An Objective Measure for Assessment of the Concatenative TTS Segment Inventories

Robert Batůšek

Department of Information Technology
Masaryk University, Brno
xbatusek@fi.muni.cz

Abstract

In the paper we present a method for assessment of the segment inventories for concatenative text-to-speech synthesis. We argue that the overall comprehensibility of the synthesized speech depends on the length of the segments — longer segments imply more intelligible speech. The problem of minimum text cover by the given segment set is formulated in the paper as well as an algorithm finding the solution. Some improvements speeding up the algorithm are discussed in the rest of the paper.

1. Introduction

Nowadays text-to-speech synthesizers use a variety of segment types. Faster computers allows us to use larger databases of segments that can significantly improve the quality of the synthesized speech. One way how to do it is to record several instances of each segment (e.g. diphone) in different phonetic or prosodic environments. This approach is followed by e.g. [1] or [2]. Some other researchers, however, try to enrich the basic segment inventory by some longer segments (see e.g. [3]). They assume (and their experiments prove it) that longer segments will sound more naturally to human listeners.

Of course, segment length is not the only factor influencing the overall quality of the synthesized speech. We have to take into account for instance the difficulty of the segment join or the difficulty of the segment modification (see e.q. [1]). However, it seems that the synthesized speech intelligibility and naturalness at least depends on the average segment length.

2. Minimum text cover

2.1. Basic definitions

In this paper we will consider only the segments corresponding to sequences of phonemes. Thus, our segments are not comparable to any diphone- or triphone-based segments. Let us now define some basic notions used in the rest of the paper.

Let us have a finite set V (a phonetic alphabet). *Segment* is a finite sequence of symbols from the alphabet V:

$$s = v_1 v_2 \dots v_k \;\; v_i \in V \; \forall i = 1, 2, \dots, k, \; k \in N$$

The number k is called the *length* of the segment and we denote it by $l(s)$.

Segment inventory is a finite set of segments:

$$S = \{s^1, s^2, \dots, s^m\}$$

m is the size of the segment inventory.

Text	a b c d b c d b a b c b	ASL
Cover 1	a b|c d|b|c d|b|a b|c|b	1.5
Cover 2	a|b c|d|b c|d|b|a|b c|b	1.3

Table 1: Covering the sample text. The phonetic alphabet is $V = \{a, b, c, d\}$ and the inventory is $\{a, b, c, d, ab, bc, cd\}$. You can see that various coverings differ in the average segment length.

2.2. Problem formulation

Given a segment inventory we would like to measure its quality, i. e. how comprehensible and/or natural speech we can expect if we use it for speech synthesis. We have already argued that the quality of the speech depends on the average length of the segment — roughly said, longer segments will produce better (more natural, more comprehensible) speech.

Given an unknown text[1] and a segment inventory, there may be more than one way how to cover the given text by the segments from the inventory. Table 1 presents an example. These covers can differ in the total number of segments needed and thus in the average segment length (ASL). The problem is formally expressed as follows:

Let C be an ordered collection of phonemes

$$C = c_1, c_2, \dots, c_n, \; c_i \in V$$

and S a segment inventory. Find a set of indices

$$I^* = \{i_1, i_2, \dots, i_{k^*}\},$$
$$\text{where } 1 = i_1 \leq i_2 \leq \dots \leq i_{k^*} = n$$

[1] For the sake of simplicity we will use the term text instead of more accurate term “phonetically transcribed text” in the rest of the paper.

such that

$$\forall j = 1, 2, \ldots, k^* - 1 : \; c_{i_j} c_{i_j+1} \ldots c_{i_{j+1}-1} \in S$$

and $\frac{n}{k^*}$ is maximized. We assume that at least one possible cover exists.

2.3. Algorithm

Denote K the length of the longest segment from the inventory, i.e. $K = max_{i=\{1,2,\ldots,m\}} l(s_i)$. In each step of the algorithm we will have a list of at most K competing solutions. The algorithm then proceeds as follows:

1. Initialize the list of candidade solutions by the following way. For each $j = 1, \ldots, K$ take the sequence $c_1 \ldots c_j$ and determine whether it forms a segment in the inventory S. In positive case add the solution $\{1, j\}$ to the list of candidade solutions.

