Measuring the confusability of pronunciations in speech recognition

Panagiota Karanasou
LIMSI/CNRS
Université Paris-Sud
pkaran@limsi.fr

François Yvon
LIMSI/CNRS
Université Paris-Sud
yvon@limsi.fr

Lori Lamel
LIMSI/CNRS
lamel@limsi.fr

Abstract

In this work, we define a measure aimed at assessing how well a pronunciation model will function when used as a component of a speech recognition system. This measure, *pronunciation entropy*, fuses information from both the pronunciation model and the language model. We show how to compute this score by effectively composing the output of a phoneme recognizer with a pronunciation dictionary and a language model, and investigate its role as predictor of pronunciation model performance. We present results of this measure for different dictionaries with and without pronunciation variants and counts.

1 Introduction

As explained in (Strik & Cucchiarini, 1999), pronunciation variations can be incorporated at different levels in ASR systems: the lexicon, the acoustic model, the language model. At the acoustic level, context dependent phone modeling captures the phone variations within particular contexts. At the lexicon level, a lexicon with alternative pronunciations is used. At the language model (LM) level, the inter-word pronunciation variations are handled with grammar network, statistical LMs or multi-word models.

The growing interest in automatic transcription of Conversational Speech (CTS) increases the need for modeling pronunciation variation. Indeed, there is a large number of possible pronunciation variants occurring in spontaneous speech; these variants often extend beyond single speech sounds (modeled by the acoustic model) and reach up to whole words or word tuples. Not even context-dependent acoustic models for sub-word units (like phonemes) are able to cover pronunciation variants of this kind (Kipp *et al*, 1997). Thus, pronunciation variation is usually modeled by enumerating appropriate pronunciations for each word in the vocabulary using a pronunciation lexicon.

However, when adding alternative pronunciations to a lexicon, there is always the potential of introducing a detrimental amount of confusability. The homophone (words that sound the same but are written differently) rate increases, which means that these additional variants may not be helpful to the recognition performance (Tsai *et al*, 2001). A typical example in English is the word *you*: the received pronunciation is */yu/* and is chosen when one single variant is used; modeling some variation requires to consider the pronunciations */yu/* and */yc/*, which both occur in our multiple pronunciation dictionary. The latter pronunciation (*/yc/*), in the phrase *you are* is easily confused with */ycr/*, the pronunciation of *your*. Such confusions, in particular when they involve frequent words, can cause a degradation of the ASR system as more alternatives are added.

A lot of work has been carried out on the generation of pronunciation and pronunciation variants independently of the speech (g2p conversion, p2p conversion) or in a task specific framework using surface pronunciations generated from a phoneme recognizer or including acoustic and language model information. However, most works lack a sense of how added alternative pronunciations will affect the overall decoding process. For example, some

Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing, pages 107–115,
Blois (France), July 12-15, 2011. ©2011 Association for Computational Linguistics

of the confusability introduced by the pronunciation model is compensated by the LM. Thus, a method for quantifying the confusion inherent in a combined acoustic-lexical system is needed. A confusability measure traditionally used to measure the uncertainty residual to a system is entropy. Specifically in an ASR system, lexical entropy measures the confusability introduced by an LM. In some previous works, lexical entropy not only takes the LM scores into account, but also integrate the scores of the acoustic and pronunciation models (Printz & Olsen, 2000). In (Wolff *et al*, 2002), the authors consider as a measure of the pronunciation confusability the entropy of the variant distribution, but they do not take into account the language model. Our aim is to integrate pronunciation model and language model information into a single framework for describing the confusability. Especially incorporating language model information would provide a more accurate reflection of the decoding process, and hence a more accurate picture of the possible lexical/acoustic confusions (Fosler-Lussier *et al*, 2002). The idea is then to introduce a measure inspired by the proposed formulation in (Printz & Olsen, 2000) but in a somewhat reverse fashion. Instead of measuring the "true" disambiguation capacity of the LM by taking acoustic similarities into account, we aim at measuring the actual confusability introduced in the system by the pronunciation model, taking also into account the LM. We call this measure *pronunciation entropy*.

