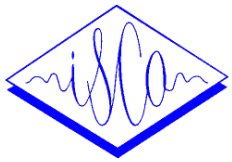

7th International Conference on Spoken Language Processing [ICSLP2002]
Denver, Colorado, USA
September 16-20, 2002

A STATISTICALLY MOTIVATED DATABASE PRUNING TECHNIQUE FOR UNIT SELECTION SYNTHESIS

Peter Rutten, Matthew Aylett, Justin Fackrell, Paul Taylor

Rhetorical Systems
Edinburgh, U.K.
www.rhetorical.com

ABSTRACT

An important topic in unit selection based speech synthesis is the scalability of such systems. Related to this problem is the question regarding the optimal size of a unit selection database. An ideal system should produce ever better synthesis results when more data is added to the system, but for a practical system this might not be the case. The unit selection criteria are generally not sufficiently developed to ensure that a system makes an optimal use of the data that it has available.

In this paper we propose a database reduction technique based on the statistical behaviour of unit selection. We investigate the effect of scaling down the database by objective and subjective criteria. We compare the proposed reduction technique with a technique that simply limits the size of unit lists to a fraction of their original size (random removal).

The results show that the proposed technique is far better than random removal, and that we can remove a significant portion of our database without causing any severe quality loss.

1. INTRODUCTION

Unit selection based speech synthesis systems succeed in realising near-natural speech quality by exploiting a large resource of carefully annotated speech. Since very little is known about the optimal design for these databases, we simply add as much data as possible.

For scalability, however, we know we can no longer rely on optimal data representations or speech compression algorithms alone to fit the system onto every platform we would like. Instead, for some applications we will need to scale down the system by removing data from the database.

Two strategies come to mind to tackle the problem of database pruning: a bottom-up and a top-down approach.

1.1. Bottom-up approach

In a bottom-up approach we try to spot spurious or redundant data purely by investigating the database itself. A typical algorithm (e.g. [2], [4]) will start by clustering speech units according to some similarity measure. From these clusters a reduced database can be constructed by gradually adding or removing units until the desired database size is reached.

A possible weakness of this approach is that it is not necessarily very closely tied to the unit selection algorithm. Similarity measures used for clustering will generally use more acoustic features than are available during unit selection. It is thus likely that similarities spotted by the reduction algorithm will be meaningless to the unit selection algorithm.

Originally we tried to synchronise the optimization criteria that are used in database reduction and in unit selection, but we soon realized that a top-down approach would be better suited for this.

1.2. Top-down approach

The top-down approach is based on the investigation of the output of the synthesizer. Such an approach has been described in [1] for the implementation of synthesis-based preselection, and in [2] for weighted vector quantization based pruning. We take it a step further by using the statistics directly to do database reduction.

This approach has the advantage that no knowledge about speech units - apart from the statistics relating to their use - is needed to reduce databases. It is further described in the next section.

2. DATABASE PRUNING BASED ON THE STATISTICAL BEHAVIOUR OF UNIT SELECTION

In the experimental system described here, the elementary speech unit is a diphone. For each diphone we have a large number of realizations in the database, i.e. diphone variants. A diphone variant list consists of all the realizations of a particular diphone in the database.

Database pruning takes two steps: first we collect statistics on the use of diphone variants, then we use the statistics to prune the variant lists.

2.1. Collecting statistics for database reduction

For our experiments we used a large database of one of our UK English speakers. We will refer to this database as the

100% database or DB_{100}.

To collect statistics on the use of diphones by the system, we synthesize a corpus of 242,793 text files (news items collected from web pages), containing 19,990,272 diphone occurences. We will refer to this corpus as C_T.

Our synthesizer converts C_T into the synthesized corpus C_S, and during synthesis we keep track of the use of diphone variants. As a result, we obtain the frequency of use for each diphone variant in DB_{100} for the synthesis of C_T.

Fig. 1 shows how the use of diphone variants tends to flatten off as the amount of material spoken increases. Eventually we used 89.97 % of the diphone variants in the database to create C_S.

