Improving Speech Synthesis of Machine Translation Output

Alok Parlikar, Alan W. Black and Stephan Vogel

Language Technologies Institute, Carnegie Mellon University

aup@cs.cmu.edu, awb@cs.cmu.edu, vogel@cs.cmu.edu

Abstract

Speech synthesizers are optimized for fluent natural text. However, in a speech to speech translation system, they have to process machine translation output, which is often not fluent. Rendering machine translations as speech makes them even harder to understand than the synthesis of natural text. A speech synthesizer must deal with the disfluencies in translations in order to be comprehensible and communicate the content. In this paper, we explore three synthesis strategies that address different problems found in translation output. By carrying out listening tasks and measuring transcription accuracies, we find that these methods can make the synthesis of translations more intelligible.

Index Terms: Speech Synthesis, Spoken Language Translation, Machine Translation Disfluencies

1. Introduction

Speech to speech translation is a difficult task. Its pipeline typically involves the cascade of an automatic speech recognizer, a machine translation system, and a speech synthesizer. Although state-of-the-art recognizers and translation systems are quite advanced, they make mistakes. Some of the common errors we find in MT output involve (i) untranslated words, or dropped content, (ii) word reordering errors, and (iii) wrong lexical choices. Consider a poor-quality translation example: `A bird sat on of the tree.` It can be difficult to read this text fluently. If we are supposed to read it out to somebody, we would try to understand what it could have meant, and would use several speech devices such as pauses, pitch, duration, etc. to communicate the content to our listener. However, standard speech synthesizers are not designed to deal with MT errors. They are trained to read text fluently. This makes the synthesized translations particularly hard to understand.

State-of-the-art speech synthesizers are typically not as intelligible as human voices can be. Further, it has been shown [1] with the help of transcription tests, that synthesized MT output is even less intelligible than synthesized natural text. We want our synthesizer to be able to communicate the content in MT output more effectively. Prosody can be an important device of effective communication. There have been efforts [2, 3, 4] to transfer prosody information from source speech into the translated speech. This can make the output sound more natural and help intelligibility. However, if MT output has errors in it that inhibit intelligibility, we want our synthesizer to gracefully handle the disfluencies in MT output, and this is the focus of our work.

In this paper, we report three methods that we used to address the problem at hand. We used human listening tasks, and specifically transcription accuracy as the evaluation strategy for all three methods.

In Section 2, we will look at an analysis of whether inserting pauses at the right places in translations can be of help. In Section 3, we describe a method of using filler words to deal with untranslated words in MT output. In Section 4, we explore whether an n-best list of translations can be exploited to select alternative easier-to-understand hypothesis for a given translation. Towards the end, we describe our current work in progress. We are working on a more detailed technique of building a synthesizer that is trained on, and optimized for machine translations.

2. Pause Insertion and Intelligibility

The example sentence, `A bird sat on <pau> of the tree`, can be more intelligible with the pause, than without it. We analyzed whether inserting pauses at the right place can make synthesis of translations more intelligible. We asked human annotators to go through selected MT output and annotate the right placement of pauses. We synthesized the translations with and without the annotated pauses and performed a listening test to evaluate content-word-error-rate of the transcriptions.

If we have human annotated pauses on sufficient data, we can train a model that can automatically insert pauses in unseen text. Indeed, the default model in Festival synthesizer has been trained from annotated data. Now, different MT systems have different output qualities, and it may very well be true that humans would annotate the output of these different systems in disparate ways. This would mean that for every new system we build, we would either need new human annotations, or a robust learning algorithm that can adapt to varying qualities of MT output. Before working on a method that can handle different qualities of MT output, we want to investigate whether inserting pauses in the right places is useful at all. This experiment thus is like an Oracle experiment to find out—if we have human annotated pauses for MT output, can we use them for improving the speech synthesis?

2.1. Human annotations of pauses in translation output

The MT output we chose for this experiment comes from a phrase based Chinese–English translation system. We chose this language pair (instead of Portuguese–English that we will use in later sections) because we wanted to include in this study translation errors in long-distance word reordering. This system was trained on about 11 million parallel sentences and it used the Gigaword corpus for language modelling. The Moses decoder [5] was used for translation. The particular test set we used was in the broadcast-news domain and had a BLEU score [6] of 14 points with 1-reference available for evaluation.

