

Using Same-Language Machine Translation to Create Alternative Target Sequences for Text-To-Speech Synthesis

Peter Cahill¹, Jinhua Du², Andy Way², Julie Carson-Berndsen¹

¹School of Computer Science and Informatics, University College Dublin, Dublin, Ireland.

²School of Computing, Dublin City University, Dublin, Ireland.

{peter.cahill|julie.berndsen}@ucd.ie, {jdu|away}@computing.dcu.ie

Abstract

Modern speech synthesis systems attempt to produce speech utterances from an open domain of words. In some situations, the synthesiser will not have the appropriate units to pronounce some words or phrases accurately but it still must attempt to pronounce them. This paper presents a hybrid machine translation and unit selection speech synthesis system. The machine translation system was trained with English as the source and target language. Rather than the synthesiser only saying the input text as would happen in conventional synthesis systems, the synthesiser may say an alternative utterance with the same meaning. This method allows the synthesiser to overcome the problem of insufficient units in runtime.

Index Terms: speech synthesis, machine translation

1. Introduction

Speech synthesis is the term used to describe the creation of speech from any source other than a human vocal tract. Modern speech synthesis systems are typically software-based, where data is input in some form to control the speech created by a speech synthesiser. Text-to-speech (TTS) synthesisers are speech synthesisers that contain some extra components in order to be able to process text to create a speech representation before the actual synthesis occurs. Examples of modern TTS systems include [1, 2, 3].

Modern speech synthesis systems can be categorised into two groups: concatenative synthesisers and parametric synthesisers. Concatenative synthesisers create speech by using a database of pre-recorded speech segments to create new words and utterances. Parametric synthesisers train models from a database of pre-recorded speech, and at run time the speech is synthesised from parameters, without using a speech database. As concatenative speech synthesisers depend on the pre-recorded speech database for all speech sounds that they can create synthesis results can significantly deteriorate when appropriate segments are not in the speech database.

This paper presents a technique which uses machine translation methods to create multiple hypotheses of an input text, so that the synthesiser can select the text that it can synthesise best. Bulyko and Ostendorf [4] introduced the concept of using weighted finite state transducers to create multiple input sentences for a synthesiser. Pan and Weng [5] have a similar approach where they use realisation trees to create a form of word lattice for the synthesiser. The architectures presented in both [4] and [5] require that the synthesiser is modified so that it can input a word lattice rather than simple text. This approach results in the systems requirement for a specific synthesiser and cannot be trivially used with any text-to-speech

synthesiser. Nakatsu and White [6] employ a more generic approach, where the language generation component is not tightly coupled with the synthesiser so that any text-to-speech synthesiser may be used. The method presented by Nakatsu and White involves generating alternative text sequences from disjunctive logical forms by using the OpenCCG realiser [7]. The experiment described in [6] uses a set of 104 sentences of a similar structure. A language generation system that is trained from 104 sentences may be enough to improve synthesis performance if the target application has a restricted domain. This paper is concerned with improving the performance of general synthesis quality, where the synthesiser is not limited to a specific domain.

Rather than using finite state methodologies as a language generation component, this paper introduces the use of a machine translation (MT) system to generate the alternative sentences. MT systems can translate text from one language to another. Such systems are trained from suitable bilingual corpora, where the system will automatically identify the relevant patterns from text. Two modern machine translation technologies include example-based machine translation (EBMT) and statistical machine translation (SMT). EMBT systems use bilingual corpora at run time to translate by analogy. SMT systems train statistical translation models from bilingual corpora.

The remainder of this paper is structured as follows. Section 2 discusses the process of creating a corpus that is suitable for same-language machine translation. Section 3 introduces the machine translation system used in the experiment. Section 4 describes how the synthesiser processes the N -best list from the MT component. Section 5 discusses the insights gained from the experiment and Section 6 concludes.

2. Designing a Same-Language Machine Translation Corpus

To train an English to English machine translation system, a parallel English text corpus is required. The parallel text corpus contains one pair of sentences for each entry. Each pair of sentences consists of a source and a target sentence, both of which have the same meaning. As same-language machine translation corpora are not available, a small corpus, suitable for a proof-of-concept experiment was developed as part of the work presented in this paper.

The CMU ARCTIC [8] corpus was used as a starting point for the MT corpus. The CMU ARCTIC corpus seemed appropriate for this task for the following reasons: it is freely available, it does not contain any non-permissive distribution terms, it is commonly used in the speech synthesis domain and there are currently 7 different free speech databases available that use the CMU ARCTIC corpus. The machine translation system that

is trained from the ARCTIC-based corpus is suitable for use with at the 7 ARCTIC synthesis voices. Other speech synthesis corpora are often only used to record a single voice, and therefore would need a different parallel text to be created for each one. By including this corpus in the machine translation training corpus, the trained machine translation system is likely to use phrases from the CMU ARCTIC corpus in the translation results. As such phrases exist in the speech database of the synthesiser, it is likely to say them at a very high quality.