To compute this measure, we will decompose the decoding process in two separate parts: the acoustic decoding on the one hand, the linguistic decoding on the other hand. Given an input signal, a phoneme recognizer is first used to obtain a sequence of phonemes; the rest of the decoding process is realized using a set of Finite State Machines (FSMs) modeling the various linguistic resources involved in the process. Doing so allows us to measure the confusability incurred by the acoustic decoder for fixed linguistic models; or, conversely, to assess the impact of adding more pronunciations, for fixed acoustic and language models. This latter scenario is especially appealing, as these measurements do not require to redecode the speech signal: it thus become possible to try to iteratively optimize the pronunciation lexicon at a moderate computational cost. Experiments are carried out to measure the confusability introduced by single and multiple pronunciation dictionaries in an ASR system, using the newly introduced pronunciation entropy.

The remainder of the paper is organized as follows. Section 2 describes the necessary Finite State Tranducers (FSTs) background. Section 3 presents the FST decoding and details the new confusability measure. Sections 4 and 5 present the recognition experiments and the pronunciation entropy results. The paper concludes with a discussion of the results and of some future work in Section 6.

2 Background

2.1 Generalities

In the last decade, FSTs have been shown to be useful for a number of applications in speech and language processing (Mohri *et al*, 1997). FST operations such as composition, determinization, and minimization make manipulating FSTs both effective and efficient.

Weighted transducers (resp. automata) are finite-state transducers (resp. automata) in which each transition carries some weight in addition to the input and output (resp. input) labels. The interpretation of the weights depends on the algebraic structure of the semiring in which they are defined.

A *semiring* is a system $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ containing the weights $\mathbb{K}$ and the operators $\oplus$ and $\otimes$, such that: $(\mathbb{K}, \oplus, \bar{0})$ is a commutative monoid with $\bar{0}$ as the identity element for $\oplus$; $(\mathbb{K}, \otimes, \bar{1})$ is a monoid with $\bar{1}$ as the identity element for $\otimes$; $\otimes$ distributes over $\oplus$: for all a, b, c in $\mathbb{K}$: $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ and $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$, and $\bar{0}$ is an annihilator for $\otimes : \forall a \in \mathbb{K}, a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$. When manipulating weighted tranducers, the $\otimes$ and $\oplus$ operators are used to combine weights in a serial and parallel fashion, respectively. A semiring is idempotent if for all $a \in \mathbb{K}, a \oplus a = a$. It is commutative when $\otimes$ is commutative.

The real semiring $(\mathbb{R}, +, \times, 0, 1)$ is used when the weights represent probabilities. The semirings used in this work are the log semiring, the entropy semiring, as well as a new, defined for computational reasons, log-entropy semiring. The log semiring is defined as $(\mathbb{R} \cup [-\infty, \infty], -\log(\exp(-x) + exp(-y)), +, \infty, 0)$. It is isomorphic to the real semiring via the negative-log mapping and is used

in practice for numerical stability.

A *weighted finite-state transducer* T over a semiring $\mathbb{K}$ is an 8-tuple $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ where: Σ is the finite input alphabet of the transducer; Δ is the finite output alphabet; Q is a finite set of states; $I \subseteq Q$ the set of initial states; $F \subseteq Q$ the set of final states; $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \times \mathbb{K} \times Q$ a finite set of transitions; $\lambda : I \to \mathbb{K}$ the initial weight function; and $\rho : F \to \mathbb{K}$ the final weight function mapping F to $\mathbb{K}$.

A *weighted automaton* $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ is defined in a similar way simply by omitting the output labels. The weighted transducers and automata considered in this paper are assumed to be trimmed, i.e. all their states are both accessible and co-accessible. Omitting the input (resp. output) labels of a weighted transducer T results in a weighted automaton which is said to be the output (resp. input) projection of T.

