Transformer-Based Voice-Controlled Check-ins at Medical Facilities

RAYAN S. WALI
Department of Computer Science, Cornell University
Ithaca, New York, United States 14850.

The advent of artificial intelligence, especially in the last decade, from the development of self-driving cars to robots that can smoothly interact with their environment, has spurred further research to be performed in areas that are in critical need for automation. One such area is the healthcare field, where medical-imaging robots are being used in limited medical facilities to conduct special operations that involve high precision that we, as humans, are not capable of doing. As doctors are at risk of contracting diseases from patients, we discuss the possibility of eventual widespread voice-controlled check-in systems taking over front-desk workers as well as robots taking over medical operations in this research paper. In particular, the medical operations we will study is lab testing automation. Furthermore, we will dive into sentiment analysis and opinion mining on the test result report comments a patient receives from their doctor once the test has been conducted by the robot. We will examine the components of artificial intelligence that are needed to make this possible, from speech and textual recognition to facial recognition implementations.

Keywords: Speech Recognition, Text Mining, Sequence Models, Transformers & Attention-Based Mechanisms

1. Introduction

A purely automated voice-controlled check-in system would first include a security measure to verify one's identity. Typically, when checking-in to a hospital, one has to show some form of an ID card to verify their identity. This could be automated with a sequence of steps that begins with facial recognition. The person who would like to be identified would need to be searched through an extensive database containing information regarding all members registered under health-care providers. This matching process could be performed with a pairwise feature correspondence checker, which takes in two images as inputs and outputs a similarity score indicating the difference in their feature descriptors. Once the individual is admitted into the hospital, they are ready to receive treatment. A robot with a pre-defined algorithm designated for performing the task the patient is looking for will then be chosen. Each one of these processes will be described in detail in the sections that follow, with emphasis placed on attention-based recurrent neural networks (RNNs) compared to the conventional Long Short-Term Memory (LSTM) model, once speech-to-text translation has been performed. The robot can be controlled by medical staff with voice commands, which are then converted to text, and finally its semantics can be extracted. With regard to automatic speech recognition (ASR), raw speech is first taken in as a set of sound waves, each carrying its own frequency and amplitude. However, before analysis, as in the standard machine learning pipeline, preprocessing must be applied to the sound wave to remove noisy components. An attention-based model, as opposed to a standard artificial neural network with a feed-forward and a back-propagation stage, will remember information from the past, and use that collective set of information to predict the future. Before delving into our healthcare application, we will give a brief tutorial on attention-based machine learning models.

2. Terminology Used throughout this Paper

We define the natural-language-processing jargon used throughout this paper. The word *corpus* is used to mean the set of language that is fed into a machine learning model as a part of the training process. A corpus consists of sequences, which are then *tokenized* — split up into tokens, which can either be words within the sequence, or even sub-sentences. Some Python libraries we will go into in the later parts of the paper are SPACY and NLTK, with NLTK primarily used to tokenize a sequence and SPACY used to extract useful information from the sequence.

3. Applications of AI

3.1 *Data Collection*

Living in an Internet-of-Things (IoT) world implies that we are dealing with a widespread transfer and collection of data. Data may be available as either in its raw form (images, speech recordings) or in a cleaned form (structured datasets/databases). The former is known as unstructured data and the latter is known as structured data. Without sensors collecting user data, we would have been unable to improve newly-built frameworks with the power of machine learning.

3.2 *Deep Learning*

Deep learning is the driving force behind the applications of artificial intelligence listed below. It is a sub-field of artificial intelligence that focuses on deep (multi-layered) neural networks. Such models tend to work well when a lot of training data is gathered and is available. However, deep neural networks are prone to overfit due to their complex structure, meaning that they must be trained with caution, especially when working with limited data.

3.3 *Handwritten Text Recognition*

Handwritten Text Recognition is part of Natural Language Processing (NLP), another sub-field of artificial intelligence. Given a stream of words of characters, it strives to extract useful information from it.

3.4 *Robotics*

Robotics plays a big part in our current semi-automated society. Artificial intelligence is embedded into robots, as they are able to learn from their mistakes. As they gather more data, they learn the surrounding environment, and thus, are able to perform the task they are given more accurately. Some common examples of robotics are self-driving cars, robotic vacuums, and medical imaging robots.

3.5 *Facial Recognition*

Facial recognition is a common artificial intelligence application that strives to classify a given face, whether it is the person's emotions or their gender. Facial recognition softwares are used by law enforcement agencies to identify who is the suspect and who is the victim. At the same time, there are circulating privacy/ethical concerns, as sensors may collect data without the permission of those whose data is collected. And, in certain cases, this can potentially worsen in the case of misclassifications, or Type I/II errors (false positives/negatives). For example, a face classification algorithm could identify an innocent person as guilty, and this is partly a reason for why complete automation is not something that

is agreed upon by everyone. Such privacy concerns are not only prevalent in facial recognition systems, but rather, are prevalent in other artificial intelligence applications, including the ones that we examined above.

