Minimum Text Corpus Selection for Limited Domain Speech Synthesis*

Markéta Jůzová and Daniel Tihelka

University of West Bohemia, Univerzitní 8, Plzeň, Czech Republic
juzova@kky.zcu.cz, dtihelka@kky.zcu.cz
http://www.kky.zcu.cz

Abstract. This paper concerns limited domain TTS system based on the concatenative method, and presents an algorithm capable to extract the minimal domain-oriented text corpus from the real data of the given domain, while still reaching the maximum coverage of the domain. The proposed approach ensures that the least amount of texts are extracted, containing the most common phrases and (possibly) all the words from the domain. At the same time, it ensures that appropriate phrase overlapping is kept, allowing to find smooth concatenation in the overlapped regions to reach high quality synthesized speech. In addition, several recommendations allowing a speaker to record the corpus more fluently and comfortably are presented and discussed. The corpus building is tested and evaluated on several domains differing in size and nature, and the authors present the results of the algorithm and demonstrate the advantages of using the domain oriented corpus for speech synthesis.

Keywords: limited domain speech synthesis, concatenative speech synthesis, text corpus, speech units, text chunks, unit concatenation

1 Introduction

The limited domain speech synthesis (LDS) may at first seem to be a trivial task, when compared to the general-purpose text-to-speech (TTS) synthesis system. However, to reach a high-level of naturalness within a LDS system is not as simple as it looks like, mainly due to the fact that any non-natural artefact occurrence following longer natural-sounding speech is perceived very negatively [1]. Therefore, we must be very careful in the process of limited-domain (LD) speech corpus preparation, since a concatenation algorithm does not have to have many speech unit candidates available to ensure smooth segments concatenation, as it is the case of "classic" general-purpose unit selection TTS system [2].

More specifically, when speech corpus for a general TTS system[1] is being prepared, we focus on short speech units (e.g. diphones) to make the speech corpus rich in, trying

* This work was supported by the European Regional Development Fund (ERDF), project "New Technologies for Information Society" (NTIS), European Centre of Excellence, CZ.1.05/1.1.00/02.0090, the Technology Agency of the Czech Republic, project No. TA01030476 and SGS-2013-032.

[1] We will limit ourself entirely to a system embedding unit selection method, since it firstly still provides more natural output than the HMM-based synthesis [3], and secondly its comparison with the LD synthesis operation is much more straightforward.

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 398–407, 2014.

to select such a text material (from various domains) which contains all possible units (regarding their phonetic and prosodic context) with the number of occurrences as large as possible or manageable [4,5]. Only in such a way we maximize the chance that an appropriate sequence of (short) units will be selected and concatenated (at least unless units' prosodic synonymy and homonymy is taken into an account [6], which is out of focus of the present paper), to realize speech without audible unnatural artefacts.

The nature of limited domain, however, attracts with the advantage of the possibility to have much smaller, and thus much cheaper speech corpus, while reaching higher level of speech naturalness than with a general TTS system [7], especially if the required domain is away from the domains of texts used in the TTS system. Naturally, it means that much longer units, like words or whole phrases, must be used in this case. On the other hand, the design of the corpus must be carried out much more carefully, since there is no "units redundancy" which we can rely on in the case of general TTS. Inappropriate corpus design can easily lead either to significantly larger corpus than would be necessary, or to the need to concatenate recorded phrases in inappropriate places, e.g. in pauses or phrase breaks. It is the score of the present paper to show how to design a minimum text corpus suitable for a limited domain speech synthesis, while preserving sufficient units redundancy for the smooth concatenation of (longer) units the system is built on.

2 Limited Domain Text Representation

There are many forms of LD text representation, depending on the size of the domain and on the variability of texts within it. The most trivial example is a strictly-limited domain characterized by several (tens, hundreds) fixed sentence structures with variable items (let us call them *slots*), and sets of words to be placed in. The example can be an automatic system informing about departures and arrivals of trains, the fragment of which is shown in the following snippet:

Jak vám mohu pomoci?	*How can I help you?*
V kolik hodin chcete jet ze stanice	*What time do you want to go from station*
Rokycany do stanice Strakonice?	*Rokycany to station Strakonice?*
V kolik hodin chcete jet ze stanice Chrást u Plzně do stanice Domažlice?	*What time do you want to go from station Chrást u Plzně to station Domažlice?*
Cesta trvá 3 hodiny.	*The journey takes 3 hours.*

where boxed items are variable parts in otherwise fixed text frames.