From the test set, we selected 65 sentences that were between 10 and 20 words long. Three people were recruited to annotate pauses in the output. They were asked to mark word boundaries where they would pause, if they were reading the

text out to somebody. All three annotators are fluent English speakers with a background in linguistics.

Humans don't always agree on pause-insertion, so it is useful to determine how much they agreed on the pauses. Since we do not have a gold standard for where the pauses should be, we can only evaluate the human ability to insert pauses by comparing one person's judgement to another. To evaluate the degree to which annotators agree with each other, we analysed the data by computing the kappa statistic [7] for the annotators. We thought of the task to be a classification of whether each word boundary is a break, or not. On average, the annotators had a kappa value of 0.66. However, this substantial level of agreement can not be trusted. The problem is that most of the word boundaries are not annotated to be breaks, and rightly so. Thus, most of the agreements between annotators was on word boundaries which were not candidates for break-insertion at all. As argued by [8], this problem will affect other standard measures of inter-annotator agreement as well.

It turns out that the inter-annotator agreement is not very high. From Table 1 we can see that there is a great difference in the number of pauses different annotators inserted. The table also leads us to infer that each person agrees with at least one other annotator in about one out of two annotations. This suggests that different people have different styles of communicating content with error, and that there may not be one true pause-insertion scheme.

Annotator Set	Number of Annotations
Annotator 1	99
Annotator 2	59
Annotator 3	88
$1 \cap 2$	41
$2 \cap 3$	31
$3 \cap 1$	53
$1 \cap 2 \cap 3$	28

Table 1: Agreement between annotators on 65 sentences

2.2. Testing Intelligibility

To assess whether synthesis with pauses as humans would insert in speech is better than the the default pause insertion model in the synthesizer (which is described in [9] and optimized from broadcast news data), we synthesized both versions using Festival [10], and performed a transcription accuracy test. For this experiment, we recruited 5 subjects. All subjects are fluent speakers of English. We selected the annotations done by one of the annotators. From the 65 sentences available, we picked 20 sentences based on two criteria: (i) The annotated breaks must not be the exact set of breaks as the default breaks predicted by our synthesizer (since otherwise we would be running comparison tests on the exact utterance), and (ii) The sentences don't have hard words (such as uncommon named entities). These 20 sentences were presented in random order to each of the subjects. Each sentence was presented randomly either synthesized using human-annotated breaks, or as the default synthesis. Subjects were allowed to listen to each audio clip at most three times. They were asked to transcribe the speech.

The data collected from the subjects was processed to normalize case. The transcriptions were manually post-edited to correct typos. Subjects were asked to insert an ellipsis as a substitute for unrecognizable portion of the audio clips. These ellipsis tokens were also filtered out. We also removed function words from the transcriptions and the references. We evaluated the word error rate over these content words. Table 2 shows the average transcription error in the two experimental scenarios.

We observe that pause-insertion can yield a 10% relative drop in the word error rate. With just 20 sentences used in the listening experiment, it is difficult to test the result for statistical significance, however, this experiment could be repeated with larger sample size for meaningful significance testing to be performed.

Break-Insertion	WER
Default	30.0%
Human-Annotated	26.6%

Table 2: Content-Word Error Rate on Transcription Task

3. Handling Untranslated Words

While pause-insertion seems to help, we decided to look into specific MT issues and try and address them with specific synthesis strategies. Untranslated words can cause problems in understanding content. If the source and target languages use the same script, the synthesizer can try to pronounce the source word and often result in misleading speech. We have a Portuguese–English translation system trained on the Europarl corpus. We intend to use this system in the context of lecture translation. The test sets we translate will not be from the domain of parliamentary proceedings, and can have high frequency of untranslated words in the output. We investigated whether we can deal with untranslated words effectively.

The method we propose is to replace untranslated words with a filler sound, such as `Umm`. Thus, an example output of `It does raise problems ``la'' again` would be synthesized as `It does raise problems ``Umm'' again`. During synthesis, we also insert a short pause before and after the filler sound. The filler is synthesized at a pitch 20% lower than that of the synthesizer. No changes were made to the overall prosody of the utterance. The motivation for the short pauses and the lower pitch is that this may alert the listeners about the disfluency at that point in the utterance.