FIG. 1. Facial Recognition Software: Pre (left) and Post (right)

4. Introduction to Neural Networks

Deep learning is the training of deep neural networks. Training deep learning models consists of two stages: (1) the feed-forward stage and (2) the back-propagation stage. The feed-forward stage simulates a forward pass through the network by taking as input the features of a particular training example and outputting a value or a set of values representing the prediction or predictions for that example.

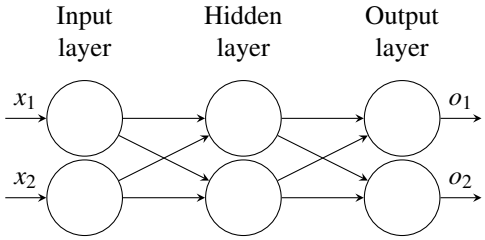

FIG. 2. Fully-Connected Neural Network with 1 hidden layer

4.1 *Activation Functions*

An activation function, denoted by $\sigma(\cdot)$, is a function applied to a value that returns another value; it is essentially a transformation of the inputted value. Some common activation functions are ReLU and sigmoid, which are defined as $\text{ReLU}(x) = max(0,x)$ and $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$ respectively.

4.2 *Weights of the Network*

Each edge connecting two nodes in the network has a weight, defined by $w_{a,b}$, where a is the incoming edge node and b is the outgoing edge node. For example, the weight connecting the top node of the input layer to the bottom node of the hidden layer is represented by $w_{1,2}$.

4.3 *The Forward Pass*

In Figure 2, we call the output of the node of the hidden layer on the top h_1, and the output of the node of the hidden layer on the bottom h_2. Then, the following equations must satisfy for activation function σ:

$$h_1 = \sigma(w_{1,1}x_1 + w_{2,1}x_2) \tag{4.1}$$

$$h_2 = \sigma(w_{1,2}x_1 + w_{2,2}x_2) \tag{4.2}$$

4.4 *The Backward Pass (Batch GD)*

The backward pass through the neural network updates the weights and biases of the network based on the cost function derived from the forward pass through the network. This can be performed by a method known as *gradient descent*, which moves the weights and biases in a direction to optimize (minimize) the cost function. Gradient descent is performed by using the following update rules:

$$w^{(t+1)} \leftarrow w^{(t)} - \alpha * \nabla J(\theta) \tag{4.3}$$

$$b^{(t+1)} \leftarrow b^{(t)} - \alpha * \nabla J(\theta) \tag{4.4}$$

In the equations above, $J(\theta)$ represents the cost function and α is a hyperparameter that represents the learning rate. For m examples, $J(\theta)$ is defined as follows:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathscr{L}\left(h_\theta(x_i), y_i\right) \tag{4.5}$$

In each timestep, the equation above is used to update the parameters of the network, until the weights and biases minimize the cost function. In the end, gradient descent satisfies the following optimization problem:

$$w^*, b^* = \underset{w,b}{\arg\min}\ J(\theta) \tag{4.6}$$

4.5 *Forms of Gradient Descent*

There are two main types of gradient descent: (1) batch and (2) stochastic. The difference between the two is that the batch form updates the parameters of the network after one epoch (the forward pass of all the examples), whereas the stochastic form updates the parameters of the network after the forward pass of each training example.

5. Overview of Attention-Based Machine Learning Models

An attention-based model is a supervised machine learning model that is an improvement over the encoder-decoder-based model used for machine translation. The architecture of the attention-based model consists of encoder and decoder stacks. The role of the encoder is to map input sequences to sequences of continuous representations. From these continuous representations, the decoder then uses them to generate a set of outputs. One key improvement of the transformer model is performance — the encoder-decoder models blow up in time complexity as the length of the inputted sequence increases. A schematic of the transformer model, a form of the attention-based model, is presented in Figure 3 to the right:

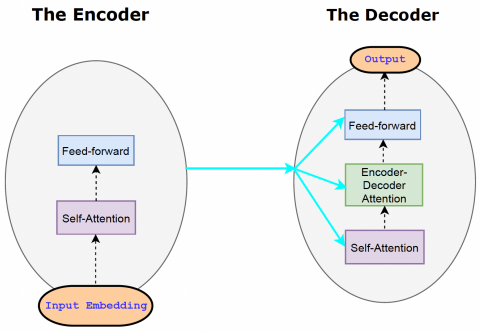

FIG 3. The Attention-Based Transformer Model