However, in general, the domain, although still limited somehow, is wider with texts showing a larger degree of variability. The example of such domain can be weather forecast, transcripts of real ATC communications (*Air Traffic Control*, provides information and advisory services to planes using a defined phraseology), or ATIS (*Airport Terminal Information System*, broadcasting informations to planes before landing), as the last two are real examples of domains we currently build the LDS system for. Few representative examples of the latest are as follows:

WIND [CALM].
[CROSS] WIND.
[GUSTING] [CROSS] WIND.
WIND [DATA NOT AVAILABLE].
WIND [120] DEGREES [3] KNOTS BETWEEN [060] AND [150] DEGREES.
WIND [230] DEGREES [9] KNOTS.
WIND [240] DEGREES [13] GUSTING [23] KNOTS.
EXPECT [TAIL] WIND.
EXPECT WIND [VARIABLE].
WIND [VARIABLE].
WIND [VARIABLE] [1] KNOT.
WIND SPEED [180] DEGREES [4] [METRES PER SECOND].
WIND SPEED [25] [KILOMETRES PER HOUR].
WIND SPEED [10] [METRES PER SECOND] [GUSTING].
WIND SPEED [5] [METRES PER SECOND] [MAXIMUM].
WIND SPEED [12] [METRES PER SECOND] [MINIMUM].
WIND SPEED [6] [METRES PER SECOND].
WIND SPEED [VARIABLE] [8] [METRES PER SECOND].

where it can be seen that the structure of phrases is much more variable with significantly shorter fixed text frames, if there are even any such at all.

Thus, in the real situations, we (or the LDS system creators in general) are facing the task of representative phrases selection, given a (large) set of "raw" texts covering the given domain. In other words, having a set of texts which may appear in the domain, the task is to select the set of segments (chunks) to be recorded which is both the smallest in size and rich in variability (i.e. covers the domain as the whole or allows a high-quality generation of the missing pieces). In principle, the following are rough ways of how texts to be recorded can be extracted from the given texts:

1. record all available sentences
   pros: no concatenation is necessary, because all possible sentences are in the corpus (theoretically), the recorded sentences are only replayed back with the highest level of naturalness
   cons: huge (= expensive) speech corpus and storage resources consumption
2. split the given text to a disjunctive set of text chunks and to record each chunk individually
   pros: the smallest possible speech corpus
   cons: chaining in the pauses between words – result will not sound fluently, speech artefacts may appear at the concatenation points
3. split the given text to a disjunctive set of text chunks, but enhance each chunk with its left and right context of the appropriate length, and to record each extended chunk individually (discussed in Section 2.1)
   pros: small speech corpus, the possibility to find the optimal concatenation point in the contexts which naturally overlap
   cons: not the smallest possible speech corpus

Here, the third way seems to be a suitable compromise, and that is the approach we adopted. The question now is how to extract the chunks, so further follows the algorithm we have developed for this task.

2.1 The Role of the Context

Let us suppose now, that we want the corpus of the minimum size. To ensure this, there must be minimum duplicity in the recorded texts. Thus, having for example the following chunks in the corpus:

WIND SPEED ...
... 4 ..., ... 5 ..., ... 6 ...
... METRES PER SECOND ...
... MINIMUM.

we want to synthesized the sentence *WIND SPEED 6 METRES PER SECOND MINIMUM*. Due to no possible overlap, the output sentence would have to be composed from the chunks:

WIND SPEED	*6*	*METRES PER SECOND*	*MINIMUM.*

Since the individual chunks start and end with pause, they have to be recorded in this way, short pause will naturally appear at each concatenation of two chunks. Even if the pause is minimized, there is still a possibility of speech style twist, since people have a natural tendency to pronounce isolated words in a different way than they would be pronounced within a phrase [8].

The presence of context word (or any arbitrary, yet natural text chunk) helps avoiding this, of course at the cost of a bit larger speech corpus. The situation is illustrated on Figure 1 and in details described in [9].

Fig. 1. The concatenation of text chunks in their contexts. The arrow mark all possible concatenation point from which the best (highlighted) can be dynamically selected during synthesis.

3 Text Selection Algorithm

To formalize further description, let $S = \{S_1, S_2, \dots\}$ be a set of all sentences from the domain we have in disposal, each consisting of a sequence of words $S_s = [w_1, w_2, \dots]$. Let us define $W = \{w_1, w_2, \dots\}$ to be the set of all unique words in the whole set S, and $W(s)$ to be the set of all unique words in a s-th sentence S_s. Similarly, let W^{II} and $W^{II}(s)$ be defined for all unique bigrams in S and S_s respectively.

Now, for each word we can simply count the number of its occurrences $C(w)$, $w \in W$, as well as the number of occurrences for each bigram $C^{II}(w)$, $w \in W^{II}$.