In this experiment, fillers replace all untranslated words—both content and function words. However, an untranslated word is very likely to be a content word. If a crucial content word is missing from the translation, it may be difficult for listeners to completely understand the utterance. If the untranslated word is synthesized verbatim, listeners may be taken by surprise on hearing the unfamiliar sounds, and that may cause intelligibility problems in the neighborhood of the untranslated word. From the point of view of comprehension, the untranslated word, and the filler are both equally not useful. However, the motivation of replacing the content word with a filler is to make the words in the neighborhood of the untranslated word more intelligible.

In order to evaluate whether the filler-insertion strategy did better than synthesizing just the untranslated word as-is, we looked at the transcription errors in the neighborhood of the untranslated word. We selected 20 sentences from the MT output that had untranslated words. We produced two synthesized versions: (i) using the untranslated word verbatim (system-verbatim), and (ii) replacing the untranslated word with a filler (system-filler). For each of the sentences randomly sorted, we

picked one variant of the synthesis randomly, and asked fluent English speakers to transcribe the speech. After collecting the transcriptions produced by four human subjects, we looked at the errors they made in the neighborhood of where the untranslated word was. The neighborhood was defined to be the nearest content word on either side of the untranslated word. Table 3 shows that the synthesis with fillers is better than synthesis using the untranslated words.

Strategy	Word Error
System-Verbatim	30.1%
System-Filler	24.0%

Table 3: Transcription accuracy in the Neighborhood of Untranslated Word

4. Phonetically hard-to-synthesize sentences

In some cases, MT output is hard to understand not because there are fluency issues, but just because the sentence is hard to synthesize. The generated translation may have two consecutive words such that the diphone at the word boundary may be a rare one. Our synthesis models may not handle it correctly, and this may result in speech that has bad joins. A synthesizer can give us a numeric score of how hard it was to synthesize a particular utterance. One such score that we are using here is the Unit-selection-cost. Unit selection synthesis [11] provides the means to deal with the issue of joining speech units from the voice database in an efficient manner. Every synthesized utterance has an associated join cost of unit selection. We can use this cost and pick alternative translations of MT output in the hope of producing more intelligible speech.

There has been similar work [12] to generate MT output that better suits a synthesizer. In their work, they use a same language MT system (English $\rightarrow$ English) to produce alternative translations for a given hypothesis. In our work, we are using the n-best list from the original MT system.

The workflow of our approach involves the following steps:

4.1. Identifying Sentences with bad joins

We take the corpus that we used to build our unit selection voice and extract a list of diphones present therein. For each sentence in the MT output, we find out if the sentence has any diphones unseen in the training data. If so, we classify this as having a bad join. In our experiments, we found that about 17% of the test set could be classified has having bad joins in this manner.

4.2. Choosing better alternative from n-best list

For every output sentence with bad joins, we take the n-best list of translations. We filter out those items from the n-best list that themselves have bad joins. We evaluate the sentence level METEOR [13] score of every hypothesis. We can not use human-provided reference translations to evaluate the hypotheses, since they are not available to us at test time. Therefore, as a first approximation, we choose the top-best hypothesis in the n-best list as a reference translation for scoring purposes. We use the METEOR metric instead of the commonly used BLEU metric because METEOR scores are reliable on a sentence level, whereas the BLEU scores need to be computed over the entire test set. We filter out n-best items that have a METEOR score of less than 0.98. The reason for using such a high METEOR cut-off is that we do not want to select an alternative hypothesis that is much worse than the top-best hypothesis. After this filtering step, we select the one hypothesis that has the least cost of unit selection. Here is an example from the system: The top-best hypothesis was `Now we are attacking the soul`. The boundary between the first two words was classified as a bad join. The unit selection join cost of this sentence was 665.8. We were able to replace it with the hypothesis `Today we are attacking the soul`, with a cost of only 480.5.

To evaluate whether alternative hypotheses are more intelligible, we selected 20 sentences with bad joins from our system's output. We synthesized, in one case the original output (system-topbest) and in the other case, the selected alternative (system-alternative). We performed a listening task on the two outputs with five subjects and had them transcribe the speech. The alternative hypothesis was different than the topbest in the place of the bad join. We measured the transcription accuracy for the words that were badly joined. That is, if words w_i and w_{i+1} had a bad join between them, we measured how many of the two words humans were able to correctly transcribe. Table 4 shows that using n-best list and replacing badly joined words with better alternatives can have a small improvement in intelligibility.