2. Take the “shortest” solution from the list, i.e. the solution whose maximal index is minimal over all solutions in the list. Let us denote this solution I and let i_k be the largest index in this solution.

3. For each $j = 1, \ldots, K$ take the sequence of symbols $c_{i_k+1} \ldots c_{i_k+j}$ and determine whether it forms a segment in the inventory S. In positive case form a new solution $\hat{I} = I \cup \{i_k + j\}$. Obviously, the index $i_k + j = i_{\hat{k}}$ is the largest index in the solution $\hat{I}$.

4. Check whether there is another solution with the largest index equal to $i_{\hat{k}}$. If not, add the solution $\hat{I}$ into the list of candidade solutions. If yes, compare sizes of these two solutions and keep only the smaller one in the list of candidade solutions.

5. Remove I from the list of solutions.

6. Repeat steps 2–5 until the end of the text is reached. For each candidade solution I calculate the average segment length as the ratio $\frac{n}{|I|}$. I^* is the solution with the highest ASL.

The whole procedure is demonstrated in Figure 1.

2.3.1. *Proof of the correctness of the algorithm*

We first prove that the algorithm finds the optimal solution. The key step of the algorithm is the comparison of solutions in step 4. Assume that $I = \{i_1, i_2, \ldots, i_k\}$ and $I^{'} = \{i^{'}_1, i^{'}_2, \ldots, i^{'}_k\}$ are two candidade solutions such that $i_k = i^{'}_k$ and $|I| \le |I^{'}|$. Denote $I^*_{i_k}$ the solution of the minimum text cover problem for $c_{i_k+1} c_{i_k+2} \ldots c_n$. For the solution of the original problem then holds:

$$|I \cup I^*_{i_k}| \le |I^{'} \cup I^*_{i_k}| \Rightarrow \frac{n}{|I \cup I^*_{i_k}|} \ge \frac{n}{|I^{'} \cup I^*_{i_k}|}$$

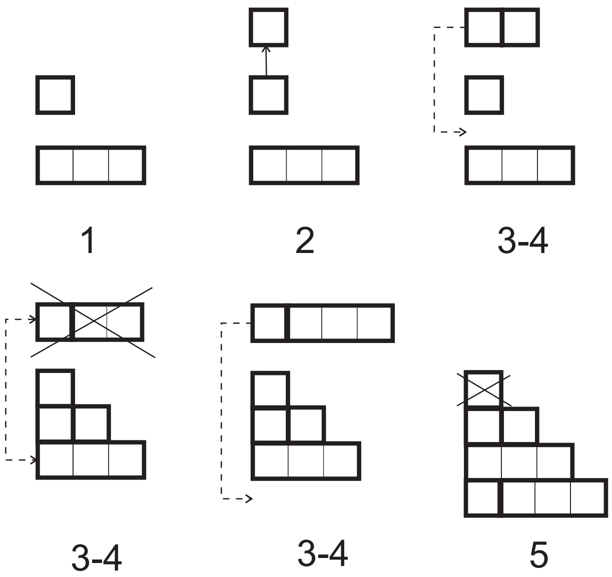

Figure 1: Searching for the minimal text cover ($K = 3$)

Thus, $I^{'}$ cannot be a part of the optimal solution unless $|I| = |I^{'}|$.

Now, let us pay attention to the time complexity. The first step of the algorithm generates at most K solutions whose largest indices are in the range $[1, K]$. Assume now that we have a list of at most K solutions whose largest indices are in the range $[i_k, i_k + K - 1]$. Take the "shortest" solution from the list as described in the step 3 of the algorithm. By adding a new segment to this solution we can generate the solutions whose largest indices will be in the range $[i_k + 1, i_k + K]$. The solutions remaining in the list have their largest indices in the range $[i_k + 1, i_k + K - 1]$. As we keep only one solution for each largest index, after applying steps 3 and 4 we will have a list of at most $K + 1$ solutions whose largest indices are in the range $[i_k, i_k + K]$. After applying step 5 we will reduce it to the list of at most K solutions with their largest indices in the range $[i_k + 1, i_k + K]$. Therefore, the time complexity of the algorithm is $O(n * K)$.

