Understandable Production of Massive Synthesis

Brian Langner, Alan W Black

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
{blangner,awb}@cs.cmu.edu

Abstract

This paper explores *massive synthesis*, or synthesis of sufficiently large amounts of content such that its evaluation is challenging. We discuss various applications where massive synthesis may apply, and their related issues. We also outline factors related to those applications that affect the perceived quality and intelligibility of the speech output, and discuss modifications of those factors that can improve the understandability of the resulting synthetic speech. There is a discussion of the challenges of evaluating this work, and of the different possible metrics that may be appropriate. Finally, we show in a simple evaluation that our modifications improve the perceived quality of the synthesis.

1. Introduction

Speech synthesis is increasingly being used to deliver spoken information to people. As its use becomes more frequent, new applications which push the limits of viable synthesis become more desirable. One such application involves converting some large amount of text-based information into speech, for listening to in situation where reading is inappropriate or impossible, such as while driving or exercising – a sort of "automatic podcast generation" task. The requirements of this application are highly understandable speech that is of sufficiently high quality that people will listen to it.

The difficulty, of course, is that this task has an enormous amount of text to be synthesized, when the potential uses are examined, to the extent that it is impossible for a person, or even a group of people to realistically evaluate it before use. This is compounded by the likelihood that new content is being continuously generated, making optimizations based on prior evaluations potentially less useful.

We are calling this application task *massive synthesis* – synthesis of such a large amount of data that more typical evaluation methods are impractical because no single person will be able to listen to enough of it. The goal of our work is to identify potential problems and find solutions that maximize intelligibility and understandability with the least manual intervention possible.

Though similar, there are two different relevant concepts here. *Intelligibility*, or how well the words a synthesizer produces can be correctly recognized, is an important measure for determining the quality of the speech synthesis. *Understandability*, or how well knowledge, information, and concepts can be transferred from the speaker to the listener, is also of great importance when considering speech applications designed to provide information. For speech, understandability builds on the intelligibility, which provides a sort of ceiling for how understandable the speech will be; less intelligible speech, by default, will be less understandable as well. Both of these are important to the task of massive synthesis; though more challenging, the ultimate goal of this work is to produce understandable, and not just intelligible, synthetic spoken output.

2. Massive Synthesis

2.1. Potential Applications

We envision several possible applications that could be classified as massive synthesis. Tasks such as error reports, business case summaries, even a news reader, all have characteristics that synthesis for them would end up as massive synthesis. Producing speech in these domains, at least on a sufficiently large scale or sufficiently often, will result in too much audio to legitimately evaluate. However, the task requires high understandability in order to be a success - listening to a news story you can't understand right away is not worthwhile, and most people would not bother.

All of these domains share the difficulty of having significant amounts of content, and generally, continuous generation of new content. However, each of the above tasks has some characteristics that may simplify them. For example, error reports are likely to have a standard format and fairly closed language, and while news stories typically have a few new or unknown words per day, they are otherwise fairly normal English text. Unfortunately, that can't be said for all massive synthesis applications. Weblogs are another potential use of massive synthesis, and though they might be thought of as amateur-produced news articles, there can be some noticable differences, both in terms of the topics covered and the vocabulary used.

2.2. Example Content

2.2.1. Obtaining a Corpus

As mentioned above, there are several applications where there is more content to be synthesized than can reasonably be heard by any individual or small group. One of these, synthesis of weblogs, is interesting because of the large amount of continuously-generated content to synthesize, as well as a potentially large pool of users to listen to the synthesized output. Though each synthesized blog may have only a few listeners, the entire space here is quite large and is clearly suitable to the problem at hand.

It is fairly easy to collect data from a number of weblogs,

though there are some concerns about making the content representative of a "generic" blog. Fortunately, there has already been an effort to create a large-scale corpus of weblog content. The TREC Blog06 corpus [1], a collection of over 100,000 RSS and Atom feeds collected over 11 weeks in late 2005 and early 2006, is an ideal example of a large corpus of this sort of text. The corpus was created by downloading the homepage and all new permalinks for each feed once a week, for a total of over 750,000 collected feeds over this time period. The corpus includes an appropriate amount of spam content for realism.

It should be noted that this corpus also has a non-negligible amount of non-English text, including French, Spanish, and German, among others. As we are only concerned with English at this time, this content was largely ignored.

2.2.2. Analysis

To examine the content of this corpus, we first did a small amount of text processing to extract the content from the surrounding HTML and meta-information from the corpus distribution. Removing this non-content information resulted in a 14 gigabyte collection of blog text. This is primarily the form in which we used the corpus.

