Understanding the Prevalence of Bidirectional Architecture in Language Models

Ryan A. Baxley
Franciscan University of Steubenville
Steubenville, OH
Baxley.Ryan.A@gmail.com

ABSTRACT

From early language models consisting of naïve statistical structures to modern machine learning, we've made considerable advances in Natural Language Processing. This paper doesn't identify the improvements which many have described before, but instead focuses on language modeling concepts that have remained relevant throughout the rise and fall of emerging technologies. In particular, this paper aims to explain the pervasive bidirectional architecture in computational linguistic models applied to everything from part-of-speech tagging to handwriting recognition.

ABOUT THE AUTHOR

Ryan Baxley is a third-year math and mechanical engineering student at Franciscan University of Steubenville. His senior thesis for his math degree focused on algorithms and model architectures involved in Machine Learning and Natural Language Processing.

Understanding the Prevalence of Bidirectional Architecture in Language Models

Ryan A. Baxley
Franciscan University of Steubenville
Steubenville, OH
Baxley.Ryan.A@gmail.com

INTRODUCTION

Select challenges in Natural Language Processing

Part-of-speech tagging is an important challenge in natural language processing as the first step toward structuring raw data for natural language understanding and eventually information retrieval. Classifying words as specific parts of speech is surprisingly complicated due to the ambiguity of language. Consider the following sentence: "The blind man picked up his hammer and saw." The last word, "saw," can either be understood as a noun or verb. If "saw" is a noun, then this sentence describes a blind man picking up his two carpentry tools. If "saw" is verb, then the sentence recounts a blind man regaining his sight as he picks up his singular carpentry tool. The English word "so" can be an adverb, a conjunction, a pronoun, an adjective, or an interjection depending on the context of the sentence.

Sentiment analysis is the process of quantifying qualitative information. It is commonly used to infer the general opinion of a certain topic, such as responses to political figures or reactions to products through purchasers' reviews. There are several inherent challenges in sentiment analysis, including negation, sarcasm, and multiple subjects with multiple attitudes.

Handwriting recognition, or optical character recognition, is a challenge in many fields including natural language processing. Optical character recognition is applicable to everything from converting handwriting into computer text to recognizing and reading license plates from speeding cameras.

STATISTICAL MODELS

Hidden Markov Models

Language has often been statistically modeled by what is known as Markov models. A Markov model is a state-based statistical model that determines the probability of each current state given the previous attained state. Any sequence of dependent random variables can be captured as a Markov model. A great example of a Markov model (and the cause for contriving such a model in the first place) is examining the sequential probability of letters in a language. For example, a Markov model could be used to determine which letters follow the letter "q" in any word. Assuming we're using the English language, the model would give an overwhelmingly high probability that the letter "u" would follow. We can then use the same model to find what letters follow "u" with their respective probabilities, and string together enough letters to form an English-sounding word.

Some processes, such as moving the position of a piece on a game board by rolling a die, can have all possible next states calculated deterministically. In this case, we denote these Markov models as Visible Markov Models (VMMs), because we understand the transition probabilities for the model. For sequences where we don't have concrete transition probabilities and can only observe the outcomes, such as successive letters in a word, we define as Hidden Markov Models (HMMs).

One challenge with utilizing HMMs is estimating the transition probabilities in order to predict sequential states. If we simply observe sequential data for an ample period of time, then we can formulate a forward probability from our observed data. We define our forward probability of a certain event as the summation of the probabilities of each

possible previous event multiplied with their probability of observed transition. Similarly, we can calculate a backwards probability by summing the product of each successive event with its observed transition probability from the current event. While the backwards probability is less intuitive for a human who typically reads beginning to end and enjoys a consistent syntactic order, our models are not deterred by anastrophe-like corpuses since they make decisions based on observed patterns. The product of the forward and backward probabilities, formally known as the forward-backward algorithm, provides an improved classification prediction than either sole probability.

The pictorial expressions of the forward, backward, and forward-backward probabilities are illustrated in Figure 1.