2.3.2. *Alternative formulation*

In the first formulation of the problem we assumed that the corpus is a (long) collection of phonetic symbols. An alternative is to consider the corpus to be a set of longer units (words, prosodic phrases). The problem can be then expressed as follows:

Let C be a set of units

$$C = \{u_1, u_2, \ldots, u_n\}, \; u_i = c_1 c_2 \ldots c_{k_i}, \; c_j \in V$$

and S a segment inventory. Find the minimum text cover for each of the corpus units. We again assume that at least one possible cover exists for each unit.

2.4. Experiments

We will now look at results computed on the test data. The data has been taken from the novel “Krakatit” written by Czech writer Karel Čapek. The text has been automatically phonetically transcribed using the procedure used in Czech speech synthesizer Demosthenes. The phonetic transcription module includes also some text normalization like rewriting letters, abbreviations and foreign words ([4]). After the phonetic transcription, all characters except symbols representing phonemes and space have been removed from the corpus. Thus, the phonetic alphabet V consisted of 35 phonemes (10 vowels and 25 consonants) used in Demosthenes phonetic transcription module and 1 extra symbol for space. The total size of the test data after applying all operations was 433,314 phonetic symbols.

We have generated 1,000 random inventories of sizes 20, 50, 100, 200 and 500. Their average ASLs found by the OTC algorithm are shown in Table 2. Column 1 shows results for inventories containing only segments consisting of 1 or 2 phonemes, column 2 shows the results for inventories containing segments consisting of 1, 2 or 3 phonemes and column 3 shows the results for inventories containing segments consisting of 1, 2, 3 or 4 phonemes. It follows from the table that adding longer segments to segment inventories is in general not effective.

Number of segments	ASLrand 1,2	ASLrand 1,2,3	ASLrand 1,2,3,4
20	1.175	1.176	1.061
50	1.3	1.303	1.142
100	1.449	1.448	1.254
200	1.645	1.644	1.425
500	1.946	1.869	1.743

Table 2: ASL of random segment inventories and of the most probable bigrams

Clearly, the ASL of the particular segment inventory depends on probabilities of segments it contains. Inventories consisting of highly likely segments should have also higher ASL. We tested this hypothesis on the set of artificially designed inventories. We determined relative frequencies of 2-,3- and 4-grams over the Czech corpus ESO ([5]). We constructed segment inventories in the same way as in the previous example, but consisting only the most probable n-grams. Results are referred in Table 3.

It is obvious that the inventories constructed from the most probable n-grams have higher ASL. Again, segment inventories containing longer segments are not significantly better.

We have also computed ASL of one segment set used for speech synthesis of the Czech language in the recent past. L-syllables are described in [6] and [7]. L-syllable is basically defined as a pair CV (we will not take all modifications into account here). Hence, there are 250 such segments. ASL of such an inventory is 1.451. It seems that the quality of L-syllable based synthesis can be increased by adding some segments increasing ASL to the segment set.

Number of segments	ASLngram 1,2	ASLngram 1,2,3	ASLngram 1,2,3,4
20	1.223	1.223	1.223
50	1.387	1.379	1.379
100	1.564	1.569	1.569
200	1.756	1.785	1.786
500	1.966	2.089	2.083

Table 3: ASL of the most probable n-grams

3. Estimation based on a language model

When designing a speech synthesis system we need to find a segment inventory achieving minimal cover of *any* text. One way to do it is to collect a large corpus of language data and to choose an inventory with the highest ASL achieved over this corpus. Until now, we don’t know any deterministic algorithm that is able to choose the optimal segment inventory. Of course, we are still able to use some non-deterministic techniques, e.g. genetic algorithms. These techniques, however, typically claim evaluation of hundreds or even thousands of candidade solutions during computation. Computing ASL over a large corpus would be therefore computationally infeasible.

Out solution is based on building a language model of the corpus and generating short sample text using this model. If the sample text is long enough and if it is generated by an appropriate language model, it will model statistical properties of the corpus with satisfactory precision. In this case we can use the sample text as a data for evaluating segment inventories significantly speeding up the computation without loss of precision.

3.1. Estimation accuracy

To test the accuracy of the procedure based on the language modeling we have made some experiments. The accuracy has been tested on the same data as in the section 2.4.

The first performed test was the dependence of the evaluation accuracy on the quality of the language model. Several n-gram language models have been built based on the statistics derived from the Czech corpus ESO. The corpus has been processed in the same way as the test data. After all operations it consisted of 329,210,001 phonetic symbols. Perplexities of language models (per phoneme) are shown in Table 4, row 1.

