THE MEASUREMENT OF ACOUSTIC SIMILARITY AND ITS APPLICATIONS

Liqin SHEN, Guokang FU, Haixin CHAI, Yong QIN

Speech Department, IBM China Research Lab, Beijing, China
E-mail: shenlq@cn.ibm.com

ABSTRACT

It's always helpful if we can predict how well a recognition system can perform for different tasks and evaluate the quality of different acoustic models. This paper presents a way to compute the acoustic similarity between words of a recognition task from the statistic models directly based on 2 kinds of definitions of the acoustic distance between recogniton units. It's applications for command and control task complexity prediction and acoustic model evaluation are discussed. The experimental results shows it's a useful measurement.

1. INTRODUCTION

For dictation systems, perplexity is widely used to measure the difficulty of a recognition task[1]. However it's well known that lower perplxity does not definitely result in higher recognition accuracy. Because it only concerns about the Language Model(LM) factor. But a recogntion task depends on both LM and Acoustic Model(AM). As we know, a recognition task is to find

$$w^* = \arg\max_w P(w \mid o) = \arg\max_w P(o \mid w)P(w)$$

P(w) is usually measured by the perlexity. When we use perplxity to measure the difficulty of a recognition task, we are presuming that $P(\mathbf{o}|w)$ for different *w* are the same. It means different *w* sound the same, which is not true. It can be seen that if there are *n* word sequences, w_1, w_2, ... w_n with $P(w_1)=P(w_2)=...=P(w_n)$, if $P(\mathbf{o}|w_1)$ is much different from other $P(\mathbf{o}|w_i)$, which means w_1 is acoustically far from other word sequences, it's easy to get w_1 as the top output when we dictate it. On the contrary, if $P(\mathbf{o}|w_1)$ is close to some other $P(\mathbf{o}|w_i)$, which means they sound similar, w_1 may be easily mis-recognized as other word sequences.

Interested readers can find Speech Decoder Entropy[2], Acoustic Perplexity[3] etc. different measurements which take into account of AM and LM interaction. In this paper, we don't focus on solving this problem. We are more interested in the measurement of the acoustic similarity. This is very useful in the study of the complexity of a command and control recognition task, where *P(w)* is practically uniform for any word, and we just need to conern the acoustic confusibility between the words. This could be also a good measurement for AMs without the requirement of test speech being recorded, as perplexity is used to measure the LM quality given a defined task . Let's say $P(o \mid w_{ix})$ is the likelihood of the speech observation sequenc **o** given word *i* with AM *x*, $P(o \mid w_{iy})$ is that with AM *y*. Here obviously the comparison of AM *x* and *y* is valid only when they are built from the homogeneous training data, but by different training parameters and strategy, and used for the same homogeneous test set.

Now the problem is given ***o,*** we try to compare $P(\mathbf{o}|w_i)$ and $P(\mathbf{o}|w_j)$. We can interprate w_i and w_j as 2 words or the same word but from different models. In [3], by using synthesizing observation sequences correponding to different words, computational results of $P(\mathbf{o}|w_i)$ and $P(\mathbf{o}|w_j)$ can be achieved. In this paper, we will compute the acoustic distance of 2 words directly from their probabilistic models.

In the next section, we will define the acoustic similary of 2 words. Then we will discuss its applications in task complexity prediction and acoustic model evaluation in section 3. Experimental results and some discussions are given in section 4. At last, potential future work are brought

2. THE MEASUREMENT OF ACOUSTIC SIMILARITY

In recognition systems, Hidden Markov Model(HMM) for a word is normally concatenated by HMMs of its acoustic unit sequence. Let's denote

$w_i=u1u2...u_n$; u_i is an acoustic unit of w_i

$w_j=v1v2...v_m$; v_j is an acoustic unit of w_j

Acoutic unit can be phone, sub-syllable or syllable etc. Let's define the distance of 2 acoutsic units：$AcDis(u_i, u_j)$, first.

2.1 The Acoustic Distance Of 2 Acoustic Units

Different definitions of the model distances between recognition units can be used such as that used in speaker recognition technology[4].

We proposed layer based distance between phone units for context dependent acoustic models, where the disctance between the centralized distribution prots at the lowest level, to the context dependent phone units till at the top level the phone units are defined[5]. Here is a brief review of such definitions.

