A deep learning approach for gesture to text in Augmented Reality user interface

Junxiao Shen[a,b]

[a]1Department of Engineering, University of Cambridge, UK
[b]js2283@cam.ac.uk

Keyword: Human-Computer Interaction, Deep Learning, Machine Learning, Gesture Text Input, Augmented Reality

Abstract: Augmented Reality (AR) has attracted much popularity across the industry, however, due to the expensive hardware and unintuitive user interface. It has not been as popular as mobile phones among normal users. A lot of research has been put into the development of a plausible user interface in AR and one of the most important areas in the text input.

1. Introduction

Text input is the most effective way to interact with computer systems so far and different methods are used in different devices. For example, QWERTY keyboard has been the most popular text input keyboard in personal computers whereas T9 predictive text was originally invented for users in the mobile phones considering the small user interface, and later on, after the screen gets bigger and smartphones came onto the market, QWERTY keyboard started to become a very effective method for text input in smart mobile phones.

In Augmented Reality, the method for text input will be different since it will be in 3D space. Due to the inaccuracy for fingers to do tap actions, the swipe method is now considered to be a more effective and more intuitive way to input texts in AR. However, in QWERTY keyboard, due to the offset between the actual movement track and the sensed movement track of the fingers, there will be a difference between the distribution of the likelihood of the word given the gesture and the actual word we intended to input.

Therefore, I came up with two ideas s to tackle this problem. One is to use T9 keyboard so to decrease the offset and the other is to use a machine learning method to minimise the offset in the QWERTY keyboard. The advantage to having a T9 keyboard is that it only has 9 blocks so the offset might be small enough compared to the actual scale of the gesture tracking. However, the difficulty is that there is not a good enough gesture model for the T9 keyboard to have a good prediction on the word based on the very limited gestures since there are only nine blocks. Furthermore, it is not a common text board for users now and it is against the intuition of users. The advantage of using QWERTY keyboard is that it has an established mature gesture model together with the language model to give a good posterior distribution over the word. Therefore, considering the pros and cons of two different designs of the text board, we decided to use the QWERTY keyboard.

2. Research Goal

Due to the hardware inability to accurately sense the trace of the figure, there is a big displacement between the trace of the figure tips we see in the user interface in HoloLens and the actual data input of the trace. Therefore, a recognition model to map the trace to word or sentences should be invariant of the starting position of the keyboard. Therefore, Deep learning can be used to train this recognition model. Innovated from speech recognition, since speech is a 1D temporal data and the trace is a temporal and spatial data, therefore, the network models and state-of-art algorithms for speech recognition can be used to innovate us to design and train the neural networks for the trace recognition.

3. Research Plan

To use deep learning, numerous data should put into the neural network. However, it is not possible to collect this amount of gestures. Therefore, we need to synthesis artificial data on our own. Furthermore, to have the right feeling of the gestures, an experiment to study the user habit for gesture input should be used to investigate how the user will input the gestures. Figure 1 shows the overall plan. it is a component diagram and it is used to visualise and help us to have a better idea of the working process.

3.1. First stage

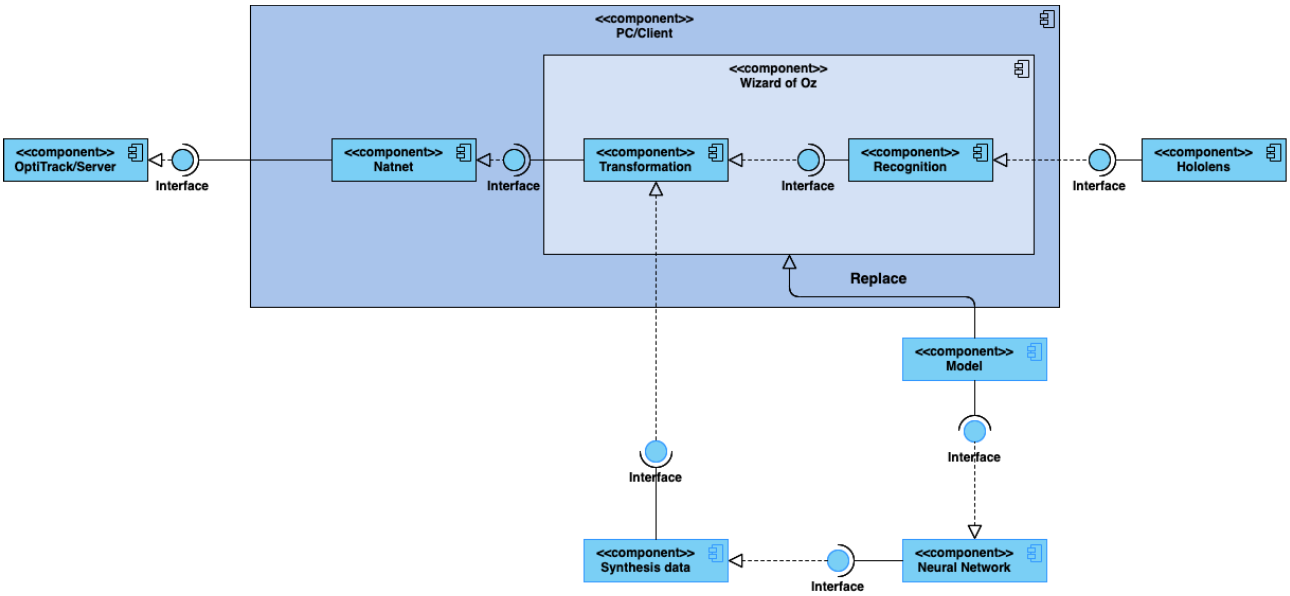
Firstly. the OptiTrack system which is a motion capture system that can accurately track the coordinates of the markers including HMD (Head Mounted Display), fingertips. The reason why we use OptiTrack is that the HoloLens inbuilt sensor is not good due to the huge latency and displacement of the actual figure and the data whereas OptiTrack is the world's most accurate and wide-area VR trackers. This leads to an ultra-low latency and better smooth tracking for HMD and fingertips. Then the data will be transferred to pc by NatNet which is an SDK for streaming motion tracking data across networks. Then the data received on the pc will be transformed to the keyboard frame since the original data is in the world frame. The trace will be compared with the previous setpoints, once it approaches the region of the points, the pc will send a query with a world to the HoloLens and it will output the word for the user.