The training corpus contained 500 (i.e. 250 pairs of) sentences. The corpus was structured so that the CMU ARCTIC sentences were always the target element in each pair of sentences. The motivation for this structure was for the machine translation system to learn to translate from *general* English to *CMU ARCTIC* English. While both *general* English to *CMU ARCTIC* English are in fact English, this method will encourage the machine translation system to translate general English (i.e. user input) to the English phrases that occur in the CMU ARCTIC corpus.

3. MATREX: Statistical Machine Translation

The MATREX (Machine Translation using Examples) system [9] was used as the MT component in the experiment presented in this paper. It is a hybrid system which exploits EBMT and SMT techniques to build a combined translation model.

In this speech synthesis task, the source and the target side of the training data are the same - English. In order to generate the N -best list for each source sentence from the monolingual corpus, a synonymous parallel training corpus is constructed. Both source and target sides of the data set is chunked by using a marker-based chunker [10]. These chunks are then aligned using a dynamic programming, edit-distance-style algorithm and combine them with phrase-based SMT-style chunks into a single translation model.

The MATREX system is a modular hybrid data-driven MT system, built following established Design Patterns, which exploits aspects of both the EBMT and SMT paradigms. It consists of a number of extendible and re-implementable modules, the most significant of which are:

- *Word Alignment Module*: outputs a set of word alignments given a parallel corpus,
- *Phrase Extraction Module*: extract the basic phrase table according to the word alignment points and then calculate the probabilities,
- *Chunking Module*: outputs a set of chunks given an input corpus,
- *Chunk Alignment Module*: outputs aligned chunk pairs given source and target chunks extracted from comparable corpora,
- *Decoder*: returns optimal translation given a set of aligned sentence, chunk/phrase and word pairs.

In the MATREX system, these modules may comprise of wrappers around pre-existing software. For example, our system incorporates a wrapper around GIZA++ [11] for word alignment and a wrapper around Moses [12] for decoding. It should be noted, however, that the complete system is not limited to using only these specific module choices. The following subsections describe the modules which are unique to the MATREX system.

3.1. Marker-Based Chunking

The chunking module used for the shared task is based on the Marker Hypothesis, a psycholinguistic constraint which posits that all languages are marked for surface syntax by a specific closed set of lexemes or morphemes which signify context. Using a set of closed-class (or “marker”) words for a particular language, such as determiners, prepositions, conjunctions and pronouns, sentences are segmented into chunks. A chunk is created at each new occurrence of a marker word with the restriction that each chunk must contain at least one content (or non-marker) word.

3.2. Chunk Alignment

In order to align the chunks obtained by the chunking procedures described in Section 3.1, we make use of an “edit-distance-style” dynamic programming alignment algorithm.

In the following, a denotes an alignment between a target sequence e consisting of I chunks and a source sequence f consisting of J chunks. Given these sequences of chunks, we are looking for the most likely alignment \hat{a} :

$$\hat{a} = \underset{a}{\operatorname{argmax}} \mathbb{P}(a|e, f) = \underset{a}{\operatorname{argmax}} \mathbb{P}(a, e|f)$$

Alignments such as those obtained by an edit-distance algorithm are considered first, i.e.

$$a = (t_1, s_1)(t_2, s_2) \dots (t_n, s_n),$$

with $\forall k \in \llbracket 1, n \rrbracket$, $t_k \in \llbracket 0, I \rrbracket$ and $s_k \in \llbracket 0, J \rrbracket$, and $\forall k < k'$:

$$\begin{aligned} t_k &\leq t_{k'} \text{ or } t_k = 0, \\ s_k &\leq s_{k'} \text{ or } s_k = 0, \end{aligned}$$

where $t_k = 0$ (resp. $s_k = 0$) denotes a non-aligned target (resp. source) chunk.

The following model is assumed:

$$\mathbb{P}(a, e|f) = \prod_k \mathbb{P}(t_k, s_k, e|f) = \prod_k \mathbb{P}(e_{t_k}|f_{s_k}),$$

where $\mathbb{P}(e_0|f_j)$ (resp. $\mathbb{P}(e_i|f_0)$) denotes an “insertion” (resp. “deletion”) probability.