We performed a word frequency analysis to determine how weblog text differed from other English text, such as news articles. Our hope was to find "blog frequent" words that would be unlikely to be synthesized well, either in terms of quality or intelligibility. Once the "unusual" frequent words were identified, we then would determine if they were present in the lexicon, and if not, if their predicted pronuciation is likely to be accurate. For words with implausible or incorrect pronunciation, they would be flagged and targetted for improvement strategies.

In general, our analysis found most of the text was typical for English, at least with the most frequent words, which is not surprising. The most frequent but atypical tokens, *html* and *blog* appeared 27th and 28th most frequently, respectively, but otherwise the top 50 words appear to be fairly normal for English text. Even the unusual words that are frequently seen tend to be normal English words, simply used more often than, say, in the Wall Street Journal. Other common words that are mishandled tend to be acronyms that should be spelled rather than pronounced (or vice-versa), such as "FAQ", or pluralized abbreviations such as "mp3s".

It is interesting to observe the frequency of "adult" content in this corpus. Though not overwhelmingly common, "porn" and variants appear several hundred thousand times in the data. This perhaps says something about what happens when content is produced anonymously, either through weblog posts or their comments.

3. Improving the Synthesized Content

For several reasons, speech output of this content is difficult to understand. Since the usefulness of a spoken report or article is very low if it can't be understood, this is a problem that must be solved. We believe there are several issues that cause the reduced understandability, but there also are likely solutions that can be implemented to mitigate the effects.

3.1. Relevant Factors

3.1.1. Non-standard Words

Though non-standard words [2] are present in many different applications, including news articles, it seems that weblog content has a higher incidence of these, and a wider variety. News articles are generally limited to numbers and some punctuation symbols, and perhaps some foreign names or words, whereas blogs can have a far greater range of non-standard tokens. These include technical jargon (particularly when the content is related to computing technology), what is termed *leet-speek* (or *l33t5p33k*), intentional or inadvertant typographical and spelling errors (such as "the-teh", "lose-loose" or "voila-viola"), *expressive spelling* (such as "soooo..."), self-censoring of expletives (as in "#*!%"), frequent usernames and handles that are often ambiguously pronounceable, as well as similar non-standard words as in news articles. To a certain extent this is due to the lack of a formal editor reviewing the content before publication, but the fact that weblogs tend to be treated more as informal conversation than a professional publication is also an influence on these trends.

Improperly rendering these non-standard words has a significant effect on the perceived quality and intelligiblity of the synthesized speech, reducing the overall understandability For the listener to understand what they are hearing, the speech output must take into account these words, and produce something more like what a person would say when reading: "leet" rather than "el three three tee".

In many cases, these non-standard words can be grouped into classes, some of which may be quite large; for example, words containing numbers or punctuation substituted for letters. For these, it may be helpful to consider them as a foreign language of sorts, and approach learning their proper pronunciation in that fashion. Techniques as in [3] would prove useful in that situation, particularly if we can devise a system where users are capable of providing feedback while listening to the content.

3.1.2. Formatting / Text Structure

Because the bulk of the content we would be synthesizing in these applications is web-published material, there is an inherent structure embedded by use of markup languages. This structure likely will provide hints for appropriate ways to segment the content, even when presenting it as speech, rather than visually. Thus, a method that takes the text structure into account will likely be easier to understand.

Even if an individual post's content has no structure or formatting beyond simple paragraphs, the entire page containing the post almost certainly will: title, content sections, comment sections, archive links, links to other sites, ads, and other items. If the goal is to synthesize the content, removing or ignoring the parts of the structure that are unrelated or unnecessary should simplify the output and probably improve how it is perceived by the listener.

Similarly, how the text itself is formatted can be used as a guide for how it should be said. Words that are emphasized in the text should probably be emphasized when spoken. Expressive spelling, as mentioned above, is another example of text formatting signifying how it should sound when spoken. When this is done appropriately, it can make the resulting

speech sound more like how a human would speak - and more understandable.

Other formatting issues can be more problematic than helpful. Improperly rendered HTML entities, for example, are likely to be very poorly understood when synthesized, and even if they can be understood, people will be unlikely to know (or care) what `&#8211;` (or as would be heard "ampersand hash eight two one one") is supposed to represent.