Context dependent acoustic models can be described as a layer based structure:

PU(Phone Unit): $CDPU_1$, $CDPU_2$, …, $CDPU_p$

CDPU(Context Dependent PU): $c_1Prot_1+c_2Prot_2+..+c_kProt_k$

Prot(Prototyes): any centralized distributions

Let' s denote 2 phone units as:

$$PU_1=\{ CDPU_1^i \}, i=1, ..., M$$
$$PU_2=\{ CDPU_2^j \}, j=1, ..., N$$

And the symmetric distance between PU_1 and PU_2 is

$$D(PU_1,PU_2)=0.5(D(PU_1/PU_2)+ D(PU_2/PU_1)$$

$D(PU_2/PU_1)$ is a directional distance from PU_1 to PU_2 and is defined as

$$D(PU_2/PU_1)=\sum_i Prob(CDPU_1^i/PU_1)\bullet D(PU_2/CDPU_1^i)$$

$Prob(CDPU_1^i/PU_1)$ is the probability of the occurance of $CDPU_1^i$ while PU_1 is present, i.e.

$$Prob(CDPU_1^i/PU_1)=\frac{\sum \text{counts of Prots in } CDPU_1^i}{\sum \text{counts of Prots in } PU_1}$$

$D(PU_2/CDPU_1^i)$ is the distance of the *i*th *CDPU* in PU_1 to PU_2 and is defined as the weighted summation of the distance of $CDPU_1^i$ to each *CDPU* in PU_2, i.e.

$$D(PU_2/CDPU_1^i)=\sum_j Prob(CDPU_2^j/PU_2)\cdot D(CDPU_2^j/CDPU_1^i)$$

$D(CDPU_2^j/CDPU_1^i)$ can be derived from the distances from prototypes, i.e.

$$D(CDPU_2^j/CDPU_1^i)=\sum_{n=1,...,N}(k_1^i)^n\sum_{m=1,...,M}(k_2^j)^m D((Prot_2^j)^m,(Prot_1^i)^n)$$

N is the number of prototypes of $CDPU_1^i$, M is that of $CDPU_2^j$. $(k_1^i)^n$ is the weight of the *n*th prototypes in $CDPU_1^i$, similar for $(k_2^j)^m$. And the likelihood distance can be used for the distance between the centralizing distributed prototypes, such as Gaussian ones, for $D((Prot_2^j)^m,(Prot_1^i)^n)$

For Chinese, as the language is syllable based, we can use syllable as an acoutic unit for *AcDis(u_i, u_j)* computing. With a large amount of AM training data, we use syllables as recognition words and remove all LM effects when decoding. Then we trained a syllable-based confusion matrix from these decoded training data. The matrix contains the count of each syllable u_i being mis-recognized as another syllable u_j, which is *cfscount(u_i, u_j)*, and the tatal number of syllable u_i in the training data *total(u_i)*. From the matrix, the probability of one syllable being mis-recognized as another one, *cfs(u_i, u_j)*, can be computed as:

$$cfs(u_i, u_j)=\frac{cfscount(u_i,u_j)}{tatal(u_i)}.$$

And we define *AcDis(u_i, u_j) = 1- conf(u_i, u_j)*. The details can be found in [6].

2.2 The Acoustic Distance Of 2 Words

We define the acoustic distance of 2 words *AcDis(w_i, w_j)* as

$$AcDis(w_i,w_j)=\frac{InsDis(w_i,w_j)+DelDis(w_i,w_j)+SubDis(w_i,w_j)}{MaxDis(w_i,w_j)} \quad (1)$$

where *InsDis(.)*, *DelDis(.)* and *SubDis(.)* are insertion, deletion and substitution distances for acoustic units separately. The substitution distance of an acoustic unit pair can be obtained with either method described in 2.1. For insertion and deletion distances, let' s see an example for futher consideration: now we have 3 words donated as

$$w_1=u_1u_2,\ w_2=u_1'u_2',\ w_3=u_1''u_2''$$

If *AcDis(u1,u1')* and *AcDis(u1, u1'')* are equal to 1, and it' s the same for *AcDis(u2,u2')* and *AcDis(u2, u2'')*, while *AcDis(u2, u1')* is close to 0, which is not the case for *AcDis(u2, u1'')*. Most likely w_1 has more chance to be confused with w_2 rather than w_3, especially when u_1 and u_2' sound more like silence. But if we just simply set the insertion and deletion distance as 1, any insertion or deletion will result in larger word distance than substitutions. Then we may get *AcDis(w_1,w_2) = AcDis(w_1, w_3)*

So Instead of setting the insertion and deletion distances to 1, we define them as the substitution distance of that inserted or deleted phone with the silence phone. i.e.

$$InsDis(u_i) = SubDis(u_i, sil\_u)$$

$$DelDis(u_d) = SubDis(u_d, sil\_u)$$

where u_i is the inserted phone, u_d is a deleted phone, *sil_u* is the phone representing silence.

With the minimum word distance as the target, we use Viterbi algorithm[7] to find the best alignment of the phone units between 2 words.