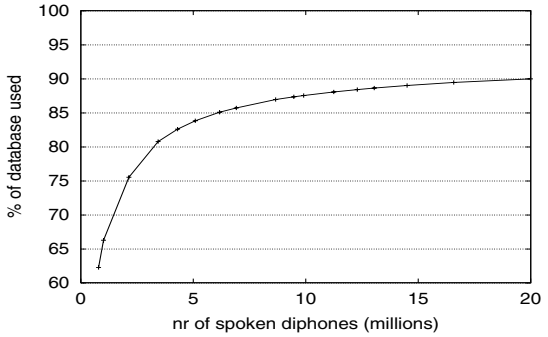

Fig. 1. Usage of database with increasing number of spoken diphones

The makeup of C_T (news items in our case) will have an influence on the statistics that will be collected. However, informal comparisons between the effect of database reduction on different genres of text, do not seem to produce different results.

2.2. Pruning algorithm

When we prune the database, we try to achieve two things:

1. keep as much as possible of C_S covered by the remaining diphone variants. The motivation behind this is that we want the reduction to have an as small as possible effect on synthesis,

2. maintain the balance of available variants for different diphones. Indeed, if we would simply remove the least used variants in the database, we would risk losing all variants for infrequently occuring diphones.

To keep the original balance of available variants for different diphones, we prune each diphone variant list individually. Within each list we remove the least used variants, that together contribute a given fraction of the occurence of the diphone in C_S.

Fig. 2 shows how we prune the variant list for the diphone /k-a/, which originally contains 216 variants. The figure shows the frequency of occurence of each variant, with the variants ordered according to this frequency. We see that the 66 least used variants account for only 5 % of the times this diphone was used in C_S. Or in other words, it is possible to reduce this variant list by 30.6 % (66/216) while keeping the coverage for this diphone in C_S equal to 95 %.

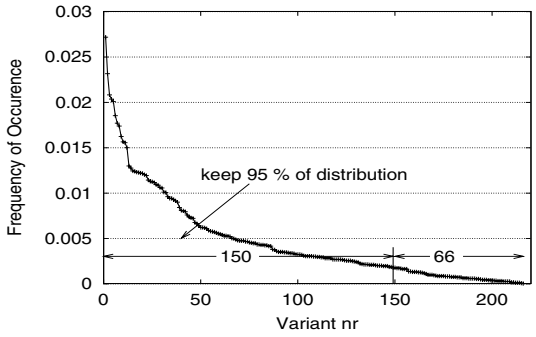

Fig. 2. Pruning of one variant list (diphone k-a)

If we call f_{cov} the fraction of C_S that is covered by the units in the reduced database, we will apply the procedure outlined above - removing $1-f_{cov}$ in the tail of the frequency distribution - to each variant list. The resulting reduction of the database depends on the shape of the individual distributions.

To find the relation between f_{cov} values and reduced database sizes, we have to apply the procedure for different values of f_{cov}. The results are shown in Fig. 3. From this plot we see, for example, that setting $f_{cov} = 92\%$ leads to a database reduction of 50 %.

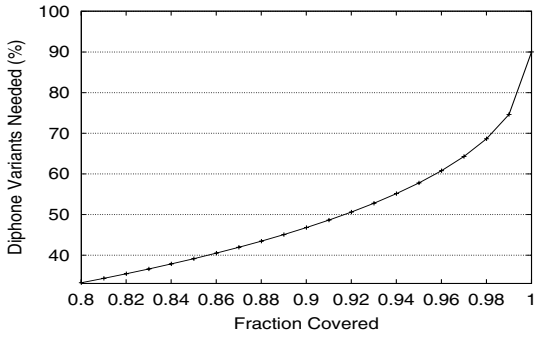

Fig. 3. Diphone variants needed to cover a given fraction of C_S

Some practical results of this procedure are evaluated in the following sections.

3. OBJECTIVE EVALUATION

To evaluate the proposed database reduction algorithm, we compare it with an approach where we limit the size of each

diphone list to a fraction of its original size, by randomly removing diphone variants. We will refer to the statistically motivated pruning technique as P_S, and the random pruning technique as P_R.