3.1.3. Content Summarization

One issue that is likely to arise, particularly when synthesizing weblogs, is the problem of having very long articles, or related to that, several new articles, that should be spoken. Is it always appropriate to read very long articles in their entirety? Will condensing several new comments to the phrase "and there are 15 new comments" or similar be sufficient, or should all of those comments be heard? These questions and other similar ones do not seem to have obvious answers, but they are at the core of providing understandable speech to people.

Like most speech applications, the answers here likely depend in some way on either the user or the domain, or possibly both. Some users might prefer condensed summaries, while others insist on hearing everything. Summaries themselves can have several options. They can summarize the main article and just indicate there are comments, summarize both the article and any comments, just say how many new posts and comments there are, or something more abstract like "several pages of ravings from a barely literate teenager", for example.

There are other, more intermediate options as well, such as *subsetting* the content. That is, speaking enough of the start to make it clear what the article is about, and then waiting for the user to indicate whether the system should continue or move on to something else. In this way, the user could more quickly "browse" through the content.

Though all of these can potentially help, the most appropriate option is almost certain to depend on user preferences.

3.1.4. Phrase Boundaries

It is fairly well known that improved phrase breaks can produce significant gains in the overall understandability of synthesized speech. This effect is likely amplified with informal writing, which is less likely to have consistent punctuation or other cues for identifying phrase breaks.

In some ways, weblog content – particularly very informally written content – can end up resembling "word soup" due to a lack of punctuation and grammatical sentences. The text, then, could be thought of in the same way as the output from machine translation engines, and synthesized appropriately. Because the language in the text is "unusual", the default naïve method to determine phrase breaks will be less effective. Something more advanced, taking things such as part of speech into account, can probably provide improved breaks.

This problem is particularly noticable for non-sentence content, such as structural or navigational information on web pages. Sometimes the information provided is important, but simply reading it out without adding better prosody and phrasing makes it too difficult to understand.

3.1.5. Multiple Voices

Another possibility to improve intelligibility and understandability would be to use multiple voices, particularly with long utterances. Using different voices for different contexts - such as one for the main content, one or two others for other comments, and one for meta information or non-primary content - could provide audible cues to where content is changing. Those cues could, in turn, make the speech easier to follow, and thus, understand.

For situations where multiple different voices may not be appropriate or desired, a similar effect might also be obtained using a single voice but changing style, particularly combined with improved phrasing.

Also, though not strictly speaking a different voice, using non-speech sounds to render some text could also provide a more natural or understandable result. For example, turning "ROFL" into an appropriate laughter sound would probably be better than trying to turn that "word" into speech. Using non-speech sounds such as beeps to indicate shifts between different content can also provide a potential increase in understandability, though at the cost of decreased naturalness.

3.2. Identifying and Correcting Problems

Of course, in order to use the strategies outlined above, it is necessary to know when and where to apply them. The most likely method to find problems is to listen to the speech output, but as we have discussed above, massive synthesis is characterized by having too much content to listen to. However, evaluating *some* of it is likely to help, particularly if we select things which are more likely to have errors.

Determining whether the synthesis is correct is, in the end, always going to require someone to listen to the speech. This manual process is both slow and expensive, but necessary. To reduce the cost, we want to find as many potential problems *without* requiring a human listener as possible. There are some heuristics we can use here. First, though we want to select examples at random, we can start by selecting those examples with words not in the lexicon, such as those we flagged from the Blog06 corpus. Using this as a guide to select candidate examples for evaluations makes it for more likely to find errors.

Still, however, the amount of content to examine is likely to be large. Therefore, some method of gauging the severity of the potential errors would be ideal, in order to prioritize error correction. This is key, because trying to find and fix all errors is unlikely to be cost effective. The more optimal approach to error correction and resolution would be to concentrate on solutions that fix large classes of errors, and simple fixes that can be implemented quickly without much effort.

3.3. Evaluation

Like the problem for speech synthesis in general, it is difficult to describe a consistent, objective measure that can evaluate this speech with regard to its quality and/or understandability Typical approaches have included mean opinion scores, modified rhyme tests, semantically unpredictable sentences, and others, and in fact these have all been present in some fashion in the Blizzard Challenge [4] in previous years. However, though these approaches are suitable for comparing different synthe-

sizers or methods, they are not as helpful for demonstrating improvement for a specific task, particularly with regard to understandability. Semantically unpredictable sentences are inherently an artificial task which may or may not have any bearing on understandability for a specific application.