Using the notations of (Cortes *et al*, 2006), if $e = (q, a, b, q')$ is a transition in E, $p(e) = q$ (resp. $n(e) = q'$) denotes its origin (resp. destination) state, $i(e) = a$ its input label, $o[e] = b$ its output label and $w(e) = E(e)$ its weight. These notations extend to paths: if π is a path in T, $p(\pi)$ (resp. $n(\pi)$) is its initial (resp. ending) state and $i(\pi)$ is the label along the path. We denote by $P(q, q')$ the set of paths from q to q' and by $P(q, x, y, q')$ the set of paths for q to q' with input label $x \in \Sigma^*$ and output label $y \in \Sigma^*$. The path from an initial to a final state is a successful path. The output weight associated by a weighted transducer T to a pair of strings $(x, y) \in \Sigma^* \times \Sigma^*$ is denoted by $T(x, y)$ and is obtained by $\otimes$-summing the weights of all successful paths with input label x and output label y:

$$T(x, y) = \bigoplus_{\pi \in P(I,x,y,F)} \lambda(p[\pi]) \otimes w[\pi] \otimes \rho(n[\pi]) \tag{1}$$

$T(x, y) = \bar{0}$ when $P(I, x, y, F) = \emptyset$.

The *composition* of two weighted tranducers T_1 and T_2 with matching input and output alphabet Σ, is a weighted transducer denoted by $(T_1 \circ T_2)$ when the semiring is commutative and when the sum:

$$(T_1 \circ T_2)(x, y) = \sum_{z \in \Sigma^*} T_1(x, z) \otimes T_2(z, y) \tag{2}$$

is well-defined and in $\mathbb{K}$ for all x, y.

2.2 Entropy Semiring

The entropy $H(p)$ of a probability mass function p defined over a discrete set X is defined as (Cover & Thomas, 1991):

$$H(p) = -\sum_{x \in X} p(x) log p(x), \tag{3}$$

where, by convention, $0 log 0 = 0$. This definition can be extended to probabilistic automata which define distributions over sets of strings. We call an automaton probabilistic if for any state $q \in Q$, the sum of the weights of all cycles at q is well-defined and in $\mathbb{K}$ and $\sum_{x \in \Sigma^*} A(x) = 1$. A probabilistic automaton such that at each state the weights of the outgoing transitions and the final weight sum to one, is a stochastic automaton. The entropy of A can be written as:

$$H(A) = -\sum_{x} A(x) log A(x), \tag{4}$$

where $A(x)$ is the output weight associated by an automaton A to an input string $x \in \Sigma^*$.

The expectation (or entropy) semiring is defined in (Eisner, 2001) as $(\mathbb{K}, \oplus, \otimes, (0, 0), (1, 0))$, where $\mathbb{K}$ denotes $(\mathbb{R} \cup [-\infty, \infty]) \times (\mathbb{R} \cup [-\infty, \infty])$. For weight pairs (a_1, b_1) and (a_2, b_2) in $\mathbb{K}$, the $\oplus$ and $\otimes$ operations are defined as follows:

$$(a_1, b_1) \oplus (a_2, b_2) = (a_1 + a_2, b_1 + b_2) \tag{5}$$

$$(a_1, b_1) \otimes (a_2, b_2) = (a_1 a_2, a_1 b_2 + a_2 b_1) \tag{6}$$

The entropy of A defined in equation (4) can be seen as a single-source shortest distance for an automaton defined over the entropy semiring (Cortes *et al*, 2006) with weights $(w, -w log w)$ where $w \in \mathbb{R}$. If the sum of the weights of all paths from any state $p \in Q$ to any state $q \in Q$ is well-defined, the shortest distance from p to q is:

$$d[p, q] = \bigoplus_{\pi \in P(p,q)} w[\pi]. \tag{7}$$

Thus, the shortest distance from the initial states to the final states for the probabilistic automaton A with weights $(w, -w log w)$ in $\mathbb{K}$ will be:

$$d[I, F] = (\sum_{x} A(x), -\sum_{x} A(x) log A(x)) \tag{8}$$

$$= (1, H(A)). \tag{9}$$

3 A new confusability measure

3.1 ASR decoding with FSTs

The recognition process can be modeled with a sequence of weighted finite-state transducers (WFSTs) (Pereira & Riley, 1996). An abstract representation of the Viterbi decoding process of the present work can be given as:

$$\hat{W} = bestpath(A \circ P \circ L \circ G), \qquad (10)$$

where $\hat{W}$ is the sequence of words corresponding to the best recognition hypothesis. A is the phoneme hypothesis lattice generated by the phoneme recognizer, P is an FST that contains a mapping from phonemes to the phonemic lexical representation of each word, L is the pronunciation model FST, containing a mapping from each phonemic lexical representation to the corresponding word, G is the language model finite state automaton (FSA), which contains n-gram statistics, and $\circ$ is the composition operator. Constraining the model by the pronunciation and the language models means that only words that are part of complete paths in the decoding will be counted as confusions. In this work, the FSTs and FSAs will be manipulated using the open-source toolkit OpenFst (Allauzen *et al*, 2007).

3.2 Decomposing the acoustic and linguistic modeling

In a first place, a phoneme recognizer generates the phoneme hypothesis lattice A from the speech signal. These phonemes are the input in the following process of consecutive compositions. The phoneme lattices are generated by the ASR system without taking into account the pronunciation nor the language model during decoding. The aim is to decompose the decoding parts in order to better evaluate the influence of the pronunciation model in the decoding process. The acoustic scores are considered stable and independent of the linguistic (pronunciation and language) confusability and thus are omitted. No time information is kept. The pronunciation model will automatically segment the phoneme sequences in pronunciations, and consequently in words.

The FST P representing the set of valid pronunciations in our lexicon (see Section 4) is then constructed; it takes as input a sequence of phonemes and returns the sequence of corresponding phonemic lexical representation (pronunciation). P is composed with each phoneme lattice. In order to account for insertions of phonemes between valid pronunciations, the topology of P is slightly expanded. This expansion simulates a simple error recovery strategy consisting in deleting superfluous phonemes in a left to right fashion. Fig. 1 illustrates this expansion on a simple example, with the use of failure transitions implemented with the so-called *phi-matchers* and *rho-matchers*. Each state in P corresponds to the prefix of an actual pronunciation: whenever we reach a state from which no continuation is possible, a phi-transition allows to reach the state corresponding to a trimmed prefix, from which the first phoneme has been deleted. This simple error recovery strategy is applied recursively. A rho-loop is finally used in the initial state, which is also the final state, in case the first or last phonemes of a sequence do not permit to complete a known pronunciation. Assume, for instance, that we reach state 2 in P (see Fig. 1), and that the following symbol is 'c', for which no transition exists. The phi-transition will then allow to move to state 3, and continue the prefix 'bc'.

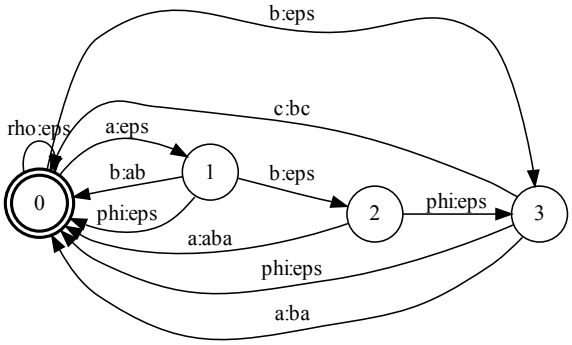

Figure 1: Expansion of the topology of the P FST with phi matchers that consume the phonemes inserted between valid pronunciations

Next, the FST L representing the pronunciation dictionary with pronunciations as inputs and words as outputs is constructed. Its weights are the conditional probabilities of a pronunciation given a word.