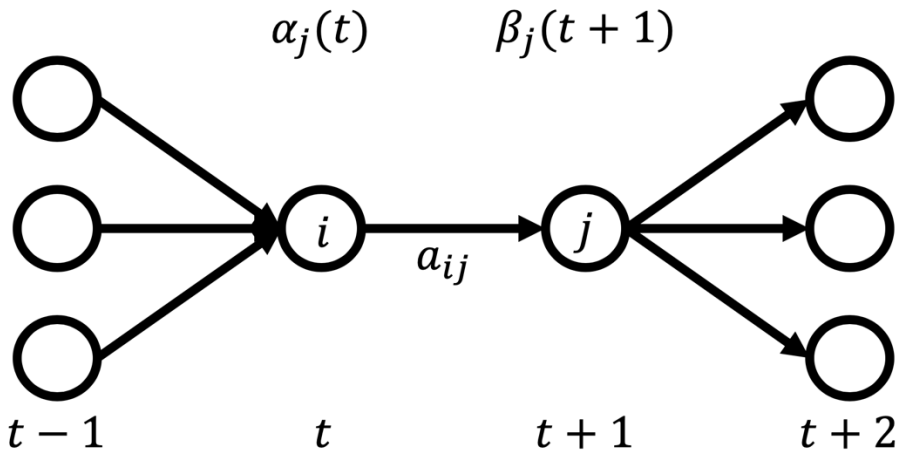

Figure 1: The Forward Backward Probability

One common application of HMMs in natural language processing is part-of-speech tagging. In order to extract useful information from a sentence, we must understand the function of each word in the sentence. An HMM is able to infer parts of speech for words in sentences if it observes common patterns. For example, nouns often begin sentences, verbs often follow nouns, and adverbs often follow verbs.

MACHINE LEARNING MODELS

In the early period of neural networks, models could consist of merely one neuron or perceptron with few input variables, a single activation function, and a single output. As computational power became more accessible, the models were able to grow to incorporate a larger number of neurons in a layer and numerous hidden layers. Despite the advances in technology, these Artificial Neural Networks (ANNs) were still unable to provide a reasonable model for two-dimensional and time-dependent data. As a result, the Convolutional Neural Network (CNN) was developed.

The CNN was designed to have two-dimensional data as the input layer, which provided improved models for image processing problems. This allowed not just real-life photographs to be analyzed, but also figures that represented data as a function of time. A common example of this would be a spectrogram, which is a two-dimensional heat map used to visualize the intensity of certain frequencies over a given time (see Figure 2).