3.2. Second stage

We will thus use the data we collected from the users' experiment to synthesis the data together with the data for the gaze collected from the experiment since we also believe that the trace of the gaze can be relevant to the final output as well. And the synthesised data will be from used to train the neural network and the trained model will thus replace the recognition component in Figure 1. The reason to synthesis the data is that unlike speech recognition, the recorded labelled data can be accessed quite easily nowadays since people can pay others to read and label whereas in our project. The dataset is limited due to /the lack of funding to hire a lot of people to generate the traces and due to the short time frame considering it is an only one-year project.

3.3. Third Stage

We will do another users experiment of the established system and test the results and compare them with the state-of-art text input system. The user experience will still be carried out in the OptiTrack system.



<<component>> PC/Client
<<component>> Wizard of Oz
<<component>> OptiTrack/Server
Interface
<<component>> Natnet
Interface
<<component>> Transformation
Interface
<<component>> Recognition
Interface
<<component>> Hololens
Replace
<<component>> Model
Interface
Interface
<<component>> Synthesis data
Interface
<<component>> Neural Network

Figure 1: Component diagram for the development

4. detailed development

4.1. User experiment design

We will use the OptiTrack to track the coordinates of the keyboard, figure tips and the HMD in the world frame. Natnnet is used to get the data from OptiTrack which is the server to send the data to the client which is the PC to run the script. A wizard of oz experiment will be designed so to mimic the perfect software for gesture input. Two data kinds will be used in the future to train the neural network, they are the trace of the figure tip and the trace of the gaze both in the keyboard frame. Firstly, a transformation software should be written to transform the coordinates of the HMD and fingertips from world frame to keyboard frame. Secondly, a recognition script should be written to recognise whether the figure tip has hit the region of the previous setpoints if they are within the region in the right order, then a word will be sent to the HoloLens. We will see whether we should use 3D data of the fingertips and gaze striaght away or project the 3D data onto the keyboard to make it 2D data since theoretically the data on the direction that is perpendicular to the keyboard should be independent of the output world but realistically speaking, this might not be the case. This will be explored later.

4.2. Data sthythes

We will generate data based on the data we obtained from the user experiments. There are many methods that i can use to generate the data. There are many existing generative models, one of the most popular ones is Generative Adversarial Networks[1] which use a generator produces training samples from the vector of random noise and use discriminator to classify whether a training sample is real or produced by the generator. However, this model is mainly used in computer vision but this can still be of usefulness to our project. More relevant generative models in Text to Speech can be very important as well such as WaveNet[2]. As can be seen from the workflow diagram in Figure 2, a quality check needs to be conducted to see whether the synthetic data is similar to real data. Different methods can be used such as calculating the KL divergence like TSNE (T-Distributed stochastic neighbour embedding) which learn low-dimensional embedding of feature vector by minimizing KL between similarities in both spaces[3]. Classifier approach can also be used so that if it fails to classify the real data and synthetic data then it is a good indicator.

4.3. Design and Train neural network

Different neural networks can be used. I would start from Recurrent Neural Networks (RNN) first since the trace of fingertips is a 3d spatial and temporal signal and this can be in analogy to speech recognition. Furthermore, we could also try to combine CNN and RNN to see whether it will lead to better results. Transfer learning can also be used since the fine-tuning train ml model with synthetic data in first iterations and the with real data in further iterations. We should also be careful that we shouldn't use synthetic data in a test set since it can have a corresponding pair of real data which may lead to a dangerous result.

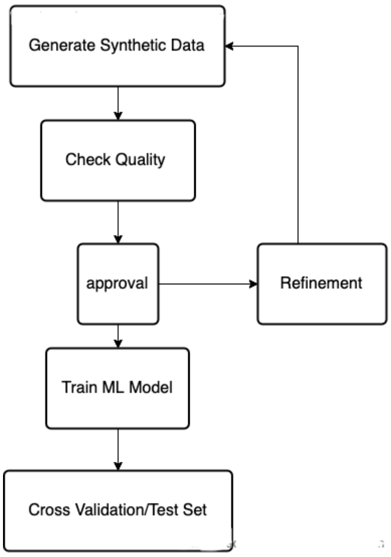


Figure 2: work flow for second and third stage

4.4. Risk Management

There are many risks comes from users, technologies and equipment. However, the biggest challenge and the risk is to generate the data and train the neural network due to several reasons.

- the data is unlike images and speech, the 3D or 2D spatial and temporal data will be not intuitive to us and having a good intuition and understanding of the data will be very important towards the training of the model and designing of the model.
- Mapping 3D or 2D signal to text is a very innovative field so not too much work has been done before and it is not sure if the previous networks for computer vision and speech would be particularly useful.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, arXiv e-prints (2014) arXiv:1406.2661arXiv:1406.2661.

[2] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, arXiv e-prints (2016) arXiv:1609.03499arXiv:1609.03499.

[3] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html

Appendix A.

Good Lucky