When no pronunciation probabilities are available, a uniform distribution over the probabilities of pronunciations of each word is applied. This FST is composed with each phoneme-pronunciation FSTs $A \circ P$ resulting from the previous composition.

A final composition is made with the FSA G that models the backoff language model, with word probabilities as weights. G is constructed as described in (Riccardi *et al*, 1996; Allauzen *et al*, 2003). This results in FSTs with phonemes as input and words as output, which are projected to the output and determinized. Then, the arc weights of each FST are normalized per state, i.e. scaled such that the probability of arcs leading out of a state (plus the probability of state finality) sums to 1 for each state. A general weight-pushing algorithm in the log semiring (Mohri *et al*, 1997) is applied for the normalization and the weights in the new stochastic FSA are converted to the desired posterior probabilities given the pronunciations. What is calculated is the conditional probability $p(w \mid a)$ of all the word sequences that can be transcribed as a and, thus, are competitors:

$$p(w \mid a) = \frac{p(a \mid w)p(w)}{\sum_{w \in W} p(a \mid w)p(w)}. \qquad (11)$$

3.3 Definition of pronunciation entropy

In order to have a measure of the confusability of the pronunciation lexicon, the entropy of the posterior probability $p(w \mid a)$ that combines the pronunciation model and the language model information is computed. As described in Section 2.2, calculating appropriately the shortest distance on the entropy semiring can result in the desired entropy. However, the entropy semiring must have components in the real semiring in order to calculate the entropy correctly, but even real numbers of double precision are not stable enough for large lattices. Thus, an expectation semiring with components on the log semiring is needed. That is why we define a new semiring, the log-expectation (or log-entropy) semiring, changing the $\oplus$ and $\otimes$ operators as well as the identities of the semiring. In this new semiring $(\mathbb{K}, \oplus, \otimes, (\infty, 0), (0, 0))$, $\mathbb{K}$ denotes $(\mathbb{R} \cup [-\infty, \infty]) \times (\mathbb{R} \cup [-\infty, \infty])$ and the operations $\oplus$ and $\otimes$ on weight pairs (a_1, b_1) and (a_2, b_2) in $\mathbb{K}$, are defined as:

$$(a_1, b_1) \oplus (a_2, b_2) = (-log(exp(-a_1) + exp(-a_2)), b_1 + b_2) \qquad (12)$$

$$(a_1, b_1) \otimes (a_2, b_2) = (a_1 + a_2, exp(-a_1)b_2 + exp(-a_2)b_1) \qquad (13)$$

When working on the log-entropy semiring, each negative log arc weight w is replaced by the new weight $(w, w * exp(-w))$. Then, the shortest distance from the initial to the final state is calculated as explained in Section 2.2. Some experiments were realised on small exemplar lattices with real arc weights and the entropy was calculated directly with the entropy semiring, already defined in OpenFst. However, for larger lattices the use of the log and the log-entropy semirings was required in order to keep the numerical stability.

4 Phoneme Recognition Configuration

The phoneme recognizer used in these experiments makes use of continuous density HMMs with Gaussian mixtures for acoustic modeling. The acoustic models are gender-dependent, speaker-adapted, and Maximum Likelihood trained on about 500 hours of audio data. They cover about 30k phone contexts with 11600 tied states. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR and MLLR techniques prior to decoding. It suffices to say that the phone labels that are produced at that stage are deterministically mapped to the corresponding phonemes, which constitute the actual labels in the phoneme lattice. The recognition dictionary is a simple lexicon made up of the same list of phonemes used to represent pronunciations in the word lexicon. A unigram phoneme-based language model was also constructed to respond to the demands of the system for language modeling, but its weight was set to zero during the decoding phase. A phoneme lattice is thus generated after a single decoding pass, with no pronunciation model nor language model information included. The lattices are pruned so as to limit them to a reasonable size. To circumvent the fact that a lattice does not always finish with an end-of-phrase symbol, which can be the case because of

segmentation based on time, an end-of-phrase symbol is added before the final state of each lattice.

