Pronunciation variation and its importance for speech recognition

DT2118 Term Paper by

Corentin Valleroy
Florian Lochet
Olivier Hamon

2012-05-31

Introduction

Since some years, speech recognition systems are often used thanks to smartphone operating systems such as Android and iOS. It is easy to use and improves the user experience.

With modern technique and modern technology, normal speech can be easily recognized and decrypted by many devices. The problem is that many countries and languages have region specific accent and pronunciation. Everyone is unique and has his own speaking styles; many factors are interacting, such as dialect and accent from the native town. Moreover, there is a difference between native and non-native speaker. It is still difficult for Automatic Speech Recognition (ASR) technology, to parse non-native speech. As another example, Siri, the Apple speech recognizer, doesn't handle well the accent of the north of England.

One big challenge for the speech recognition world is to well understand all the pronunciation phenomena for modeling individual variation in spoken language.

There are different techniques for modeling pronunciation variation, and they can be implemented in different part of the speech recognizer system. In this report, we tried to talk about some techniques to handle these pronunciation variations.

1. Lexicon Adaptation

When an ASR uses units other than words for modeling the acoustic model, a lexicon is used. A lexicon is basically the correspondence between the acoustic model units and the words in the vocabulary [1].

When the lexicon uses multiple pronunciations for the same word, it is possible to use pronunciation probability. Using this, it is possible to inhibit confusions due to rare pronunciations.

Considering the basic ASR classifier equation (1), this equation can be decomposed to include the pronunciations β to the word W.

$$\widehat{W} = arg\max_W p(O|W)\,P(W) \quad (1)$$

$$\begin{aligned} p(O|W)\,P(W) &= \sum_{B\in\beta} p(O|B,W)\,P(B|W)P(W) \quad (2) \\ &\approx \max_{B\in\beta}[p(O|B,W)P(B|W)P(W)] \end{aligned}$$

The third line is the Viterbi approximation of using only the best pronunciation. The pronunciation probability P(B|W) is included in the language model, and we have the language model probability P(B|W)P(W) [1].

There are two different kind of modeling for these methods: the direct and the indirect one. With the former, pronunciation variants are derived directly for each word. With the latter, pronunciation rules are indirectly created and used to generate new pronunciations.

1.1 Lexicon manually generated

The naive approach for handling pronunciation variation is to manually transcribe all the different pronunciations for all the different words in the lexicon or all the different pronunciation rules. It is called knowledge-based method. The creation of the lexicon consists of using phonological rules and knowledge sources, for example handcrafted dictionaries or the linguistic literature to generate variants. It is long to perform and it is quite difficult to think to every single pronunciation rules.

REALIZE	r iy1 l ay2 z
REALIZE (2)	r iy1 ah0 l ay2 z

Figure 1: Example of the CMU lexicon with multiple pronunciation entries for one word.

1.2 Lexicon generated by computer

Creating the lexicon by hand is time consuming and mistakes can easily appear. Generating the lexicon by computer could be more efficient and the result could be better. Basically, the ASR uses databases of speech to find the variations present into the language. This approach is called data-driven pronunciation modeling. This technique focuses on finding the pronunciations that are the more efficient for the objective criterion [3].

There are different techniques to do that, but the main concept is the same.

- The first step is to automatically generate alternative transcriptions of reference sentences. This will reveal the true pronunciation of the speaker. To perform that, we can use knowledge or a phone recognizer.

- The second step is to align the reference and the alternative transcriptions. The same phones are mapped together to figure out which sound is added or deleted.

- The third step is to derive rules from the alignment. Many procedures exist, the most famous one is to use CART tree for generalizing the context of a rule automatically.

- The fourth step is to assess and prune the tree. A lot of rules could be generated with the previous step. For example rules that are rare could be deleted.

- The fifth step is the generation of pronunciation variants from the rules for modifying the lexicon. Some prunings have to be done to limit the size of the lexicon.

1.3 Results

Some studies have shown that increasing the recognizer lexicon with pronunciation variants found in a general-purpose does not improve the performance of the ASR [4] as we can see in the table below.

Multiple pronunciation by word	Speaker 1 word error rate	Speaker 2 word error rate	Speaker 3 word error rate	Speaker 4 word error rate
no	34.7	40.4	27.6	22.2
yes	32	38.2	26.7	20.1

According to Yang, Martens and Kessens [2], the average number of variants per word should not be higher than 2.5, or the system would perform worse than the baseline system without multiple variants.

