I547 Final Write Up: Limited Speech Recognition

Peter Zhang (pazhang)

1 Introduction: Background and Scope

I decided to implement a simple speech recognition system as my final project for I547. Since speech recognition can get extremely complicated, I had to limit my search space to a confined set of words. I included 15 unique words from the first stanza of *Maamme*, the Finnish national anthem in my vocabulary, which are listed below:

1. suomi
2. synnyinmaa
3. sana
4. kultainen
5. laaksoa
6. kukkulaa
7. ei
8. rantaa
9. rakkaampaa
10. kuin
11. kotimaa
12. pohjoinen
13. maa
14. kallis
15. isien

There are also a few other words in the first stanza that I was unable to include, such as "vettä" and "tää", because I ran into problems with using unicode properly with the programming tools that I chose.

Originally, I was considering using a syllable-based approach that would recognize individual syllables within a given audio input, which can in turn be converted into a word. During

implementation, however, I was not able to decide on how to extract a finite amount of features from a segment of the audio until the very end. Alternatively, I devised a two-stage pipeline that processes a segment of audio, which is certain to contain one of the 15 words, as a whole, with mechanisms that can extract certain features based on the audio segment and thus prunes the search space to limit branching. The system, however, is not very accurate: it works well when detecting some words with specific features, but has a much higher failure rate when it comes to other words.

2 Feature Extraction

2.1 S Extraction

Some of the words in the vocabulary contain the letter *s*. Since written Finnish has a purely alphabetically based phonetical system, every word is read exactly the way it is spelt, and there is only one way each letter can be pronounced. This means that the letter *s* is always pronounced as a hard *s* in English. We can take advantage of this fact since the letter *s* has a very distinct frequency and amplitude combination. Take the word *synnyinmaa* and *laaksoa* for example, figures 2 and 1 show each word's most prominent frequency and amplitude plot, which are convolved with a windowing function for smoothing purposes, and are normalized.

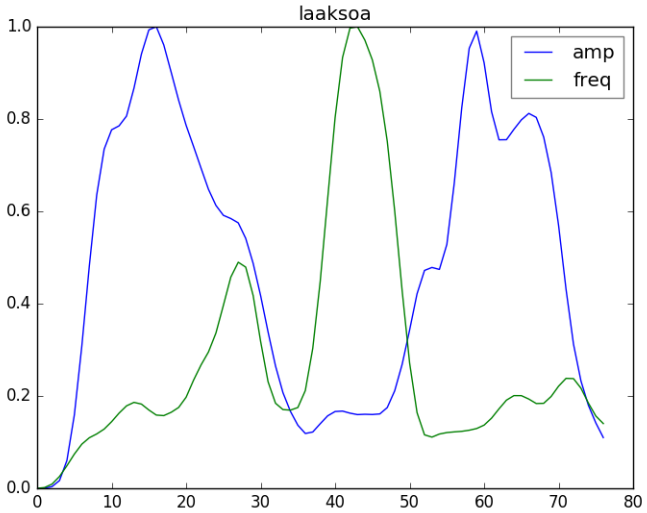

Figure 1: laaksoa

As one can observe in both words, the frequency at where the *s* would be is relatively high, while the amplitude is fairly low. Using this knowledge, we can generalize that whenever we observe a segment with this pattern, then we have seen an *s*. Furthermore, by calculating the location of the *s* within the audio segment, we can even have a confident guess to where

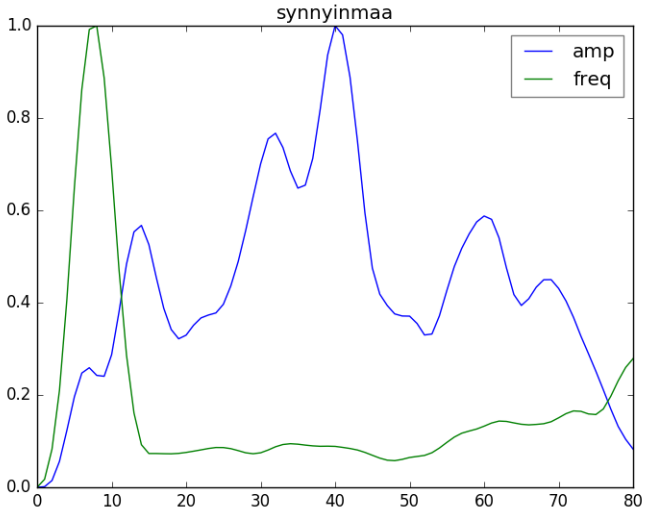

Figure 2: synnyinmaa

the *s* is located. In our examples given above, it is rather clear from the plot that the *s* is most likely located somewhere in the middle for *laaksoa* and somewhere in the beginning for *synnyinmaa*, both of which are correct. By observing the actual frequency of *s*, it can be determined that the frequency cutoff would be somewhere around 3000HZ, thus providing us with a mechanism that may disambiguate the sound of *s* comparing to other consonants.