Let further

$$C = \sum_{w \in W} C(w) \tag{1}$$

$$C^{II} = \sum_{w \in W^{II}} C^{II}(w) \tag{2}$$

denote the total number of unigrams (i.e. words) and bigrams in S, and

$$C(s) = \sum_{w \in W(s)} C(w) \tag{3}$$

$$C^{II}(s) = \sum_{w \in W^{II}(s)} C^{II}(w) \tag{4}$$

denote the total number of unigrams and bigrams in a particular sentence S_s.[2]

The task of the selection algorithm described further is to select new set of sentences S^* containing (all) text chunks which appeared in the original set S, but minimum in size and preserving a context through which individual corresponding chunks can be concatenated together, as discussed in Section 2.1.

For the algorithm we also define a constant $R \in (0, 1)$, managing the ratio between unigrams (value closer to 0) and bigrams (value closer to 1), which affects the selection algorithm as described in Section 3.1. It is also possible to define the maximum number of text chunks M which will be selected by the algorithm and recorded later on. Note however, that too small value of M will lead to S^* where $W^* \subset W$ (i.e. some words of the domain will not exist in the selected chunks), but the choice of the suitable value of M depends on many factors like the size and character of the domain, budget limitations, speaker availability, and so on.

The proposed chunks selection algorithm for the speech corpus building is a simple loop in the interval $l = 0, 1, \ldots, M$, in which the following sequence of operations is carried out:

Sentence evaluation – all sentences $s = 1, 2, \ldots$ are assigned with score computed as

$$\sigma_s = R\frac{C^{II}(s)}{C^{II}} + (1 - R)\frac{C(s)}{C} \,. \tag{5}$$

The reason of adding counts of different bigrams and unigrams is to cover more data with the output text corpus, the example is illustrated in Section 3.1.

Choice of the best sentence – that sentence with the highest σ is selected:

$$s^* = \arg\max_s \sigma(s) \tag{6}$$

The sentence S_{s^*} will contribute the most to the coverage of the limited domain, so it is added to the set of text chunks S^* to be recorded.

[2] Note that although $C(w)$ and $C(s)$ can be interchanged, we will strictly use indexing s for sentences and w for words or bigrams in case of $C^{II}(w)$.

Resetting of counters – we does not have to consider the words from the selected sentence (and thus the bigrams as well) any more, since they are from now included in the LD text corpus and their repetition will only increase the size of the corpus without adding any real benefits (plus it violates our requirement for disjunctive set S^*). Therefore, we can set the $C(w) = 0, \forall w \in W(s^*)$ and $C^{II}(w) = 0, \forall w \in W^{II}(s^*)$.

Cutting of phrase to chunks – to prevent the consideration of (now) needless bigrams in the selection procedure, we split the sentences in S containing bigrams in $W^{II}(s^*)$ according to the following scheme:

1. examine all $s = 1, 2, \ldots, s \neq s^*$
2. for the given s examine all bigrams from $w = W^{II}(s)$ in the order of their occurrence in S_s
3. if $w \in W^{II}(s^*)$,
   - split the sentence into two in such a way that the left part S_s^l will end with the first word of the bigram and the right part S_s^r will start with the second bigram's word
   - clear S_s^l or S_s^r if it contains one phone only (that from w)

   otherwise keep S_s unchanged
4. remove the original S_s and add the splits into the set (if they are not empty) $S = (S \setminus \{S_s) \cup \{S_1^l, S_2^r\}$ and start from 1 with the updated set

Each sentence can, of course, be split to more than two chunks, depending on the bigrams in the selected sentence S_{s^*}. In the following example, S_{s^*} contains text chunks *WIND SPEED* and *METRES PER SECOND* which are also contained in another sentence in the set:

S_{s^*}:	*WIND SPEED 5 METRES PER SECOND GUSTING.*
S_s, $s \neq s^*$:	*WIND SPEED 12 METRES PER SECOND MINIMUM.*
S^l from 4th step:	*... SPEED 12 METRES ...*
S^r from 6th step:	*... SECOND MINIMUM.*

Naturally, the better the domain is described (i.e. more texts from the domain we have in disposal), the better the algorithm will work, and of course the lower is the chance that some of the words or texts chunks are missing in the speech corpus. Sometimes, however, it is not simply possible, for example where names (both first and especially surnames) appear in the domain. Before the algorithm start, nevertheless, we can extend the set S with "cloned" sentences, where additional names are added. Similarly, we can extend the set with numbers, hours and any other slot types for which we need to ensure full (or higher) coverage.