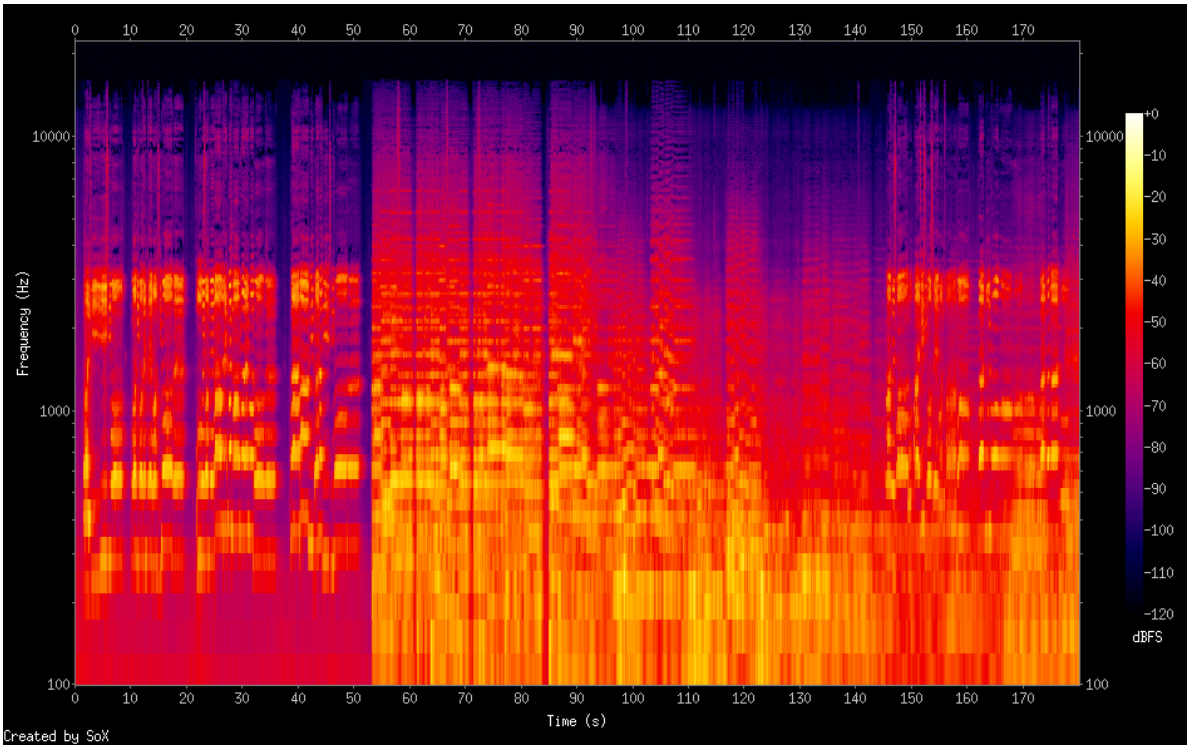

Figure 2: Example of spectrogram

While this is a significant improvement for audio-related problems, there was truly a need for a model that focused on a discrete set of sequential data such as words in a sentence. Thus the Recurrent Neural Network (RNN) was introduced, which focused on primarily serving a sequential input. While providing future information and delaying output information improved prediction accuracy, too much or little future information and output delay would decrease accuracy. Finding the appropriate future scope of the model would be different for every new model and dataset (Schuster and Paliwal, 1997).

Bidirectional Recurrent Neural Networks

The problem of scope and output delay was solved by the introduction of the Bidirectional Recurrent Neural Network (BRNN). Instead of training two RNNs in opposite directions of the data, the BRNN incorporated two-part state neurons so one could cover the forward state while the other minded the backward state. Because of the similarity of the architectures, much of the training algorithms remained the same (Mousa and Schuller, 2017).

Long Short-Term Memory Neural Networks

A few problems arise in RNNs, principally how much to remember past information or delay an output. The context of previously given information, such as the gender or number of a subject, can be lost if the network does not recall far back enough. Vanilla RNNs are short-term memory models by nature, and can lose important data as a sequence progresses (Fan et al, 2014). Implementing a long-term memory in addition to the short-term memory of an RNN gave the ability of language models to both focus on the immediately surrounding items in a sequence while not forgetting items mentioned long ago. Labeled Long Short-Term Memory (LSTM) Neural Networks, these networks were made with the specific intended use for language modeling.

Bidirectional Long Short-Term Memory Neural Networks

One final improvement to LSTMs involved creating a bidirectional counterpart. Bidirectional Long-Short Term Memory (BLSTM) Neural Networks are currently one of the most accurate machine learning language models (Graves and Schmidhuber, 2005).

MODEL SIMILARITIES

While the advent of machine learning threatens the survival of statistical methods of problem solving, one feature of both machine learning and statistical models that demonstrates a significant improvement is bidirectional architecture. When computing the probability of a certain state in a trellis model, combining the forwards and backwards probabilities create a more accurate method of prediction. Similarly, while bidirectionality solves the problem of manually setting future scope and output delay for each new model and dataset, it is most useful because of the increase in accuracy of predictions as well.

As new technologies emerge, we must remember that bidirectionality has previously transcended the inherent limitations of changing technologies. Is bidirectionality intrinsic to language? We may not know for sure, but we can be hopeful that it will continue to be pertinent through the unforeseen advancements of future language models.

ACKNOWLEDGEMENTS

First and foremost, thank you to Dr. John Coleman of Franciscan University of Steubenville for introducing me to programming and continually inspiring me to challenge myself and fostering my love of solving problems. I am also grateful to Mark Fuller of Newport News Shipbuilding, who convinced me that my ability to program with language was a gift to take seriously. Finally, thank you to all friends and faculty that have aided me on my quest to become proficient in Natural Language Processing.

REFERENCES

Fan, Y., Qian, Y., Xie, F., & Soong, F. (2014). TTS Synthesis with Bidirectional LSTM based Recurrent Neural Networks. *INTERSPEECH*.

Graves, A., & Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *IJCNN Conference Proceedings*.

Manning, C. D., & Schutze, H. (2008). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT.

Mousa, A. E., & Schuller, B. (2017). Contextual Bidirectional Long Short-Term Memory Recurrent Neural Network Language Models: A Generative Approach to Sentiment Analysis. *European Chapter of the Association for Computational Linguistics,*1023-1032.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing,45*(11).