Strategy	Word Error
System-Topbest	28.9%
System-Alternative	24.7%

Table 4: Transcription accuracy on words with bad joins

5. Conclusions and Future work

Speech to speech translation needs to be very intelligible, in order to be useful in contexts such as lecture translation. Disfluencies in MT output make its synthesis hard to understand. In this paper we show that synthesis strategies of pause insertion, replacing untranslated words with fillers, and using alternative translations from an n-best list to tackle bad phonetic joins can have a positive impact on transcription accuracy. In the future, we would like to combine these and additional strategies and do a more detailed evaluation with larger data samples and more subjects.

The evaluation metric we have reported here is "transcription accuracy". It is not clear how well it correlates to content comprehension. We could continue using transcription accuracy as our evaluation metric, or work on designing something better. The Blizzard challenge [14] uses transcription accuracy on semantically unpredictable sentences (SUS) for evaluation. However, SUS are not meant to be understood, whereas our goal is to make MT output more understandable. An alternative method that we could use to evaluate comprehension would be to have subjects listen to synthesis of a translated lecture snippet, and then question them on the important contents.

The approaches we looked at in this paper do not deal with the larger issue in MT output: that it is locally coherent, but often globally incoherent. The ungrammaticality of the output can lead listeners into dangling garden paths, making it impossible to understand the content. To deal with this issue, we are currently working on building a synthesizer that is optimized to speak machine translations. When people are asked to read out MT output to somebody, they adopt a style that best suits

the content. We are building a synthesizer that can copy this specific style. In order to do this we take the output of an MT system, select appropriate prompts for coverage and have a human read them in the most understandable style for that content. We then build a voice using this data following our standard pipeline for building voices. In addition to features already used in training the voice, we are using scores from the translation lattice in our decoder as features. By doing so, our models can take into account how confident the MT system was somewhere in the sentence.

6. Acknowledgements

This work was supported by the Fundação de Ciência e Tecnologia through the CMU/Portugal Program, a joint program between the Portuguese Government and Carnegie Mellon University.

7. References

[1] L. M. Tomokiyo, K. Peterson, A. W. Black, and K. A. Lenzo, "Intelligibility of machine translation output in speech synthesis," in *Proceedings of Interspeech*, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.65.5216

[2] J. Adell, A. Bonafonte, and D. Escudero, "Disfluent speech analysis and synthesis: a preliminary approach." in *3rd International Conference on Speech Prosody*, May 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.74.7777

[3] P. D. Agüero, J. Adell, and A. Bonafonte, "Prosody generation for speech-to-speech translation," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.79.6949

[4] V. K. R. Sridhar, S. Bangalore, and S. Narayanan, "Factored translation models for enriching spoken language translation with prosody." in *Proceedings of Interspeech*, 2008.

[5] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation." in *ACL*. The Association for Computer Linguistics, 2007. [Online]. Available: http://dblp.uni-trier.de/db/conf/acl/acl2007.html\#KoehnHBCFBCSMZDBCH07

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[7] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, April 1960. [Online]. Available: http://dx.doi.org/10.1177/001316446002000104

[8] M. Stevenson and R. Gaizauskas, "Experiments on sentence boundary detection," in *Proceedings of the sixth conference on Applied natural language processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2000, pp. 84–89. [Online]. Available: http://dx.doi.org/http://dx.doi.org/10.3115/974147.974159

[9] P. Taylor and A. W. Black, "Assigning phrase breaks from part-of-speech sequences," *Computer Speech & Language*, vol. 12, no. 2, pp. 99–117, 1998.

[10] P. Taylor, A. W. Black, and R. Caley, "The architecture of the festival speech synthesis system," in *The Third ESCA Workshop in Speech Synthesis*, 1998, pp. 147–151.

[11] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*. Washington, DC, USA: IEEE Computer Society, 1996, pp. 373–376.

[12] P. Cahill, J. Du, A. Way, and J. Carson-Berndsen, "Using same-language machine translation to create alternative target sequences for text-to-speech synthesis," in *in Proceedings of Interspeech 2009*, 2009.

[13] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0909

[14] A. W. Black and K. Tokuda, "The blizzard challenge 2005: Evaluating corpus-based speech synthesis on common datasets," in *in Proceedings of Interspeech 2005*, 2005, pp. 77–80.