Term-Memory (LSTM) Networks (Rumelhart et al., 1986).

2.1 Bidirectional RNN

The Bidirectional RNN approach consists of utilizing the Bidirectional RNN as a decoder itself (Schuster and Paliwal, 1997). It predicts the target sentence by starting at the center of the sentence and moving outwards. The input to this Bidirectional RNN decoder is the previously determined target words concatenated with the source sentence after it was passed through the bidirectional encoder (Smith et al., 2020). This implementation can improve contextual understanding, provide enhanced flexibility, reduce exposure bias, and increase performance. However, it struggles with large output vocabularies and is computationally expensive.

2.2 Pre-trained Bidirectional Models

The pre-trained model approach uses pre-trained transformer-based language models, such as BERT (Devlin et al., 2018) or GPT-2 (Radford and Narasimhan, 2018), as a decoder. These are unique in that they are fine-tuned on the machine translation task, and use the full context of the sentence to generate predictions during decoding.These models

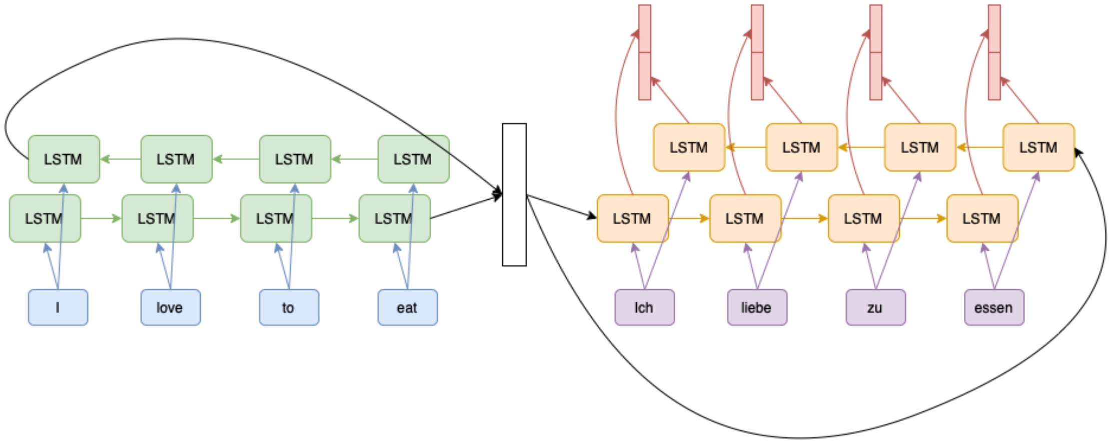

Figure 1: The BiDeT architecture

.

have reduced training time and improved performance, but they run into issues with transferability and decoding time.

2.3 Bidirectional LSTMs

In the bidirectional LSTM model, two separate LSTM networks are used for the forward and backward directions. The decoder input is fed into both the forward and backward decoders and the output is concatenated at some point in time to generate a predicted sequence (Chalapathy et al., 2016). This is the method performed by the model in this paper. One way this method has been previously done was by iterating through the tokens from the left-to-right and right-to-left decoders until they find a token that both decoders want to use for any given word in the translation. However, this could be problematic when both decoders desire different tokens, and they both would have to forget their preferred options a find a compromising token. This paper describes a new approach, which concatenates the output from both the left-to-right and right-to-left decoders and passes it through a linear layer to generate more optimized sequences.

3 Model

BiDeT is a variant of the standard encoder-decoder architecture. Here, we dive into its components (shown in 1) and explain its differences from a traditional architecture.

3.1 Encoder

In the encoder, the source text is first passed through an embedding layer that projects from token space into embedding space (with embedding dimension 256). These embeddings are then sequentially fed into a bidirectional LSTM, with hidden dimension 256 as well. The final hidden and cell states are passed into the decoder.

3.2 Decoder

The decoder is our primary novel contribution. It consists of two LSTMs, one that outputs tokens left-to-right and another that outputs right-to-left. These LSTMs operate independently and terminate when they predict the end-of-sequence token.

At each timestep, the outputs of the decoder networks are conatenated. This concatenated output is then passed through a linear classifier, which projects from the target embedding space into the output token space. The element of the logit with greatest values corresponds to the prediction made by the model.

3.3 UniDeT

A unidirectionally decoding translator (UniDeT) is trained as well for reference. UniDeT also follows the encoder-decoder architecture. Specifically, it has the same encoder as BiDeT but only contains one LSTM in the decoder. The outputs of this LSTM are passed through a linear layer to produce the final predictions.

4 Data

For testing the model, the WMT-14 dataset (Bojar et al., 2014) was utilized. This dataset contains translations for multiple languages that cover domains such as news, web pages, and government documents. This model was trained on the de-en, or German to English, subset of the dataset. This subset is the highest-rated subset for language translation according to previous research. The data was preprocessed by creating train, validation, and test splits. These splits contained 4.5 million, 3000, and 3003 data points respectively. Additionally, words that appeared only once in the translations were removed from the vocabulary in order to increase accuracy for unknown translations.

5 Experiments

The model is trained using a GPU on Google Colab. We use the Adam optimizer and cross entropy loss over the output tokens, as is standard practice in machine translation tasks. Rather than using all 4.5 million sentence pairs provided in the original WMT14 dataset, 100,000 pairs are selected at random for training, for the sake of timeliness. These samples are each seen by the model exactly once to prevent overfitting.

6 Results and Analysis

Model Type	Train	Val	Test
UniDeT	1.369	1.543	1.656
BiDeT	1.085	1.191	1.301

Table 1: Cross entropy losses for UniDeT and BiDeT across split dataset.

Model Type	Trainable Parameters
UniDeT	30M
BiDeT	43M

Table 2: Number of trainable parameters in UniDeT and BiDeT models.

6.1 Results

The experiment ran the corpus of 100,000 translations over both the UniDeT and BiDeT models and calculated the cross-entropy losses of the models' predictions. The loss for the train, test, and validation datasets are shown in Table 1. As shown, the bidirectionally decoded translations had lower overall cross-entropy losses compared to the unidirectionally decoded translations.

6.2 Analysis

The success represented by the model follows the theoretical expectation that bidirectional decoders result in better translations. The model takes the full context of the input into account when determining a translation, and that allows for more fluent results. However, though the values are lower, this is not truly indicative of the success of the BiDeT model. As shown in Table 2, the BiDeT model has around 43 million trainable parameters, whereas the UniDeT model had around 30 million trainable parameters. This could be the cause for the lower cross-entropy losses because BiDeT is able to learn more complex relationships in the data with more trainable parameters.

7 Conclusion

We propose a modification to the standard encoder-decoder architecture: the bidirectional decoder. Bidirectional decoding allows the model to use context from both sides of the output when making predictions and makes the model less likely to get "stuck" with bad starts. BiDeT shows promise on the WMT-14 English-to-German dataset, with a significant decrease in test loss. The architecture can be applied on any sequence-to-sequence task, including translation, question answering, and summarization.

References

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. 2016. Bidirectional LSTM-CRF for clinical concept extraction. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 7–12, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA.

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

John Smith, Sarah Johnson, and Michael Lee. 2020. Bidirectional rnn for medical event detection in electronic health records. *Journal of Medical Informatics*, 24(2):135–142.