IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019) ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)

Voice Assisted Web Design by Means of Deep Learning

Mrs. Sunitha Patel M S[1], Mr. Sanjay S[2], Ms. Zoha Afreen[3], Ms. Thanushree[4], Ms. Varshitha M[5]
[1]*Assistant Professor*
Dept of CSE, ATME College of Engineering, Mysuru.

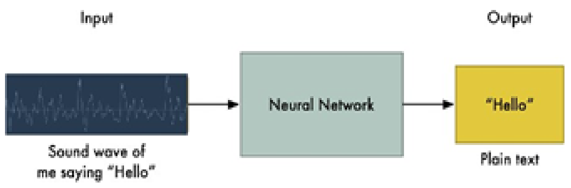
Abstract - This paper presents a system to automate the process of web design by taking in voice input. There are four phases to this project. First being data-processing, which is converting analog signals to digital which can be further processed. Second phase is based on a combination of the deep bidirectional LSTM recurrent neural network architecture and the Connectionist Temporal Classification objective function. Third phase is concerned with extracting key words which are known beforehand.
The Fourth and the final phase is to output a CSS file in a proper syntactical manner.

I. INTRODUCTION

Automation is evolving quickly and business intelligence in applications is a new form of high-quality automation. In the technology domain, the impact of automation is increasing rapidly, both in the software/hardware and machine layer.
One field that has not seen much automation even though technology has advanced is web development. People have to sit down and type in all the code by themselves which is a hectic process. For that matter the task of programming itself can be automated to some degree. The idea is that we can just speak to the computer and it types in for itself.
The process of typing itself is automated, instead of typing for hours and hours we can just speak and the computer does all the hard work. This greatly improves speed at which we can develop programs.
This paper aims at doing just this task as mentioned above. The program recognizes the user's voice input, process it and convert it to text format. This text file is further processed to build a basic CSS file with few commonly used selectors and properties such as background-color, font-size, text-align, etc.

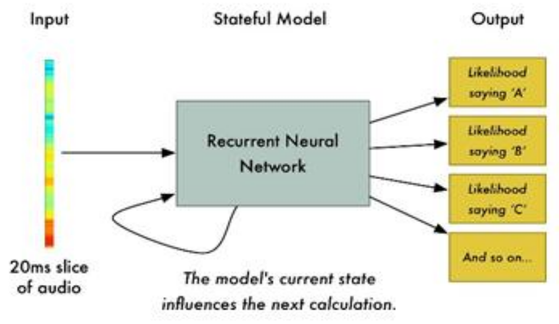
II. PYAUDIO

PortAudio is a free, cross-platform, open- source, audio I/O library. It lets you write simple audio programs in 'C' or C++ that will compile and run on many platforms including Windows, Macintosh OS X, and Unix (OSS/ALSA). It is intended to promote the exchange of audio software between developers on different platforms. This is a famous C library to deal with audio.
PyAudio provides bindings for PortAudio which allows to play or record audio. This library is open-source and cross-platform.
PyAudio has inbuilt functions which allow us to read in data from microphone or write to a speaker.

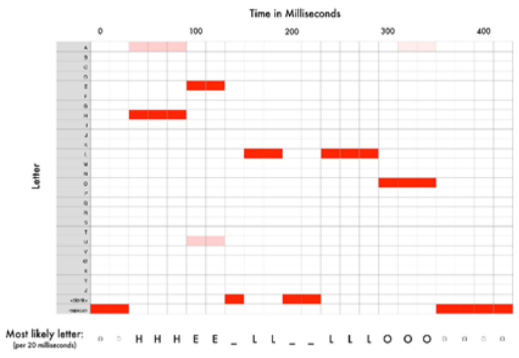


III. RECURRENT NEURAL NETWORK

Recurrent Neural Networks are the state-of- the-art algorithm for sequential data and among others used by Apples Siri and Googles Voice Search.
Recurrent nets are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, or numerical times series data emanating from sensors, stock markets and government agencies. These algorithms take time and sequence into account, they have a temporal dimension.



Now that we have our audio in a format that's easy to process, we will feed it into a deep neural network. The input to the neural network will be 20 millisecond audio chunks. For each little audio slice, it will try to figure out the letter that corresponds to the sound currently being spoken.
We'll use a recurrent neural network—that is, a neural network that has a memory that influences future predictions. That's because each letter it predicts should affect the likelihood of the next letter it will predict too. For example, if we have said "HEL" so far, it's very likely we will say "LO" next to finish out the word "Hello". It's much less likely that we will say something unpronounceable next like "XYZ". So, having that memory of previous predictions helps the neural network make more accurate predictions going forward.
After we run our entire audio clip through the neural network (one chunk at a time), we'll end up with a mapping of each audio chunk to the letters most likely spoken during that chunk. Here's what that mapping looks like for me saying "Hello":



IV. CONNECTIONIST TEMPORAL CLASSIFICATION

Connectionist Temporal Classification (CTC) is a way to get around not knowing the alignment between the input and the output.

Neural networks (whether feed forward or recurrent) are typically trained as frame- level classifiers in speech recognition. This requires a separate training target for every frame, which in turn requires the alignment between the audio and transcription sequences to be determined by the HMM. However, the alignment is only reliable once the classifier is trained, leading to a circular dependency between segmentation and recognition (known as Sayre's paradox in the closely-related field of handwriting recognition). Furthermore, the alignments are irrelevant to most speech recognition tasks, where only the word-level transcriptions matter. Connectionist Temporal Classification (CTC) (Graves, 2012, Chapter 7) is an objective function that allows an RNN to be trained for sequence transcription tasks without requiring any prior alignment between the input and target sequences.

V. RAKE

Rapid Automatic Keyword Extraction (RAKE) is an algorithm to automatically extract keywords from documents. Keywords are sequences of one or more words that, together, provide a compact representation of content (see reference below). RAKE is a well-known and widely used NLP technique, but its concrete application depends a lot on factors like the language in which the content is written, the domain of the content and the purpose of the keywords.

The implementation in this library is mainly aimed at English. With additional resources, it is also applicable to other languages.

VI. CONCLUSION

In this paper we show a possibility of using neural network architecture to automate the process of typing. Typing can be replaced by just speaking to the computer and can save a lot of time.

Once this system is complete, it can serve as a platform for various other applications.

VII. REFERENCES

[1]. Alex Graves, Navdeep Jaitly. Towards End-to-End Speech Recognition with Recurrent Neural Networks, 2014.

[2]. Alex Graves, Santiago Fern´andez, Faustino Gomez, Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Un-segmented Sequence Data with Recurrent Neural Networks, 2006.

[3]. Dr. R.L.K.Venkateswarlu, Dr. R. Vasantha Kumari, G.Vani JayaSri. Speech Recognition By Using Recurrent Neural Networks, 2011.

[4]. Ambar Dutta, A Novel Extension for Automatic Keyword Extraction, 2016.