3. THE APPLICATIONS OF ACOUSTIC SIMILARITY

There are many cases when we have a new recognition task, we are not sure how better a recognition system can perform and if it can meet the requirement. The size of the task is a important hint. However the same size of tasks may result in much different performane. For example in Chinese, the recognition rate for 1000 stock names are much better than that for the same ammount of people's names. It turns out that another key factor is the acoutic similarity among the words in a recognition task. Now we will derive a computational measurement of the recognition complexity and use it to predict the performance of a recognition task.

3.1 Prediction Of A Command and Control Recognition Task Complexity

Given an AM, for a recognition task *T* with *N* as its vocabulary size, let's denote *V* as the task vocabulary, and $w_1,..., w_N \in V$. It's quite natural to define the recognition complexity as:

$$Comp(T) = 1 - \frac{1}{N(N-1)/2} \sum_{\substack{i=1,..N \\ j=i+1,...N}} AcDis(w_i, w_j)$$

With this definition, you may find that the complexity for many recognition tasks are quite even. Actually, in a recognition task, no matter how a word is acoustically far away from others, as long as there is one word sounds closely to it, it can be easily decoded wrong into that word. So it may be more reasonable to define

$$Comp(T) = 1 - \frac{1}{N} \sum_{w_i \in V} \min_{\substack{w_j \in V \\ w_j \neq w_i}} (AcDis(w_i, w_j))$$

However w_i is not necessarily mis-recognized into its closest word in the real case for many random factors. To be robust, *m* closest words are considered rather than only the closest one, and we chose *m* as the same value of the fast match short list which is widely used in recognition systems.

$$Comp(T) = 1 - \frac{1}{N} \sum_{w_i \in V} (\frac{1}{m} \sum_{\substack{w_j \in SL\_V(w_i) \\ w_j \neq w_i}} AcDis(w_i, w_j)) \quad (2)$$

where $SL\_V(w_i)$ is the set of the *m* closeset words to w_i.

(2) is still debatable, for *m* is the upper boundary of the fast match short list. For those which are within the $SL\_V(w_i)$ but still acoustically very far from w_i, they don't contribute to confuse the correct recognition but disturb *Comp(T)* somehow. A threshold q can be applied to quantize those $AcDis(w_i, w_j)$ which are larger than q to 1.

The error rate should be dependent on the *Comp(T)*, and irrelavant to the task size ideally. The fact of larger task size normally results in higher error rate is because the larger a task size is, the more possibility to have more confused words in the short list, and get higher task complexity *Comp(T)*. With a good amount of test sets, we can derive a curve of the function of the recognition error rate vs. the task complexity.

3.2 Evaluation Of The Quality Of Acoustic Models(AM)

When we have different LMs, we usually use the perplexities of a given test corpus to measure which one is better. To measure AM is much more complicated, because it depends on what training data are used and what a recognition task is for etc. A better AM for a desktop task is obviously not necessarily better than its inferior for a telephony task. However we can still use *Comp(T)* defined in (2) to guide our AM training process, where the training and target task data are from the homogeneous channels.

Many parameters could be tuned in AM training process. To test these AMs obtained with these different parameters, a large ammount of recordings need to be collected and decoding will be performed on them. Actually we can compute *Comp(T)* for a given vocabulary, which has a good acoustic unit coverage, with these different models. Different models will have different $AcDis(u_i, u_j)$ thus different $AcDis(w_i,w_j)$ for the same pair of (w_i,w_j). The less *Comp(T)* for a AM, the better this AM.

4. EXPERIMENTS AND DISCUSSIONS

More focus are put into the relationship between the task complexity vs. the recognition error rate when we perform experiments.

In real case, we presume a task vocabulary(*V*) is the same as the test one(let's denoted as *T_V*), because all the words in the task have the equal chance to be used by users. But in expriments, we don't have enough recordings if we want to test how the task complexity is related to the error rate for a large

task. So we designed 3 different task sizes, while used the same test recordings for our test.

The recordings of T_V is a subset of V. Here we chose 10 speakers, 15 utterances per speaker. 150 words in these utterances. And the 3 different tasks(V) are:

V_A: 300 words

V_B: 1000 words

V_C: 3000 words

It can be seen that besides 150 test words which belong to T_V, no matter how other words (V - T_V) are confused with each other, they will not influence the error rate given the defined test set. So $w_i \in V$ in (2) need to be modified as $w_i \in T\_V$ when $T\_V \neq$ V. In order to make sure we cover most task complexity range during the test, for each word in T_V, we chose different words with different distances from large to small to that word to compose tasks(i.e. V) with different complexities.

Let's investigate the exprimental results for these 3 tasks.