The FST approach described in Section 3 is applied for word decoding. A 4-gram word LM is used, trained on a corpus of 1.2 billion words of texts from various LDC corpora (English Gigaword, Broadcast News (BN) transcriptions, commercial transcripts), news articles downloaded from the web, and assorted audio transcriptions. The recognition word list contains 78k words, selected by interpolation of unigram LMs trained on different text subsets as to minimize the out-of-vocabulary (OOV) rate on set of development texts. The recognition dictionary used as a baseline is the LIMSI American English recognition dictionary with 78k word entries with 1.2 pronunciations per word. The pronunciations are represented using a set of 45 phonemes (Lamel & Adda, 1996). This dictionary is constructed with extensive manual supervision to be well-suited to the needs of an ASR system. Other dictionaries with and without counts and variants were also tested, as described in the next section.

A part of the Quaero (www.quaero.org) 2010 development data was used in the recognition experiments. This data set covers a range of styles, from broadcast news (BN) to talk shows. Roughly 50% of the data can be classed as BN and 50% broadcast conversation (BC). These data are considerably more difficult than pure BN data. The part of the Quaero data that was used resulted in 285 lattices generated by the phoneme recognizer. This is a sufficient number of lattices to have statistically significant results that can be generalized. The FST-based decoding is applied to these lattices. Table 1 summarizes some of the characteristics of the lattices and FSTs used in the composition process: the average number of states and arcs of the lattices A and of the phonemes-to-pronunciations FST P, of the pronunciations-to-words FST L and of the 4-gram word LM FSA G. Their size indicates that we are working in a real ASR framework with FSMs of large size. Thus, there is an important computational gain by the fact that the FST approach permits to change a part of the decoding without repeating the whole process.

Table 1: *Average number of states and arcs in the lattices*

Num.	A	P	L	G
States	303	180,833	2	83,367,599
Arcs	353	503,234	171,272	200,322,203

5 Pronunciation Entropy Results

The presented pronunciation entropy is an average of the entropy calculated on the FSAs of the word sequences generated after the application of the FST decoding on the output lattices of the phoneme recognizer. The pronunciation entropy is calculated for the baseline dictionary with and without frequency of occurrence counts, as well as for the "longest" baseline (keeping only the longest pronunciation per word in the original recognition dictionary) and the "most frequent" baseline (keeping only the most frequent pronunciation per word in the original recognition dictionary based on counts collected on the training data). The higher order language model (LM) used in the decoding of the word recognition experiments is a 4-gram.

In Table 2, the pronunciation entropy is presented when 2-, 3- and 4-gram LMs are used for the FST-based decoding. As expected, as the order of the LM diminishes, the entropy increases. The results when the order of the LM diminishes warrant some more thoughts. The difference in entropy even between the use of 4-gram and 2-gram is smaller than expected. The decoding is actually restricted by the given lexicon, that does not permit pronunciations, and thus words, to be correctly recognized if there is an error in the phoneme sequence. Deletions, insertions and substitutions of phonemes are ignored, with the exception of insertions between valid pronunciations. By manual observation of the best word hypotheses and their comparison with the corresponding references, it was thus noticed that not many long sequences of words are correctly recognized and, consequently, the impact of using a longer n-gram is relatively limited.

It is worth staying a bit longer in Table 2 to compare the pronunciation entropy of the baselines which contain one or more pronunciations per word (upper part of the table) and the single-pronunciation baselines (lower part of the table). The entropy is lower in the single pronunciation baselines, and its lowest score is observed when the "longest" baseline is used. The fact that its entropy is lower even

Table 2: *Pronunciation entropy on baselines*

4g LM	**3g LM**	**2g LM**
Baseline with uniform probabilities		
4.003	4.025	4.025
Baseline with counts		
4.065	4.083	4.108
Baseline longest with uniform probs		
3.013	3.024	3.022
Baseline mostfreq with uniform probs		
3.669	3.689	3.756

compared with the "most frequent" baseline, which is also a single-pronunciation baseline, may be explained by the fact that the most frequent pronunciations represent better the spoken terms that can be often easily confused. Especially in spontaneous speech, some function words are often pronounced similar to other function words and may not be easily distinguished by the LM. This is particularly a problem for frequent words that are easy to insert, delete or substitute.