Assuming that the parameters $\mathbb{P}(e_{t_k}|f_{s_k})$ are known, the most likely alignment is computed by a simple dynamic-programming algorithm.¹

Instead of using an expectation-maximization algorithm to estimate these parameters, as commonly done when performing word alignment [13, 11], we directly compute these parameters by relying on the information contained within the chunks. The conditional probability $\mathbb{P}(e_{t_k}|f_{s_k})$ can be computed in several ways. Three main sources of knowledge were considered: (i) word-to-word translation probabilities, (ii) word-to-word cognates, and (iii) chunk labels. These sources of knowledge are combined in a log-linear framework. More details about the combination of knowledge sources can be found in [14].

Phrases of at most 7 words were extracted on each side. These phrases were then merged with the phrases extracted by the baseline system adding word alignment information, and this system was seeded with this additional information.

¹This algorithm is actually a classical edit-distance algorithm in which distances are replaced by opposite-log-conditional probabilities.

4. Incorporating MT N-best into Speech Synthesis

The concept presented in this paper is to use the N -best hypotheses from the machine translation system as input to the synthesiser. From the set of N -best hypotheses, the synthesiser itself can identify which one of the hypotheses it can say best, after looking at the content of each sentence and the units that are available in its speech database.



Figure 1: Overview of synthesis from the same-language machine translation system.

Figure 1 illustrates an overview of the synthesis process. The input text is passed to the same-language machine translation (SLMT) system. The output of the SLMT system is several hypotheses, which are all passed to the speech synthesiser. The speech synthesiser will then estimate how well it can synthesise each of the SLMT hypotheses. The text which the synthesiser estimates it can say best is the one that is chosen to be said in the final synthesis result.

For the experiments presented in this paper, unit selection diphone synthesiser was used. As a concatenative synthesiser, a unit selection synthesiser contains two metrics that can describe the quality of the utterance being synthesised: the total number of joins required and the total path cost.

4.1. Total Number of Joins Required

In all concatenative synthesisers, speech segments from a speech database are concatenated to create the target speech. It is common for concatenative synthesisers to encourage continuous segments from their database, so that if the target sequence were to contain N units, the synthesiser will need to concatenate fewer than N segments. It is reasonable to assume that the fewer the segments that need to be joined, the better the speech will be. For example, if the synthesiser can create the target utterance by using just 1 segment, then the speech would sound perfect as it would be a section of a spoken utterance.

When the synthesiser is presented with several texts it is possible for it to count how many joins each utterance would require. This can indicate which utterance will sound best.

4.2. Total Join Cost

The general unit selection algorithm as presented by Hunt and Black [15] consists of searching for the best path of units through a fully connected network of units. For each candidate unit in the target sequence, the candidate unit is measured in terms of its similarity to a target unit, as well as how well it could be concatenated to a previous unit. A search then occurs through all possible units to see which sequence of units has the best cost. The search is ideally done using a Viterbi search, although some systems may perform the search using some form of optimisation which is not guaranteed to return the optimal path but will still return a near-optimal path at a significantly

lower computational cost.

5. Results

A diphone unit selection speech synthesiser was trained on the CMU ARCTIC BDL speech data. The machine translation system was trained on the corpus that was created as part of the work presented in this paper. A test corpus was also created, which intentionally contained phrases that do not occur in the synthesiser's speech data. Due to the fact that the machine translation training corpus is relatively small when compared to typical machine translation corpora, the test set intentionally used some of the vocabulary present in the machine translation corpus. The test corpus contained 30 sentences. This approach seemed sufficient for a proof-of-concept experiment.

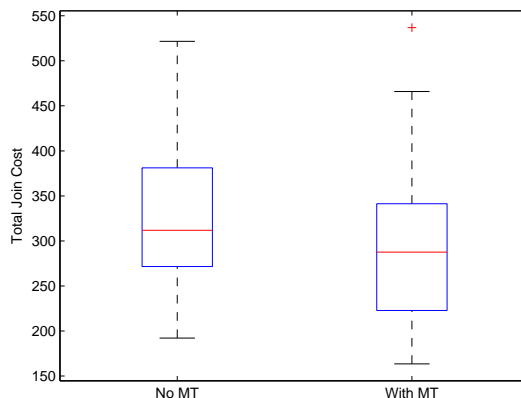


Figure 2: Total join cost for the original text compared to the best SLMT text.

Figure 2 illustrates the results from the experiment. In the majority of cases, the synthesiser produced better results when using the set of sentences in the machine translation N -best output. Figure 2 contains a box-plot of the total join costs when the system did use the MT component (*With MT*) and when it did not use the MT component (*No MT*). In 2 of the 30 utterances, the original utterance could be synthesised better than any of the machine translation N -best output. In such cases, this occurs due to the training data being relatively small, where the N -best list may only contain a single entry. This justifies the system architecture as illustrated in figure 1, where the original text is also considered equally to the N -best candidates from the machine translation.