2.2 -kk Extraction

The strongly gradated consonants *-kk*, *-tt* and *-pp* is a special phonological case in Finnish. Since it is virtually impossible for a human to stretch a consonant like *k* or *t* as one would do with other consonants (such as *s*), the strong gradation is read through adding a short period of silence before reading the sound of the weak gradation. For example, the word *rakkaampaa* is literally read as `ra-(silence)-ka-am-pa-a`. As an example, figures 3 and 4 show the frequency and amplitude plots of *rakkaampaa* and *kukkulaa*.

In both cases, the silence is fairly well presented by having both the amplitude and the frequency dropped below 20% of their peaks. This pattern is especially observable on the amplitude curve, which drops almost to below 1%. Given this characteristic of *-kk* consonants, we can also guess whether a *-kk* sound exists in the word, and effectively limit the possible candidates to only two words, since there are no other words that have the strong consonant gradation pattern.

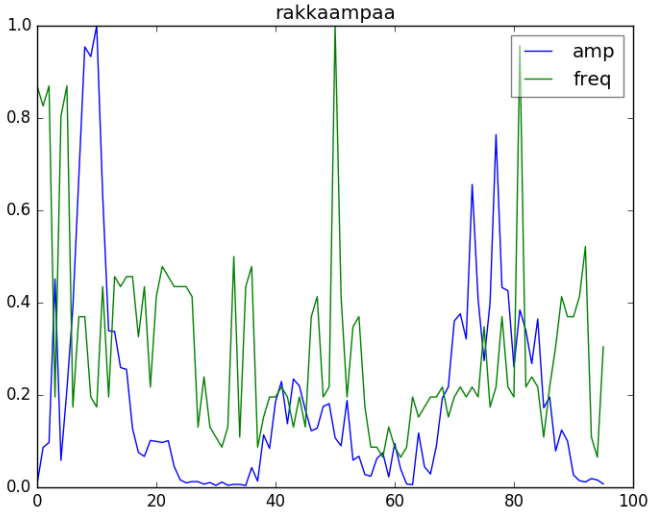

Figure 3: rakkaampaa

2.3 Number of Syllables Detection

Even though it may be difficult to determine where each individual syllable starts and ends in a continuous wave, guessing the number of syllables that exists in each individual sound wave is much easier. Since syllables usually are consists of only one vowel, which generally has a higher amplitude than its accompanying consonants, one can approximate the number of syllables (vowels) within an audio segment by counting the number of distinct local maxima in the amplitude plot. i.e.,

$$arg(X) = \{X|1 \text{ if } X \text{ is true, } 0 \text{ if false}\}$$
$$syllables = \sum arg(\tfrac{dA}{dt} = 0)$$

Where A is the amplitude curve.

One can observe from amplitude plots for sana(figure 6, 2 syllables) and ei(figure 5, 1 syllable) that the local maxima do correspond to the number of syllables.

Using this method, we can further disambiguate between words that may belong to the same categories as determined by the previous methods, such as *sana* and *synnyinmaa*. Here, both words begin with *s* and do not have *-kk-*, but *sana* has only 2 syllables while *synnyinmaa* has between 3 to 4 syllables (the last *a* can either be emphasized or not), thus differentiates the two words.

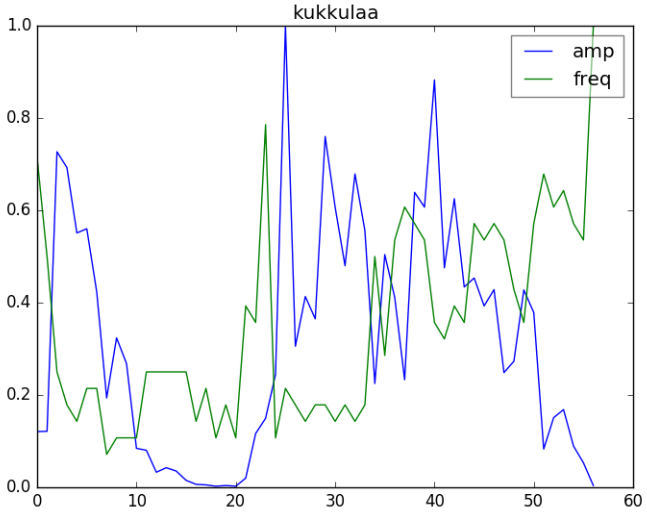

Figure 4: kukkulaa

3 Trend Approximation

After the categorization of an input audio using the extracted features above, we end up with the following categories of words:

1. synnyinmaa (1st s, syllables >=3)
2. suomi sana (1st s, syllables <=2)
3. laaksoa (middle s)
4. isien (early-middle s)
5. kallis (last s)
6. rakkaampaa kukkulaa (-kk, 3 syllables)
7. ei kuin maa (1 syllable, no feature)
8. rantaa (2 syllables, no feature)
9. kotimaa kultainen pohjoinen (3 syllables, no feature)

Since some of the categories only contain 1 word, we can guess that the word represented by the audio signal is that word with some confidence. For those categories with more than one words, however, the "audio signature" of each word will have to be calculated for any possibility of disambiguation among them. In order to do this, the following assumptions are made:

a. Every word has a similar frequency and amplitude trends across different recordings, even if the trend might not completely align with each other at the same time.

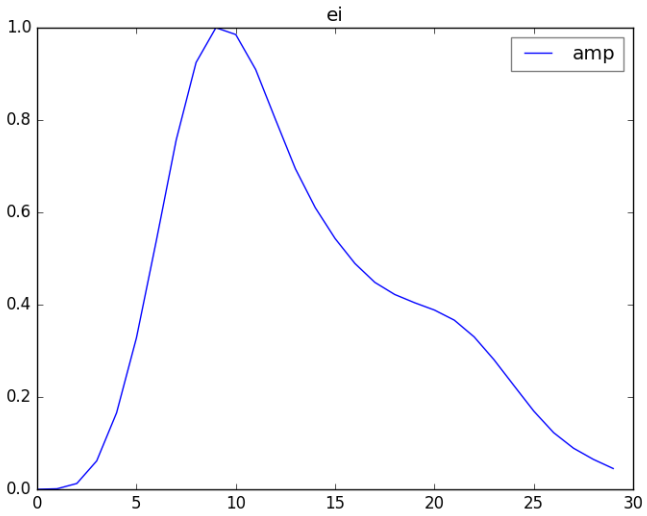

Figure 5: ei

b. Different words display different frequency and amplitude trends comparing to each other.

Take figures 7 and 8 for example. The figures illustrate the frequency and amplitude changes in two different samples of the same word “kotimaa”. As one can observe, both the frequency and the amplitude display similar trends for both waves, with the trend especially visible on the amplitude plot despite of the difference in phase.

Comparing the frequency plot against another three-syllable word like “kultainen”, which is shown in figure 9, one can see that the two are quite different. Their amplitude signatures, howver, are not exactly that much different, as shown in figure 10.

Using a series of sample files that I have recorded myself, I try to capture the general trend of individual words by stretching their frequency and amplitude signatures so that different samples of the same word would have their peaks align, and then compute an average based on the sum of the stretched waves. Take the amplitude signature of “kotimaa” illustrated above as an example, a stretched and averaged amplitude can be seen in figure 11.

After the average is computed for all recorded samples, an incoming, unlabeled wave can be classified through feature extraction described in the previous section, and then be further analyzed by finding a word among the potential candidates that has the least mean squared error in both the amplitude and the frequency signatures. This word is the end result of the pipeline.