A final observation from Table 2, comparing "Baseline with uniform probabilities" and "Baseline with counts", can be that pronunciations with counts do not reduce confusability. It is normal not to see a lot of changes because in any case the majority of words has only one pronunciation and thus probability 1 which do not change when counts are added. In addition, counts are not available for all words, but only for those observed in the training data. When no counts are available, uniform probabilities are applied. Thus, finally there are no great differences between the dictionary with counts and the dictionary without counts. Moreover, it could be that counts only for a few words create an inconsistency that explains the light deterioration of the pronunciation entropy.

The increase in entropy is much greater when more pronunciations are added in the dictionary as can be seen in Tables 3 and 4. The n-best pronunciations are added in the "longest" and the "most frequent" baselines. The M1, M2 and M5 in these tables correspond to the 1-, 2- and 5-best pronunciations generated automatically using Moses (Koehn *et al*, 2007) as a g2p converter, being trained on the baseline dictionary (with 1.2 pronunciations per word). Moses has been successfully used as a g2p converter for several languages, and for English it gives state-of-the-art results (Karanasou & Lamel, 2011). The results in Tables 3 and 4 are calculated with the 4-gram LM.

Table 3: *Pronunciation entropy with the 4-gram LM after adding n-best pronunciations, produced by a Moses-based g2p converter, to the "longest" baseline*

Training condition	M1	M2	M5
Multiple pronunciations	4.523	6.578	10.005
Baseline longest	3.013		

Table 4: *Pronunciation entropy with the 4-gram LM after adding Moses' n-best pronunciations to the "most frequent" baseline*

Training condition	M1	M2	M5
Multiple pronunciations	5.185	6.914	10.077
Baseline most frequent	3.669		

These results suggest that there can be a large influence of the pronunciation dictionary in the confusability of an ASR system, not sufficiently compensated by the language model. However, when adding as much alternative pronunciations some non-uniform probabilities should be used to moderate confusability. If not, the uniform probability contributed to each variant of a word with multiple pronunciations is lower. Thus, for highly probable words, since the system will have the tendency to choose them, the confusability will increase. But if the pronunciation probabilities are also taken into account, this confusability can be moderated, because a pronunciation of a word with lower probability and lower confusability (higher pronunciation probability) can be prefered from a pronunciation of a word with higher probability but lower pronunciation probability. We have started working on this direction and plan to see if our measure will actually improve when pronunciation probabilities are added to the decoding.

More confusability is observed when adding variants to the "most frequent" than to the "longest" baseline. This is consistent with the explanation given above supporting that the most frequent baseline presented more confusability than the longest baseline because it is closer to the real spoken terms

that are often difficult to distinguish.

6 Conclusion and Discussion

A new measure of the confusability of the pronunciation model during the decoding phase in an ASR system, that integrates also language model information, was presented and results were reported using baseline dictionaries with one or more pronunciations per word, with and without counts, as well as on dictionaries extended with variants generated by a state-of-art data-driven method.

It is not straightforward to find a correlation between this work and ASR performance. The follow-up of this work will be to examine this correlation and propose a combined measure of confusability and accuracy for the selection of pronunciation variants and for the training of weights for the existing ones. What makes this procedure particularly complicated is the fact that confusable words are a non-negligible phenomenon of natural speech and ignoring them severely reduces the completeness of the dictionary, meaning that a consistent set of pronunciations is not necessarily connected with a pronunciation network of low perplexity.

A drawback of the work so far is that sequences obtained from the phoneme recognizer contain many errors. To avoid the problems caused by the low performance of the phoneme recognizer, some phoneme substitutions should be permitted. For this, a confusion matrix and a consensus representation could be useful.