6. Discussion

Although the experiment resulted in improved performance, a number of areas for further investigation were identified:

- While the total join cost and number of joins are suitable for predicting the final synthesis quality, they are not necessarily optimal. Further investigation is required to identify whether other parameters can improve the system.
- It is necessary to verify that the entries in the N -best list are well-formed and represent the same meaning as in the input sentence. In some situations not all entries in the N -best list are going to contain perfect alternatives.

Some N -best entries are likely to contain incorrect grammar, syntax or words. In such cases it is optimal to only use a subset of the possible entries. As the training data for the machine translation system increases the N -best list will contain better candidates.

- It is reasonable to assume that the larger the training corpus is the better results will be. While the corpus of 500 sentences is significantly larger than previous work on language generation for synthesis, it would be ideal for the MT training corpus to encompass all of the ARCTIC corpus.

It seems reasonable that addressing these points will improve performance significantly. It is also possible for the corpus to be extended beyond the size of the CMU ARCTIC corpus, where there could be several source sentences for each ARCTIC target sentence. This would allow a much larger corpus to be created while still taking advantage of the fact that the system is targeted for ARCTIC speech databases.

7. Conclusions

This paper is concerned with creating alternative input text for text-to-speech synthesisers. This approach is used to improve the performance of speech synthesisers by enabling the synthesiser to choose one of several possible sentences. Previous work on the topic have included language generation methodologies (e.g. finite state) to create the alternative text for a synthesiser.

The concept of using machine translation technology rather than conventional language generation methodologies is presented, where a same-language machine translation corpus was created to train the system. Although the corpus was designed to translate to the same language as its input, the target phrases in the corpus were intentionally the same phrases as in the synthesisers speech database (i.e. the CMU ARCTIC corpus). The motivation for using this corpus is that the same-language MT system can be used in any of the 7 freely available ARCTIC-based synthesis speech databases.

An experiment is presented which shows that the incorporation of the MT system improved the performance of the synthesiser in terms of the range of the join cost scores as well as the median join cost scores.

For future work, the authors plan to develop the MT corpus further, so that it includes the entire CMU ARCTIC corpus, as well as to investigate alternative synthesis quality estimation methods.

8. Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142. The opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Science Foundation Ireland.

9. References

- [1] R. Clark, K. Richmond, and S. King, "Festival 2—build your own general purpose unit selection speech synthesiser," *5th ISCA Speech Synthesis Workshop*, pp. 173–178, 2004.
- [2] E. Klabbbers, K. Stober, R. Veldhuis, P. Wagner, and S. Breuer, "Speech synthesis development made easy: The Bonn Open Synthesis System," *Proceedings of Eurospeech 2001*, pp. 521–524, 2001.
- [3] H. Zen, T. Nose, J. Yamagishi, S. Sako, and K. Tokuda, "The HMM-based Speech Synthesis System (HTS) Version 2.0," *The 6th International Workshop on Speech Synthesis*, 2006.
- [4] I. Bulyko and M. Ostendorf, "Efficient integrated response generation from multiple targets using weighted finite state transducers," *Computer Speech and Language*, vol. 16, no. 3, pp. 533–550, 2002.
- [5] S. Pan and W. Weng, "Designing a speech corpus for instance-based spoken language generation," in *Proceedings of Int. Conf. on Natural Language Generation*, 2002, pp. 49–56.
- [6] C. Nakatsu and M. White, "Learning to say it well: Reranking realizations by predicted synthesis quality," in *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, vol. 44, no. 2, 2006, p. 1113.
- [7] M. White, "Reining in CCG chart realization," *Lecture notes in computer science*, pp. 182–191, 2004.
- [8] J. Kominek and A. Black, "The CMU ARCTIC speech databases for speech synthesis research," *Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMULTI-03-177 http://festvox.org/cmu_arctic*, 2003.
- [9] J. Tinsley, Y. Ma, S. Ozdowska, and A. Way, "MaTrEx: the DCU MT System for WMT 2008," in *Third Workshop on Statistical Machine Translation, ACL*, 2008, pp. 171–174.
- [10] N. Gough and A. Way, "Robust large-scale EBMT with marker-based segmentation," in *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation (TMI-04)*, 2004, pp. 95–104.
- [11] F. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.
- [12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al., "Moses: Open source toolkit for statistical machine translation," in *Annual meeting-association for computational linguistics*, vol. 45, no. 2, 2007, p. 2.
- [13] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational linguistics*, vol. 19, no. 2, pp. 263–311, 1993.
- [14] N. Stroppa and A. Way, "MaTrEx: DCU machine translation system for IWSLT 2006," in *Proceedings of the International Workshop on Spoken Language Translation*, 2006, pp. 31–36.
- [15] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of ICASSP*, vol. 1, 1996.