3.1 The Impact of *R* value

The choice of value R has a decisive influence on the sentence selection. To illustrate it, we created an artificial set of sentences $S = \{S_1, S_2, S_3, S_4\}$

S_1: A B C D E
S_1: A A A D
S_1: A A A D
S_1: A B A A E D

with the values of unigram/bigram counts shown in Table 1, and with the values of per-sentence unigram, bigram and $\sigma(s)$ (see Equation (5)) shown in Table 2.

Table 1. The counts of unigram and bigram occurrences per token $C(w)$ and $C^{II}(w)$, and their sum as defined by Equations (1) and (2).

w	A	B	C	D	E
$C(w)$	10	4	3	3	2
C	22				

w	AA	AB	AD	AE	BA	BC	CA	CD	DE	ED
$C^{II}(w)$	4	4	1	1	1	3	1	1	1	1
C^{II}	18									

Table 2. The illustration of the impact of R-value on the sentence choice. Bold values are the highest values of Equation (5) denoting the selected sentence s for the given R.

			R										
s	$C(s)$	$C^{II}(s)$	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
1	22	9	0.50	0.55	0.60	0.65	0.70	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**	**1.00**
2	13	5	0.28	0.31	0.34	0.37	0.40	0.43	0.47	0.50	0.53	0.56	0.59
3	19	11	0.61	0.64	0.66	0.69	**0.71**	0.74	0.76	0.79	0.81	0.84	0.86
4	17	12	**0.67**	**0.68**	**0.69**	**0.7**	0.71	0.72	0.73	0.74	0.75	0.76	0.77

When the value of R is shifted towards bigrams ($\rightarrow 1$), the algorithm prefers sentences with the higher number of higher frequent bigrams (AA, AB), while it prefers unigrams (words) when the value ($\rightarrow 0$). The value near 0.5 leads to the balance of choice between both the most common unigrams and bigrams.

If the value M is defined as $infinity$, meaning that we want to generate all chunks from the set S (i.e. 100% coverage of the domain), any value of R has no influence on the resulting S^* set, only the order of chunks selection is different.

4 Helping Speaker in the Recording

For the recording of the corpus, we have used our own tool described in [4]. Contrary to the recording for general-purpose TTS, where whole phrases and/or sentences are presented to the speaker, the text from the domains were smashed to chunks which can be difficult for a speaker to record them in the required prosody style. The fact is that a chunk is required to be pronounced in such a way to fit seamlessly into the phrase (or phrases) where it belongs to.

To help the speaker keep the style while recording, we decided to present the chunks in relation with the original phrase:

```
WIND SPEED 2 KILOMETRES PER HOUR.
```

```
... 3 KILOMETRES ...
... 4 KILOMETRES ...
... 9 KILOMETRES ...
```

In the recording tool, the whole text is recorded into a single prompt (which is cut later). The speaker was instructed to read the first line as a whole, and to read the following chunks in the same style as the chunk within the first line, with short pause between chunks. Since the human memory is capable to remember recent events better than further, and since the speaker is usually professional or semi–professional, there is good chance that the chunks will be recorded in very similar (or even undistinguishable) style to keep prosody and co-articulation. In the case that there are more chunks, new recording prompts are generated, like:

```
# WIND SPEED 2 KILOMETRES PER HOUR.
... 2 KILOMETRES ...
... 7 KILOMETRES ...
... 5 KILOMETRES ...
... SPEED 1 KILOMETRE PER HOUR.
```

The speaker was instructed not to read the '`#`'–starting line (although he/she could), it is intended to make the speaker recall the style in which the preceding prompt was read (naturally, chunks from the given phrase follow during the recording). And still, if the chunks' style diverges slightly (which cannot be entirely avoided), there is the context, in which the optimal concatenation point is searched for to ensure as smooth transition as possible.

5 Results and Evaluation

We tested the algorithm on several domains. The results of using this approach are described in the following subsections, for all experiments we set $R = 0.5$. To quantify the behaviour of the algorithm somehow, we compare the size of the original set S and the generated set S^* from the viewpoint of:

- the number of items in the sets, where the items are sentences in S and chunks in S^*,
- the average number of words per set item (average sentence/chunk length),
- the total number of words in the sets (i.e. C from Equation (1)),
- the number of different words in the sets (i.e. cardinality $\|W\|$),

while the set S can be considered either in *full* form (with "classic" texts) or in *reduced* form where particular slot values can be replaced by appropriate place–holders, e.g. numbers by `$NUM` and so on. The size of S^* is always presented in its "full" form.

5.1 ATIS Limited Domain

For this domain, we had thousands of "real" sentences, automatically downloaded every 2 hours from the public web page where actual ATIS informations are published. The downloading started at autumn 2013.