We create reduced databases by removing 25%, 50%, 75% and 87.5% of the diphone variants. We consider three objective evaluation criteria: synthesis overlap, database contiguity and voiced polyphone conservation.

3.1. Synthesis overlap

A first thing to look at is the effect of database reduction on the diphone selection, for a test corpus not included in C_T. Indeed, the fact that we aim at keeping f_{cov} of C_S covered does not mean that we will actually see this happening if we use the reduced database to synthesize C_T. Since the database has drastically changed, it is very likely that the overlap in chosen synthesis units will be a lot lower than what we specified in the reduction algorithm.

We have synthesized 100 sentences, extracted from the Timit database (not in C_T), and measured the overlap between reduced- and full-database synthesis. The results are shown in Fig. 4. It shows that P_S has a much smaller effect on the unit selection then P_R. E.g. 50% database reduction with P_R leads to 83% different units, and the same reduction achieved by P_S leads to only 38% different units.

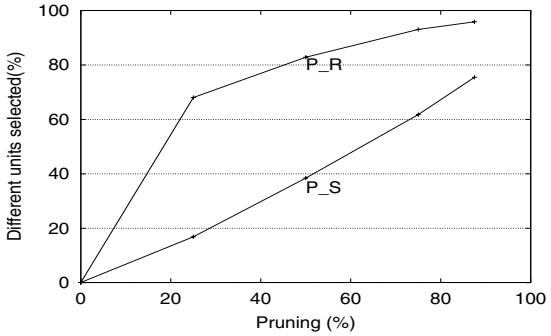

Fig. 4. Difference in chosen units between full and reduced database synthesis

3.2. Database contiguity

We can investigate the properties of the reduced database, in terms of the contiguity that is preserved when removing diphones. Originally, the database consists of complete utterances, so all diphones have a neighbour on both sides (except for the first and last diphone in each utterance). When we remove diphone variants from the database, this will no longer be true - the database will become discontiguous.

We assume that there is a link between contiguity of the database and synthesis quality: if the database is highly contiguous, it is more likely that the synthesizer can use polyphone chunks instead of isolated diphones. So the higher the contiguity, the better the database is.

We can quantify the contiguity of the database in terms of the number of contiguous diphones:

$contiguity = (n_{cont}/N_{cont}) * 100$

with n_{cont} the number of contiguous diphones, and N_{cont} the maximum number of contiguous diphones in the reduced database.

Fig. 5 shows the contiguity of pruned databases, for both pruning methods. It is clear that the P_S leads to a more contiguous database than P_R.

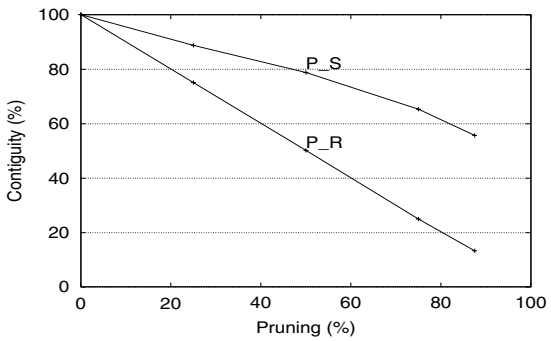

Fig. 5. Contiguity of pruned database

3.3. Conservation of voiced polyphones

Apart from the general contiguity, we can also look at the contiguity in terms of the number of available polyphones that start and end in a pause/fricative/stop. We expect that the output speech will sound smoother if the database contains a rich collection of such polyphones.

Fig. 6 shows the number of phonetically different polyphones of a certain length (measure by the number of diphones) that are available in the full database and in 50% pruned databases. P_S clearly conserves more of these chunks than P_R does.

4. SUBJECTIVE EVALUATION

By performing listening tests we can evaluate the effect of database reduction on synthesis quality.

In the design of the experiment we take into account that the synthesis quality, produced by a unit selection synthesizer, can be quite variable. In preliminary experiments we noticed that the quality of some sentences improved by reducing the database. We can explain this behaviour by assuming that there is a lot of noise inside the algorithms - and that this can cause the output to be better or worse whenever you make any change to the database.