Accuracy of the evaluation procedure has been tested on the text consisting of 10,000 phonemes generated by the particular language model. The procedure has been tested over 1,000 randomly chosen inventories of various sizes and consisting of segments of various lengths. The averages of differences between optimal and estimated ASLs are summarized in row 2. The average difference decreases from 0.46 for the unigram model to 0.01 for the trigram model, then remains stable. Even more interesting are correlation coefficients (row 3). The correlation for the simplest model is 0.86, for the others it is almost 1. This result tells us that even if we cannot rely on the absolute value computed we can use it for comparison of the inventories.

Order of the model	1	2	3	4	5
Perplexity	24.1	13.1	8.8	5.7	4.2
Accuracy	0.46	0.04	0.011	0.012	0.011
Correlation	0.86	0.992	0.998	0.998	0.998

Table 4: Average differences and correlation coefficients of n-gram language models.

Another factor influencing evaluation accuracy is the length of the generated text. This influence has been again tested across 1,000 randomly chosen inventories and with several language models. Table 5 shows the average differences and correlation coefficients. It follows from the table that text with length about 1,000 is sufficient to reliably estimate average segment length of the segment inventory.

Length	10	20	50	100	200
Accuracy	0.43	0.15	0.07	0.05	0.03
Correlation	0.47	0.78	0.94	0.97	0.99
Length	500	1,000	2,000	5,000	10,000
Accuracy	0.025	0.016	0.015	0.012	0.013
Correlation	0.992	0.996	0.997	0.998	0.998

Table 5: Dependence of the estimation accuracy on the length of the generated text.

3.2. Advantages

One of the main advantages of the ASL estimation based on language modeling has been already mentioned — it is speeding up the evaluation with only a slight loss of accuracy. But, it is not the only advantage.

Obviously, the optimal segment inventory of the limited domain speech synthesizer will differ from the inventory of the general text-to-speech system. It is sometimes the case that the TTS system designer is not able to collect enough data to built a reliable language model of the particular domain. When this situation occurs, it can be solved by building a general language model from a large corpus and to interpolate it with the domain dependent model. This merged model can be used to generate sample text for the estimation procedure described above.

4. Conclusions and future work

A method of measuring the quality of the segment inventory of a concatenative TTS system has been presented in the paper. We believe that the criterion we call average segment length is reasonably defined and can serve as a rough measure of quality of the segment inventories consisting of variable length segments.

We are aware of the fact that such a criterion needs to be more precise. In our future work we would like to incorporate some other features to it, especially some coarticulation and prosodic properties of segments. Our assumptions has to be proven in tests with human subjects as well.

The approach described in this paper will be used to search for the optimal segment inventory of the Czech test-to-speech synthesizer Demosthenes.

5. Acknowledgements

The author is grateful to Ivan Kopeček and Pavel Gaura for their valuable comments. This research has been supported by the Grant Agency of the Czech Republic under the grant GA 201/99/1248.

6. References

[1] Nick Campbell and Alan W. Black, “Prosody and the selection of source units for concatenative synthesis,”, in *Progress in Speech Synthesis*, chapter 22, pp. 279–292. Springer, Berlin, Germany, 1996.

[2] Roger Guaus i Térmens and Ignasi Iriondo Sanz, “Diphone-based unit selection for catalan TTS synthesis,” in *Proceedings of TSD 2000*, Brno, Czech Republic, 2000, Springer.

[3] Jon R. W. Yi and James R. Glass, “Natural-sounding speech synthesis using variable-length units,” in *The 5th International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[4] Robert Batůšek and Jan Dvořák, “Text preprocessing for Czech speech synthesis,” in *Proceedings of TSD’99*, Pilsen, Czech Republic, Sept. 1999.

[5] Karel Pala and Pavel Rychlý, “Mutual information in Czech corpus ESO,” in *Proceedings of TSD’98*, Brno, Czech Republic, Sept. 1998.

[6] Robert Batůšek, “Statistics of the syllable segments for speech synthesis of the Czech language,” in *Proceedings of TSD’98*, Brno, Czech Republic, Sept. 1998.

[7] Ivan Kopeček, “Speech synthesis based on the composed syllable segments,” in *Proceedings of TSD’98*, Brno, Czech Republic, 1998.