Table 1-3 and corresponding Fig. 1-3 are the numeric results and plots of the experiments repectively.

Note that for all Tables below, *Comp(T)*=Comp(T) * 1000*

Table 1. Experimental results for task V_A.

Comp(T)*	96	96.8	97.4	101.3	130.1
Errors	0	0	2	2	4

Table 2. Experimental results for task V_B

Comp(T)*	120.9	164.5	170	171.1	174.6	185.2	231.3
Errors	0	3	3	5	8	9	16

Table 3. Experimental results for task V_C

Comp(T)*	214.5	293.9	296.2	304.5	310.5	348.9
Errors	2	4	5	7	11	18

Fig. 1 Task Complexity vs. Recognition Errors for VA

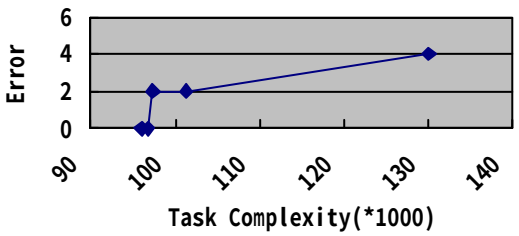


Fig. 2 Task Complexity vs. Recognition Errors for VB

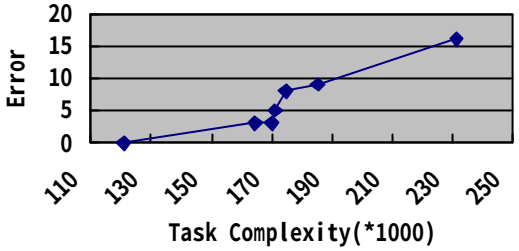
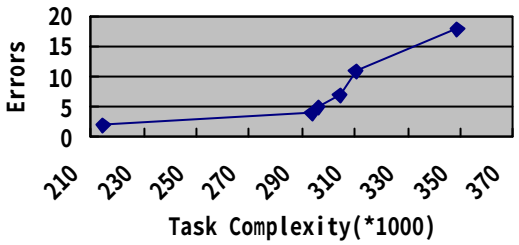


Fig. 3 Task Complexity vs. Recognition Errors for VC



It can be seen that for the same task, the error rate goes up almost consistently with the task complexity. As we analyzed in the 3rd section, this trend proves the definition of the task complexity is reasonable.

However, we didn't get the conclusion that the error rate is related to the task complexity and irrelevant to the task size. The reason is very complicated. Actually the error rate is very related to the test recordings we are using. For a word, even there are a lot of confusing words in the task, if the recording for this word is perfectly spoken by a speaker which has a good match to the AM, it could be recognized correctly. For another word which is on the contratry, it could be recognized wrong if it is spoken by a bad speaker. So the task complexity will not have the direct relationship to the error rate in different tasks when the acoustic distances for the test words have different distributions in different tasks. However the same distance distribution for the test words is not guaranteed in the horizontal comparisons among tasks. While within the same task, we chose out-of-test words for each in-test word with the same acoustic distance trend, so the vertical comparison is consistent for each word, thus for different task complexities, and we got what can prove our definitions.

In summary, to get a statistically correct conclusion, we need a large amount of test data rather than every word in T_V is spoken once by one speaker, same as the real case where each word could be spoken by many speakers. But the vertical comparisons prove the task complexity definition based on the acoustic similarity can be a quantitative meausrement

5. FUTURE WORK

For Chinese, the word definition is very flexible. If we can use the acoustic similarity togather with LM statistics, we can improve the cases where LM statistics are similar while the acoustic similarity is close also, which are the culprits of most of the errors, by modifying the vocabulary in a task. In this way we can optimize a recognition vocabulary.

6. REFERENCES

[1] Jelinek F. Statistical methods for speech recognition, *The MIT Press*, 1997.

[2] Ferretti M., Maltese G., and Stefano S.. Measuring information provided by language model and acoustic model in probabilistic speech recognition: Theory and experimental results. *Speech Communication*, 9:531-539, 1990.

[3] Harry P., Olsen P. Theory of acoustic confusibility. *To be appearred in ASR2000*, Paris, France, Sept. 2000.

[4] Homayoon S. etc., A distance measure between collections of distributions and its applications to speaker recognition. *ICASSP' 98*, pp.753-756

[5] Fu G., Shen L., Model Distance and it' s application on mixed language speech recognition system. *To be appearred in ISCSLP' 2000*, Beijing, China, Oct. 2000

[6] Shen L., Qin Y., Chai H., Tang D., *Character error correction for Chinese speech recognition system.* ISCSLP' 98, Singapore.

[7] Rabiner L., Juang B., Fundamentals of speech recognition, *Prentice Hall*, 1993.