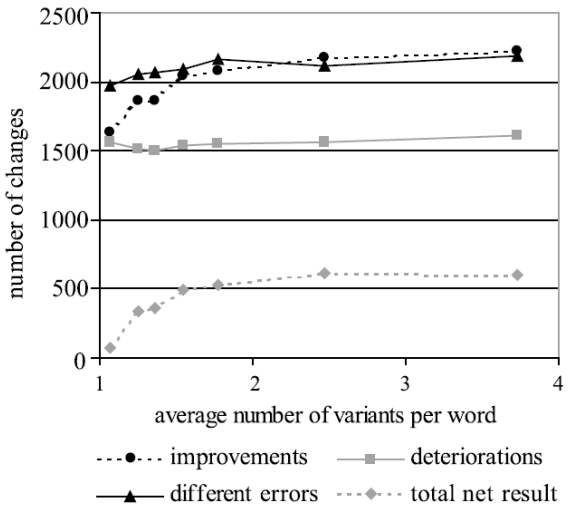

Figure 2: Different types of changes for testing condition on the error analysis

2. Variation decision tree and pronunciation variation model

The two previous parts show how we can create a new lexicon from rules or pronunciation variants in order to improve the robustness to pronunciation variations. However, it can be time-consuming to create such lexicon, and we are restricted by the size of the dictionary, which can't contain all possible variations of every word in the language: the lexicon would otherwise be much too large. Indeed, the variations are generated by many factors: the local accents (dialects), the fact that the speaker may not speak in his native language, whether it is spontaneous speech or well-articulated speech, and even the age of the speaker.

In order to avoid this problem, but keeping the benefits of this kind of implementation of pronunciation variations, different methods can be used: pronunciation decision tree or pronunciation variation model. Both are based on rules that describe the variations in the pronunciations of some phones.

As there is no dictionary generated from these rules, they can be as accurate as needed. For instance, we could choose some rules like the confusion between short and long vowel:

- i => in "**i**s" it is a short vowel and in "sk**i**" it is a long vowel: /I/ <=> /i/
- o => in "h**o**t" it is a short vowel and in "hell**o**" it is a long vowel: / ɑ/ <=> / oʊ /
- …

These particular rules are more likely to be useful if the speaker is not using his native language, but it may also be the case if we try to decode spontaneous speech after training on well-articulated speech (spontaneous speech will shorten the vowels). Although we don't always know the situation in which the system will be used, sometimes it is possible to choose the rules used depending of the situation: if the users are more likely to be foreigners for instance, some specific set of rules could be chosen rather than another to handle the most current mistakes in the pronunciation.

An important question is the generation of this set of rules. As we said before, it can be created by hand, using some known pronunciation variations, as the ones we mention before. But instead of these knowledge-based methods, we can also use data-driven methods that use real speech to find the pronunciation variations automatically [1] (as we have seen in the part 2).

2.1 Pronunciation variation decision tree

This method consists in creating a decision tree for each word representing its pronunciation variations. To create the tree, we place the first phoneme of the word at the root, and then at each step we consider the phoneme next to the last one, and the question is: "What are the possible pronunciations of this phoneme, given the previous and the next phoneme?"

The possible pronunciations are given by the rules that we discussed before. Here is an example coming from [5], which shows the decision tree for the word "rak^" in Thai language; the rules used to generate the tree are:

- /r/ <=> /l/ (/r/ sometimes becomes /l/ and the contrary also, depending of the accent)
- /a/ <=> /aa/ (short and long vowel)

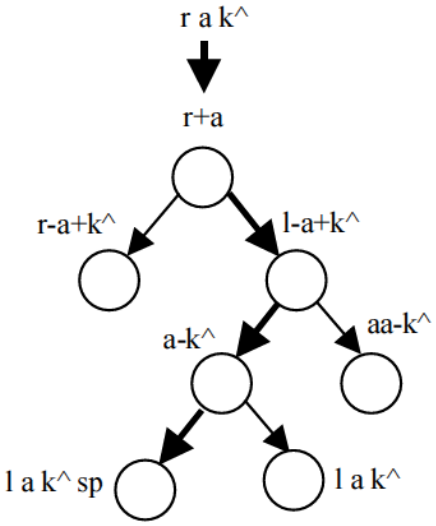

Figure 3: example of a pronunciation variation decision tree

The root asks “what are the possible pronunciations of /r/ followed by /a/?”. According to the rules, it can be /r/ or /l/, thus we create the second level. Then the question is “what are the possible pronunciations of /a/ followed by /k^/ and preceded by /l/?”: /a/ or /aa/.