Lastly, the pronunciation entropy introduced in this work is a measure of the confusability at the sentence level. It would be interesting to try and measure confusability at a sub-sentence level, such as at a word level using confusion networks, as the error rate of an ASR system is also calculated at a word level. At a more general sub-sentence level, some word-context could be taken into account to better model the cross-word confusability, because when adding many variants to an ASR system what is more important than the homophone rate in the dictionary, is to measure this rate in the data. In fact the homophone rate in the original recognition dictionary (baseline) is 1.16, while in the baseline "longest" is 1.10 and in the baseline "most frequent" is 1.15. When adding up to 5 pronunciation variants to the baseline "longest" or the baseline "most frequent", the homophone rate becomes 1.24 in both cases. All these rates are close to one another, so it seems that what mostly influences confusability are some frequent homophone words or word sequences.

Acknowledgments

This work is partly realized as part of the Quaero Programme, funded by OSEO, the French State agency for innovation and by the ANR EdyLex project. The authors wish to thank T. Lavergne for sharing his expertise on the OpenFst library.

References

Allauzen,C., Mohri, M. and Roark, B.. 2003. Generalized algorithms for constructing statistical language models. In *Proceedings of ACL*, volume 1, pages 40-47.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W. and Mohri, M.. 2007. OpenFst: a general and efficient weighted finite-state transducer library. In *Proceedings of CIAA*, pages 11-23.

Cortes, C., Mohri, M., Rastogi, A. and Riley, M. D.. 2006. Efficient Computation of the Relative Entropy of Probabilistic Automata. In *Proceedings of LATIN*, volume 3887, pages 323-336.

Cover, T.M. and Thomas, J. A.. 1991. Elements of Information Theory. John Wiley & Sons, inc., New York.

Eisner, J.. 2001. Expectation Semiring: Flexible EM for Learning Finite-State Transducers. In *Proceedings of FSMNLP*.

Fosler-Lussier, E., Amdal, I. and Kuo, H.-K. J. 2002. On the road to improved lexical confusability metrics. In *Proceedings of PMLA*, pages 53-58.

Karanasou, P. and Lamel, L.. 2011. Pronunciation Variants Generation Using SMT-inspired Approaches. In *Proceedings of ICASSP*.

Kipp, A., Weswnick, M.-B. and Schiel, F.. 1997. Pronunciation modeling applied to automatic segmentation of spontaneous speech. In *Proceedings of Eurospeech*, pages 1023-1026.

Koehn, P., Hoang H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E. 2007. Moses:Open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177-180.

Lamel, L. and Adda, G.. 1996. On designing pronunciation lexicons for large vocabulary continuous speech recognition. In *Proceedings of ICSLP*.

Mohri, M., Riley, M. and Sproat, R.. 1997. Finite-state transducers in language and speech processing. In *Computational Linguistics*, volume 23-2, pages 269-311.

Pereira, F.C.N. and Riley, M.D.. 1996. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing*, pages 431-453.

Printz, H. and Olsen, P.. 2000. Theory and practice of acoustic confusability. In *Proceedings of ISCA ITRW ASR*, pages 77-84.

Riccardi, G., Pieraccini, R. and Bocchieri, E.. 1996. Stochastic automata for language modeling. In *Computer Speech and Language*, volume 10, pages 265-293.

Strik, H. and Cucchiarini, C.. 1999. Modeling pronunciation variation for ASR: A survey of the literature. In *Speech Communication*, volume 29, pages 225-246.

Tsai, M., Chou, F. and Lee, L.. 2001. Improved pronunciation modeling by inverse word frequency and pronunciation entropy. In *Proceedings of ASRU*, pages 53-56.

Wolff, M., Eichner, M. and Hoffmann, R.. 2002. Measuring the quality of pronunciation dictionaries. In *Proceedings of PMLA*, pages 117-122.