Looking at the corpus sizes in the Table 3, it can bee seen that S^* is much lower than full S. The selected chunks occupy only 0.15% of all the possible sentences, while covering 100% of the full set.

Table 3. The comparison of original sentences set size to the size of selected chunks.

	S, reduced	S, full	S^*
items in sets	1,177	510,311	3,368
number of words C	7,408	8,952,311	13,507
number of different words	728	737	737
average sentence length	6.3	17	4.0

5.2 ATC Communication Messages

The domain of *Air Traffic Control* has a rather strictly defined phraseology, nevertheless it is both quite large and not strictly abided by pilots and controllers. In the input, moreover, we have transcripts of real communications instead of phrases defined by the phraseology, so there is much higher variability in the form of a many particular phrases (inserted or missing "filler" words, change in the word order, etc.).

Due to the more vague nature of the domain, we, therefore, did not required full coverage in the selection, but we have choose $M = 1,000, 2,000, \ldots, 5,000$ of selected chunks. Therefore, in Table 4 we further present the perceptual coverage of words (comparing W items from full S and S^*) showing how big the intersection of the two sets is, and the number of words $\notin S^*$ as well as the highest $C(w)$ for $w \notin S^*$.

Table 4. The comparison of original sentences set size to the size of selected chunks.

	S, full	S^*				
number of sentences	45,551	1,000	2,000	3,000	4,000	5,000
number of words C	391,429	6,919	12,276	15,455	19,567	23,431
number of different words $\|\|W\|\|$	2,012	938	1197	1347	1513	1617
coverage of $\|\|W\|\|$	100%	46.6%	59.5%	66.9%	75.1%	80.4%
number of missing words	–	1,074	815	665	499	395
$\max(C(w))$ for missing words	–	18	8	4	3	2

It can be seen from Table 4, that the coverage reached is rather high and all the most frequent words are included in S^*. Although there is still a significant number of word and bigram tokens missing in the selected text, and thus they will have to be generated by a general TTS, their frequency is very low.

6 Conclusion

The paper presents the algorithm for building a limited–domain text corpus, which is as small as possible possible, while still allowing natural and smooth transition between the recorded chunks by keeping sufficient context in which an optimal (not being a

pause) concatenation point can be found. It is demonstrated, that even with the context, the number of prompts to be recorded is only a fragment of the original domain. What remains to be said is, that all the presented corpora have already been recorded, tested and evaluated, and the dominance of the LD synthesis using them has been proved, as discussed in [9].

Special thanks are due to the National Grid Infrastructure MetaCentrum providing access to computing and storage facilities under the program LM2010005 “Projects of Large Infrastructure for Research, Development, and Innovations”.

References

1. Brenton, H., Gillies, M., Ballin, D., Chatting, D.: The uncanny valley: does it exist. In: 19th British HCI Group Annual Conference: workshop on human-animated character interaction. (2005)
2. Tihelka, D., Kala, J., Matoušek, J.: Enhancements of Viterbi search for fast unit selection synthesis. In: INTERSPEECH 2010, proceedings of 11th Annual Conference of the International Speech Communication Association. (2010) 174–177
3. Grůber, M., Hanzlíček, Z.: Czech expressive speech synthesis in limited domain: Comparison of unit selection and HMM-based approaches. In Sojka, P., Horák, A., Kopeček, I., Pala, K., eds.: Text, Speech and Dialogue. Volume 7499 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 656–664
4. Matoušek, J., Tihelka, D., Romportl, J.: Building of a speech corpus optimised for unit selection tts synthesis. In: LREC 2008, proceedings of 6th International Conference on Language Resources and Evaluation, ELRA (2008)
5. Matoušek, J., Romportl, J.: On building phonetically and prosodically rich speech corpus for text-to-speech synthesis. In: Proc. of the second IASTED Int. Conf. on Computational intelligence, San Francisco, ACTA Press (2006) 442–447
6. Tihelka, D.: Towards automatic measure of similarity for use in unit selection. In: Signal Processing, 2008. ICSP 2008. 9th Int. Conf. on, Beijing, China (2008) 637–642
7. Black, A.W., Zen, H., Tokuda, K.: Statistical parametric speech synthesis. In: Proc. ICASSP, 2007. (2007) 1229–1232
8. Labov, W.: The Social Stratification of English in New York City. Center for Applied Linguistics, Washington, D.C. (1966)
9. Jůzová, M., Tihelka, D.: Tuning limited domain speech synthesis using general tts system. In: Accepted at Text, Speech and Dialogue 2014. (2014)