Brendan Jobus - 18321740

Voicefox is a system that helps to defend against voice synthesis attacks on speech verification applications, it does this by extending the authentication process slightly, after passing a sample, Voicefox passes the sample through the transcriber and test the output against the reference text, it uses some extra rules as well because simply rejecting a sample if it doesn't transcribe to be 100% correct would cause lots of false rejections to occur for actual human samples. The reason for needing to do this is because of the vulnerability of speech verification software to speech synthesis. Speech verification has a very high true reject rate(that is, at correctly rejecting voices that are not correct), but it also has a very high false accept rate against synthesized samples(that is, they have a high rate of acceptance when it comes to voices that are synthesizing the desired voice). Due to the fact that the transcriber is almost always already built into the system, the cost for implementing Voicefox is very small, and it itself is not speech verification, but rather, it's an extension of speech verification that is used to help defend against voice synthesis attacks.

The basis of the work is to do with phonemes, STT systems turn spoken words into written text by recognizing phonemes, with speech synthesis, they use a dataset of someone speaking, break down the audio into phonemes and use them to build a model to imitate the speaker. The problem with this is that you can't simply put phonemes together like puzzle pieces as they will just sound robotic, and while this has been solved well enough for us to not notice, for a machine this is not the case.

To prove that their hypothesis that a transcriber is better at detecting a synthesized voice, they tested three transcribers against three state of the art voice transformation and synthesizer tools, they then used a natural voice dataset that they found online and trained a model with datasets that they gathered themselves to train the data and test their hypothesis. What they found was that, with the natural voice dataset, when passing through full sentences, they had a word error rate(the ratio of incorrectly transcribed words to the number of words in the speech) of around 10% when passed through the transcribers, when they used only PGP words, this rose to an average of 50%, with one of the transcribers giving a WER of 85.77%(this makes sense as the transcribers use context and grammar to transcribe text, and with PGP words, this is more difficult because they only have a few words and not a properly constructed sentence). With the other synthesized datasets, they got a WER on average two times higher when it came to regular sentences and double the rate when it came to the PGP words, with one of the datasets averaging over 100% WER(this is possible if the transcriber thinks that they are adding words that actually there).

While this shows that there are more errors produced by the synthesized voice, this doesn't help much if you don't have a way to implement this information, to do this, they created some post transcription rules. Firstly, they set rates at which the WER would cause the system to reject a sample, they tested different values at which this would occur to see how it would affect the false reject rate, they found that with an acceptable WER of ½, they got an FRR of 5% with the natural voice dataset, which had a WER over the entire dataset of around 10%. On top of this, they reject samples in which words are transcribed to words not in the dictionary to reduce the false accept rate. By using only these two rules, they managed to achieve an FAR of 0%.

Source: https://dl.acm.org/doi/10.1145/3427228.3427289