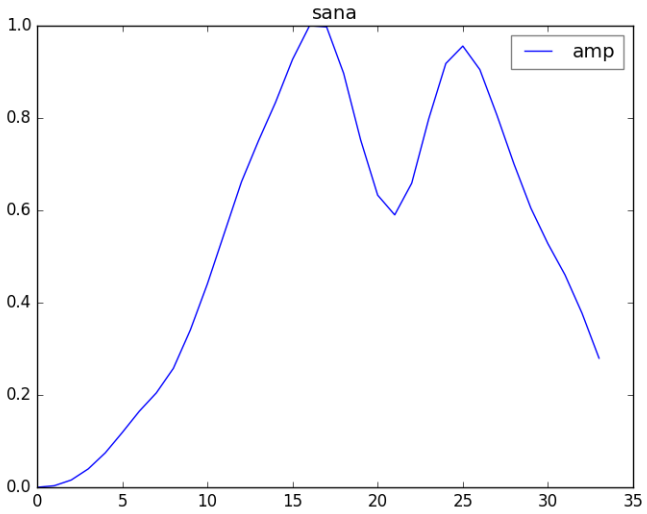

Figure 6: sana

4 Result and Conclusion

Even though the second stage of the pipeline (trend approximation) somewhat made sense in my planning, the actual trials proved the method ineffective, especially when the words that belong to the same category as characterized by the first stage of the pipeline have similar amplitude and frequency signatures, such as “kotimaa” and “kultainen”. As a result, the system can recognize some classes of words rather accurately, while completely fails (almost randomly chooses a candidate) in other categories.

I originally was planning on using a hidden Markov model (HMM) to approach the problem, but I had trouble deciding on how to extract feature vectors from the audio, which made me falling back to the method described in this report. As I was recording the audio samples, however, some of the trends in the audio became rather (painfully) obvious just by looking at their spectrograms. For example, the word “pohjoinen” displays a signature decrease and then increase in its most prominent frequency around the syllables “-joi-”, while the word “kultainen” has an increasing frequency near the syllable “-tai-”. If I had more time, I would extract features in the frequency-amplitude domain and implement an HMM model instead.