To train the system, a pronunciation variation re-label algorithm is used: the speech database is used to train the initial acoustic model with the re-estimation algorithm and the present phoneme transcription. Then the Viterbi algorithm is used on the decision trees to find the best pronunciation of each word given the acoustic model, and the phoneme transcription is updated. This is applied several times until the model converges, which means that the log probability is less than in the last model.

2.2 Pronunciation variation model

This method doesn’t use a variation decision tree, and it doesn’t create a new lexicon either. Instead, we train the model from the initial transcriptions.

As in [5], we want to be able to recognize a word when the phoneme /I/ (short vowel) is pronounced /i/ (long vowel) for instance, the model is modified: we can tie the start of the /I/ state with the start of the /i/ state, and tie their end in the same way. This shows what we would have:

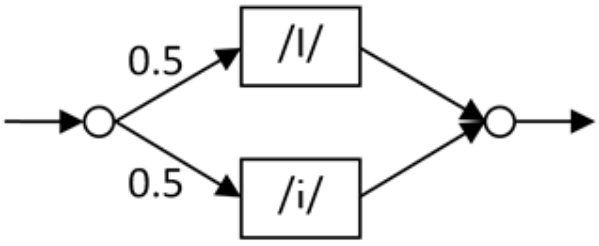

Figure 4

We choose a probability of 0.5 for each pronunciation, but we can train again the new model in order to obtain the maximum likelihood solution. This way, doing the same for all rules allows the model to handle the pronunciation variations described by the rules.

2.3 Results

The comparison of these two different methods is realized in [5]. They used three different initial phoneme transcriptions to compare the results: the transcriptions generated automatically by using Thai Grapheme-to-Phoneme (G2P) developed by NECTEC [6] (I), the transcriptions edited by expert labelers (II), and the transcriptions generated from re-label training processes (III). They also used three different methods: with no specific pronunciation variation technique (Figure 5 - top), with the re-label algorithm using the pronunciation variation decision tree (Figure 5 - middle), and using the pronunciation variation model (Figure 5 - bottom).

Phoneme transcription	% correction	% Accuracy
I	70.36	67.87
II	78.52	73.63
III	74.01	71.56

Phoneme transcription	% correction	% Accuracy
I	77.87	72.77
II	78.52	73.63
III	78.11	72.91

Phoneme transcription	% correction	% Accuracy
I	77.66	72.46
II	79.42	74.11
III	80.46	75.42

Figure 5: with no specific (top), re-label algorithm (middle), using the pronunciation variation model (bottom)

We see that depending of the initial phoneme transcription, using the re-label algorithm gives similar or better results, up to +7.51% for correctness and +4.90% for accuracy. The pronunciation variation model improves the results even more as it reaches 80.46% of correctness and 75.42% for accuracy with initial phoneme transcription III.

3. Acoustic model improvement

The pronunciation variation can also directly be handled by the acoustic model. The adaptation of the model can increase its robustness, when building a robust acoustic model is one of the main challenges in the field of speech recognition. The robustness of a particular acoustic model goes through different parameters, such as the selection of an appropriate acoustic unit.

Depending of the kind of speech studied, units often used for modeling are word, syllable, and phone. Most speech recognition systems use phone as the unit of modeling, since it is a good trade-off between the amount of speech required and the over generalization, to have a robust model. Using other unit than a phone has been an interest for a long time and has been investigated for example in [7]. In a study of Holter and Svendsen [8], they described optimal acoustically based units with units shorter than phones.

After the selection of the unit for modeling, we can use different approaches for modeling them, for example Hidden Markov Model (HMM), artificial neural network (ANN) [9] or template model, etc.

Since Hidden Markov Model is one of the most widely used approaches in statistical speech recognition, because of its robustness, we will focus on it and on its possible improvements in this paper.

As in ANN field, a possible improvement is to start the learning of the model already with data. Usually an HMM is initialized with a flat initialization by calculating a global means and variances, but we can also start with better and more accurate data. This will permit to improve the final learning state, and by sharing the data with the community of speech recognition, to have a better and better model. Obviously this depends on the overall parameters of the model (number of HMM, acoustic unit used, context…). As seen in [10], an equi-probable initialization works slightly better than a typical random initialization.

