Robust keyword spotting in embedded systems

Ekaterina Chuikova

1 Background and Problem Statement

Speech recognition is one of the interesting modern technologies: with the rapid development of mobile devices, speech-related technologies are becoming increasingly popular. On the one hand, smartphone with speech recognition allows to make user's life more convenient: voice search makes possible to perform more complex search queries in a shorter time. On the other hand, modern speech recognition technologies can make user's leisure more interesting: recently, virtual assistants have been gaining popularity (e.g. Yandex's Alice [1], Google Now [3], Apple's Siri [2], Microsoft's Cortana [4] and Amazon's Alexa [5], etc.). There assistants can conduct a full-fledged voice dialogue with the user, help him, give advice, and most importantly - understand his speech and talk to him.

An interesting technological breakthrough is to enable hands-free speech recognition experience without the user ever touching the device. This feature allows the user to perform actions on his smartphone without being distracted from the main occupation. For example, to build a route while driving.

The following technology is used for the problem solving: the system aim to detect some predefined phrase (wake-up word) from a continuous stream of audio used to initiate the interaction with a device. The problem of detecting of wake-up word is called Keyword Spotting (wake-word detection). For example, a user may say "Ok Google, play some music". Once a trigger phrase (e.g. "Ok Google") is detected on device, the connection is opened to the server and the audio corresponding to the rest of the query (e.g. "play some music") is sent for transcription using a server-side ASR system.

Typical applications exist in environments with interference from background audio, reverberation distortion, and the sounds generated by the speaker of the device in which the KWS is embedded. A KWS system should demonstrate robust performance in this wide range of situations. Furthermore, the computational complexity and model size are important concerns for KWS systems, as they are typically embedded in consumer devices with limited memory and computational resources, such as smartphones or smart-home sensors. These constraints can often result in limited KWS system accuracy. But as KWS system determines different

states of the device, very high detection accuracy for a very low false alarm (FA) rate is critical to enable satisfactory user experience.

2 Purpose

The main purpose is to create robust keyword spotting embedded system. The system must detect the trigger word with small False Accepted and False Rejected rates, and have a small footprint.

3 Literature review

Traditional approaches for KWS are based on Hidden Markov Models with sequence search algorithms [6]. With the advances in deep learning and increase in the amount of available data, state-of-the-art KWS has been replaced by deep learning-based approaches due to their superior performance. Deep learning-based KWS systems commonly use Deep Neural Networks (DNNs)[7], Convolutional Neural Networks (CNNs) [8][9], Recurrent neural networks (RNNs) [10] [11], Convolutional Recurrent Neural Network (CRNN) [12], Attention-based Models [13].

4 What is already done

1. Based on studied literature two types of NNs were chosen and implemented: RNN with attention and CNN.

2. For KWS a dataset is needed in a special format: the word + query, but the data should be processed for direct training and testing. Also, in real conditions system is continiously listen for a wake word, it can be interpreted as testing with a sliding window.

   These NNs were trained with a dataset with 4 different wake words.

   Training data represents 1.4 sec length audio counting wake word (for four classes) and audio without wake word (for the fifth class). Validation is in the same format. Data for test represents a long audio, result is testing with a sliding window.

3. NNs result on test and val were compared, the best results was for RNN.

4. The network learning time, GPU load and metrics (roc-auc, FA rate, FR rate) on test data were considered for comparison.

At the moment current NN shows really good results. For robust system NN should be trained also on noisy data. But in speech recognition validation on data augmentation may be not

enough because of some voice effects in noisy environment (e.g. Lombard effect). For more effective quality control a work with ReSpeaker controller has begun (to test the neural network online in any conditions).

References

[1] https://alice.yandex.ru/

[2] https://www.apple.com/siri/

[3] https://www.google.com/intl/ru/landing/now/

[4] https://www.microsoft.com/en-us/cortana

[5] https://developer.amazon.com/alexa

[6] J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish *Continuous hidden Markov modeling for speaker-independent wordspotting* IEEE Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1990, pp. 627630

[7] G. Chen, C. Parada, and G. Heigold *Small-footprint keyword spotting using deep neural networks* Proceedings International Conference on Acoustics, Speech, and Signal Processing, 2014, pp. 4087-4091.

[8] T. N. Sainath and C. Parada *Convolutional neural networks for small-footprint keyword spotting* in Proceedings of Interspeech, 2015, pp. 1478-1482

[9] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous *Trainable frontend for robust and far-field keyword spotting* arXiv preprint, arXiv:1607.05666, 2016

[10] K. Hwang, M. Lee, and W. Sung *Online keyword spotting with a character-level recurrent neural network,* arXiv preprint arXiv:1512.08903, 2015.

[11] S. Fernandez, A. Graves, and J. Schmidhuber *An application of recurrent neural networks to discriminative keyword spotting,* Artificial Neural Networks. Springer, 2007, pp. 220229

[12] C. Lengerich, and A. Hannun *An end-to-end architecture for keyword spotting and voice activity detection,*arXiv preprint arXiv:1611.09405, 2016.

[13] Changhao Shan, Junbo Zhang, Yujun Wang, Lei Xie *Attention-based End-to-End Models for Small-Footprint Keyword Spotting* Interspeech, 2018

[14] https://www.kaggle.com/c/tensorflow-speech-recognition-challenge