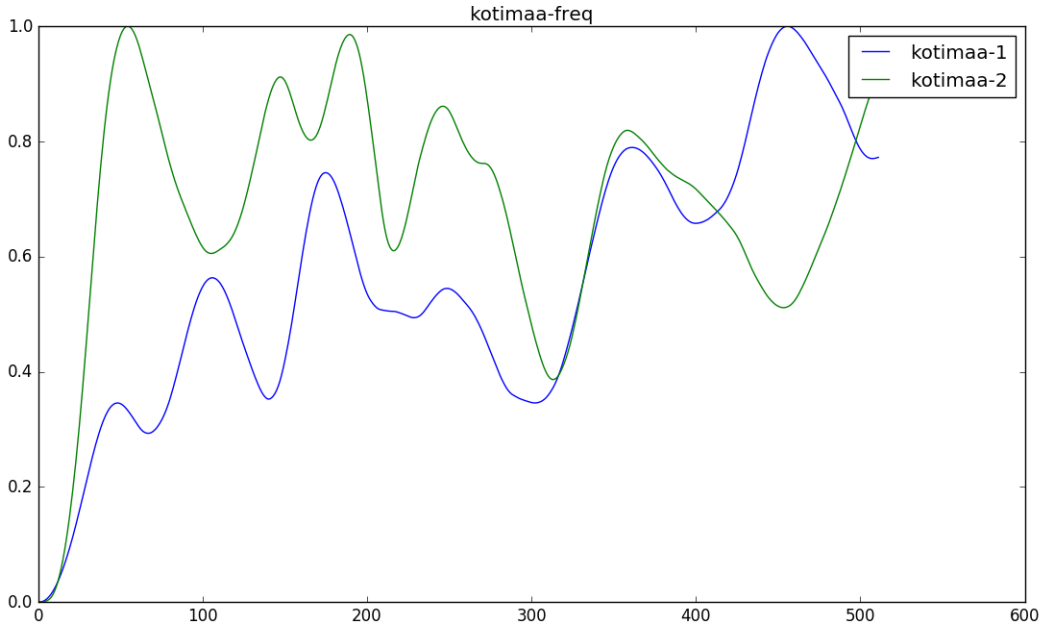

Figure 7: Kotimaa's frequency trend

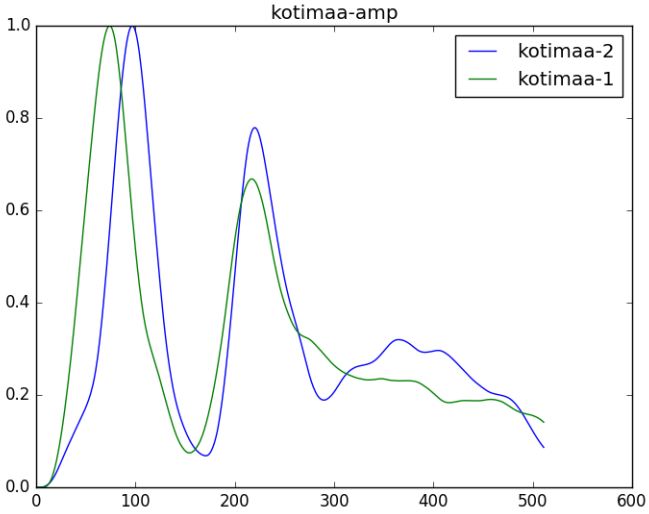

Figure 8: Kotimaa's amplitude trend

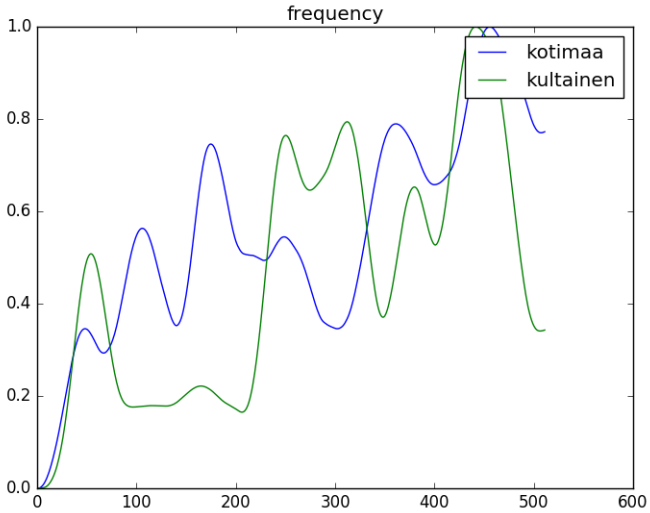

Figure 9: Kotimaa and Kultainen's frequency trends

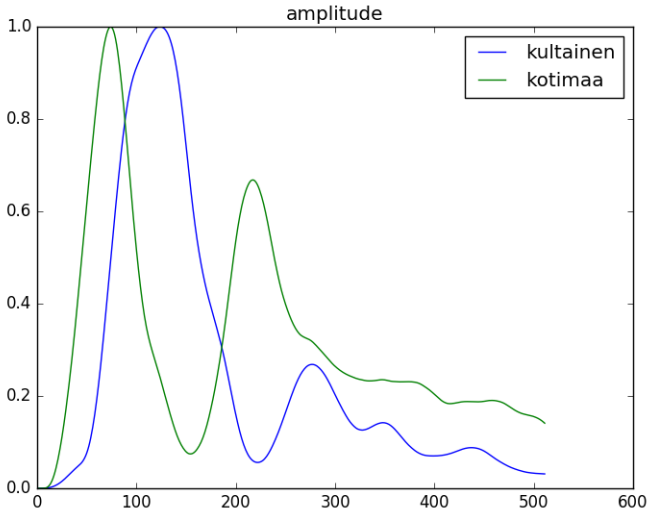

Figure 10: Kotimaa and Kultainen's amplitude trends

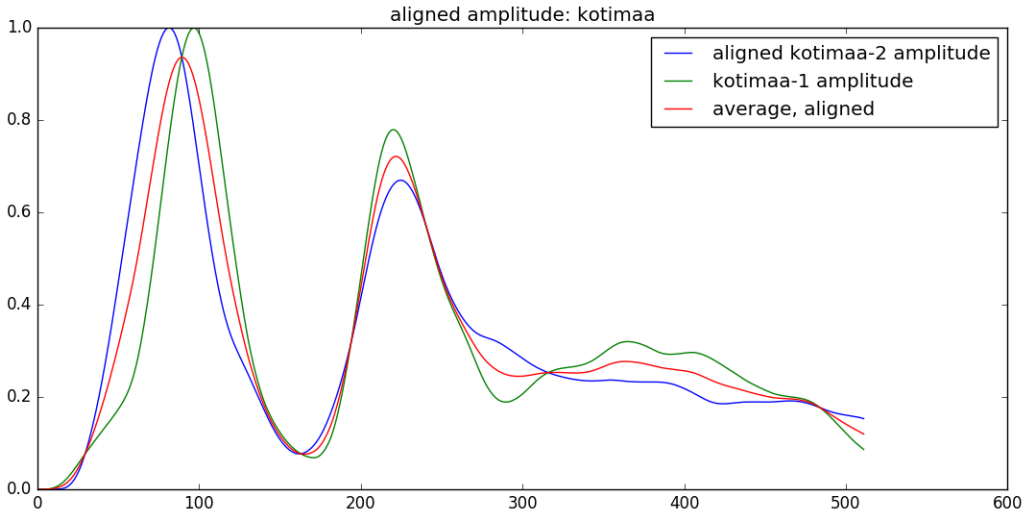

Figure 11: Kotimaa's amplitude signature, aligned