In spite of all the advantages of a HMM, this model is limited in the expression of state's duration. The classical HMM has not any modeling of the duration in one state. Indeed the probability to stay in one state a consecutive number of periods Pi(d) can be calculated as the probability of (d-1) auto transitions followed by a transition to the next state. Thus, we can add this modeling of the duration, because a difference in pronunciation may induce a difference in duration. A first means to model this duration is to incorporate it in the global recognition unit, which means to model the time spent in one state without taking into account the previous state [11][12].

Another way to improve HMM is to change the topology. The most common is the left to right topology, as shown in this figure:

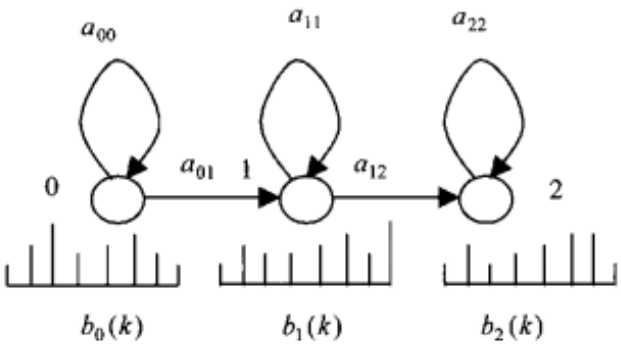

Figure 6: Typical left to right topology for HMM

The number of states in this model depends on the acoustic unit used. For example for a phone it is 3 to 5 states, for a short word 2-3 states per phoneme, and for a long word it is 1 or 2 states per phoneme. In a speech recognition system with large vocabulary, multiple HMMs will be used to model the whole acoustic model. Each HMM models an acoustic unit.

Normally, each probability is defined by a single Gaussian. To add more flexibility to the model and to enable it to better model the pronunciation variation, we can add more Gaussian mixture to the model. However, this will cause some problems such as longer time of computation (for example for the re-estimation step), and of training [13].

4. Conclusion

In this paper we have seen the importance and the problems that can be caused by variations of pronunciation in ASR. This was only a general overview and state of the art of some different techniques which can be used to prevent that, and to limit its influence on the recognition capabilities. These techniques can improve the robustness at different levels of the ASR, such as the acoustic model or the lexicon. Maybe some improvements could also be done by taking into account the grammar in the language model.

The importance of the pronunciation is a very wide field with many ongoing researches.

REFERENCES:

[1] Ingunn Amdal, Eric Fosler-lussier, "*Pronunciation Variation Modeling in Automatic Speech Recognition*", 2003.

[2] Yang, Q., J.-P. Martens, "*On the importance of exception and cross-word rules for the data-driven creation of lexica for ASR*", 2000.

[3] Mirjam Wester, "*Pronunciation modeling for ASR – knowledge-based and data-derived methods*", 2002.

[4] Amdal, Svendsen, *"T. Evaluation of pronunciation variants in the ASR lexicon for different speaking styles"*, 2002.

[5] Supphanat Kanokphara, Virongrong Tesprasit, Rachod Thongprasirt, "*Pronunciation Variation Speech Recognition Without New Dictionary Construction*", 2003.

[6] P. Tarsaku, V. Sornlertlamvanich, R. Thongprasirt, *"Thai Grapheme-to-Phoneme using Probabilistic GLR Parser"*, 2001.

[7] T. Svendsen, K. K. Paliwal, E. Harborg, P. O. Husby, "*An improved sub-word based speech recognizer*", 1989.

[8] T. Holter, T. Svendsen, "*Incorporating linguistic knowledge and automatic base form generation in acoustic subword unit based speech recognition*", 1997.

[9] Ken Chen, Mark Hasegawa-Johnson, *"Modeling pronunciation variation using artificial neural networks for English spontaneous speech*", 2004.

[10] M.A. Ferrer, I.G. Alonso and C.M. Travieso, "*Influence of initialization and stop criteria on HMM based recognisers*", 2000.

11] Furguson J. D., "*Variable duration models for speech*", 1980.

[12] Vasegh S., "*State duration modeling in hidden Markov models*", 1995.

[13] Lawrence R. Rabiner, B. H. Juang, "*Mixture autoregressif hidden Markov models for speech signals*", 1993.