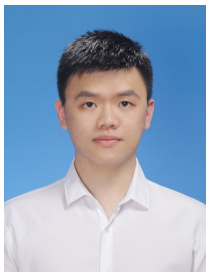
# Peijie Chen


**Email** scorpionsand@163.com
**Address** University of Edinburgh, Edinburgh, UK
**Phone** 7422583873(UK)
15521029020(CHINA)

## Objective

Automatic Speech Recognition

## Education

### MSc Speech and Language Processing

**University of Edinburgh** Edinburgh， UK
2018-2019

1. Executive training courses in speech technology, machine learning, deep learning and signal processing.
2. Upper Second Honours in the first semester

### BA Linguistic

**South China Normal University**
Guangzhou, China
2014-2018

1. GPA 3.8/5.0
2. Published papers about linguistic and literature

## Projects

***Digits recognizer***

1. Building a speaker-independent digit recognizer based on HMM-GMM with HTK.
2. Designing and executing a number of experiments to explore what factors affect the WER, for example the number of states in the HMMs.

***Diphone Speech Synthesiser***

1. Using python to take text input from a user and convert it to a sound waveform containing intelligible speech.
2. A waveform concatenation system, whereby the acoustic units are recordings of diphones.

***Building an uni selection voice in Festival with HTK***

1. Building a unit selection voice for a text-to-speech synthesiser from my own voice. Creating a working voice that can be loaded into Festival (a TTS system by Edinburgh) and used to generate intelligible speech.
2. Varying the contents of the database to discover the effect on the synthetic speech.

***Predicting subject-verb agreement with RNN***

1. Implementing of a few critical parts of a recurrent neural network and the back propagation algorithm.
2. Executing experiments with training regimes, and to adapt the model to an interesting psycholinguistic task that tests the models behaviour on a phenomenon that humans process effortlessly--number agreement between subject and predicate in English.

***Exploring distributional similarity in Twitter***

1. Collecting data from Twitter.
2. Executing experiments to find out what are (some of) the pros and cons of different simple methods for computing similarity between words?
3. Finding out how the different methods are affected by word frequency.
4. Finding out the systematic differences in how different methods rank the similarities between words.

***N-gram Language modeling***

1. Building a trigram language model over characters : read in a text file, collect counts for all character 3-grams, estimate probabilities, and write out the model to a file.

**Skills**

- Knowledge of algorithms about NLP and speech processing
- Proficient in Python
- Familiar with using Linux and Bash
- Familiar with toolkits like Kaldi and TensorFlow

**Languages**

**Chinese**
Mother Tongue
**English**
Advanced