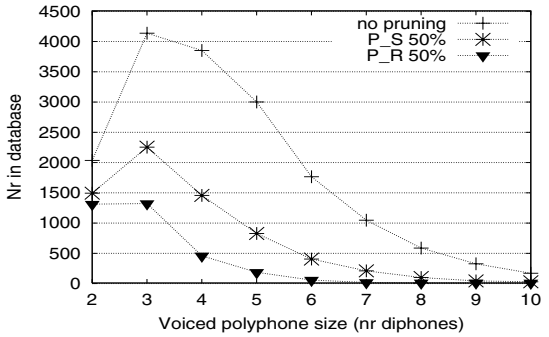

Fig. 6. Voiced polyphone distribution in full and pruned databases

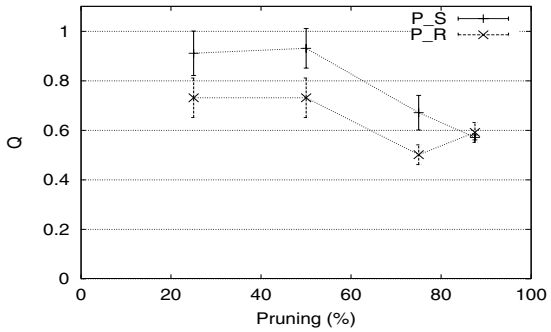

Fig. 7. Quality degradation due to database reduction

From a large set of short sentences (5 to 10 words long), we preselected 25 sentences that sound good, and 25 sentences that sound less good. This makes up a test set of 50 sentences that we presented in a blind comparative listening test, for different degrees of reduction, and for the two different reduction techniques. The listeners were asked to give a score (first better, second better, both equal) for each pair of utterances. The results are translated to a quality measure as follows:

$Q = 1 - (n_{full} - n_{red})/N$ with n_{full} = number of times full database version preferred, n_{red} = number of times reduced database version preferred, N = total number of sentences evaluated.

This would result in $Q = 0$ if the listeners always prefer the full database version, and $Q = 1$ if the listeners choose the full database version an equal number of times as the reduced database version, or if they showed no preference at all.

Four listeners performed each test, the results (average, and standard error for Q) are shown in Fig. 7.

In this figure we see that P_S systematically performs significantly better than P_R, up to a reduction of 87.5% where both methods lead to a similar quality. This result agrees with the objective measures we have used to compare the quality of the pruned databases.

For P_S the effect on the output speech quality is small up to at least 50% pruning. The P_R method already causes a very noticable quality drop if we prune 25% of the database.

Somewhere between 50 and 75% pruning, both methods show a drop in quality. This indicates that the critical database size is situated somewhere between these levels of pruning.

5. CONCLUSIONS

Database pruning based on the statistical behaviour of unit selection is a promising reduction technique.

It is simple, in a sense that no complex algorithm is needed to decide which diphone variants to keep and which ones to remove - which is not the case with bottom-up approaches to database reduction.

It is efficient, leading to a better quality than random list pruning, according to both objective and subjective criteria.

A further advantage is that database pruning is automatically coupled with the unit selection mechanism. Whenever improvements are made to unit selection, it is simply a matter of collecting new statistics on the use of the database in order to redo the pruning.

The results indicate that our unit selection synthesis system reaches a more or less stable quality level with a database that is much smaller than the one we are currently using. We can prune the database down to at least 50% of its original size before we notice a significant drop in the output speech quality.

6. REFERENCES

[1] Alistair Conkie, Mark C. Beutnagel, Ann K. Syrdal, Philip E. Brown. Preselection of candidate units in a unit selection-based text-to-speech synthesis system. ICSLP 2000, Beijing, 2000.

[2] S. Kim, Y. Lee, K. Hisose. Pruning of redundant synthesis instances based on weighted vector quantization. In Eurospeech 2001, volume 3, pages 2231-2234, 2001.

[3] A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In ICASSP-96, volume 1, pages 373-376, Atlanta, Georgia, 1996.

[4] Alan W. Black and Paul Taylor. Automatically clustering similar units for unit selection in speech synthesis. In Eurospeech 1997, volume 2, pages 601-604, Rhodes, Greece, 1997.