There are other possibilities, however. Asking listeners to rate which of two or more examples they prefer, or "like more", could be a useful dimension presuming the voice being used is the same and the quality level is consistent across different utterances. However, such an open-ended criterion may not be capturing the desired information about quality and understandability, though a large evaluation with many examples and explicit directions should be able to demonstrate improvements over a baseline. Another option would be to design a test similar to reading comprehension tests for children; by providing the content, and then specific questions about what was present, it should be possible to identify differences in understandability. The drawback to this sort of approach is the effort and cost required to design and implement it; it is likely to be far more expensive than typical synthesis evaluations.

4. Simple Evaluation

4.1. Test Examples

Given all of the issues related to how the synthesis is perceived, as well as the cost-benefit analysis to dealing with them, we implemented a number of modifications to weblog-style text. These modifications include a set of "number-to-letter" rules that effectively translate common "leet" words into pronounceable English, rules for words such as "iTunes" that use case to identify syllable boundaries, and lexical entries for several common non-standard words like "pwn" and "kthx", among others.

To test our modifications, we synthesized random comments and articles from several blogs and content sources: Slashdot [5], MetaFilter [6], LiveJournal [7], as well as a random Wikipedia article [8] and text from the Blog06 corpus. We felt these were fairly representative of the types of content that we have been working with. All examples were selected randomly, with the only constraints on the content being non-pornographic, and total playing time under 40 seconds.

Each of the examples was synthesized with a default Festival [9] installation and using our modifications. We used one of the Nitech HTS Arctic voices [10], because we felt, based on the results of past evaluations, the HTS voice would provide consistent, good-quality synthesis and reduce perceived quality differences between multiple utterances. The original content was identical between the modified and unmodified versions, though obviously the modified output might contain different phrases due to the token modifications.

4.2. Task Setup

Subjects were asked to listen to 6 different content examples, one from each method, for a total of 12 different wavefiles. For each example, they were instructed to identify which of the two waveforms they felt was better, and then rate on a scale of 1 to 5 how much better. The order of presentation was randomized, such that the same method was not used to generate the first presented wavefile for all examples.

Five subjects, all of whom are familiar with speech synthesis, took part in this evaluation. Each was given a URL that outlined the task to them, and provided the wavefiles to listen to. Subjects could listen to the examples using either speakers or headphones, but were encouraged in either case to listen to each file as few times as possible.

4.3. Results

All subjects universally preferred the modified examples to the unmodified ones. Though we expected a clear preference to emerge, it is still somewhat surprising that this preference was complete in all cases.

There was less consistent cross-listener agreement in the degree of preference, however, with some examples showing strong agreement and others almost none. In general, the average preference was fairly weak, so despite a clear preference for the modified utterances, that preference does not seem to indicate a strong improvement over the baseline. The preference scores are shown in Table 1. These results are not statistically significant due to the limited sample size.

	Min Pref	Avg Pref	Max Pref
Ex 1	1	2.2	3
Ex 2	1	3	4
Ex 3	1	1.8	2
Ex 4	1	2	3
Ex 5	2	3	5
Ex 6	2	3	4

Table 1: Degree-of-preferencescores from this evaluation.

5. Discussion

As the results from our evaluation show, it is clear that some fairly simple modifications will result in speech which is perceived as better to at least some degree. More thorough or complex changes might produce an even more obvious user preference. Our results, unfortunately, are lacking more detailed comments that would prove useful in how the speech was perceived. It may be that listeners found both examples to be poor or difficult, and one was simply "less bad" than the other.

It seems likely, based on some past evaluations and anecdotal experiences, that improved prosody will be required to have truly understandable synthesis of lengthy items. The machine-like qualities simply make it harder to concentrate on the speech, with the result being longer utterances are far more difficult to understand. On some level this is likely to be a memory issue – people have limited auditory memory [11], and even natural speech is hard to remember after hearing a long talk. However, the fact that people can routinely go to an hour-long lecture and come away having learned something suggests memory is not a valid excuse to hide behind. It seems highly unlikely that the same lecture, if delivered by a speech synthesizer, would be as well understood, or received by the audience, even with a modern, state of the art synthesizer. We would like to, with this work, be able to "close the gap" and reduce the understandability differences between synthetic and natural speech.

One area which we discussed but have not explored here is utilizing the structure of the content to help influence its synthesis and presentation. Other recent work [12] suggests this can

be helpful, both in terms of resulting understandability, but also with summarizing complex information into something more suitable for spoken output. We feel that looking further into this has high potential for improving massive synthesis.

As we discussed above in regarding evaluation, a massive synthesis application will never be able to be quality-checked in the same fashion as, say, a limited domain synthesizer. To help alleviate that issue, we believe having users of these applications provide feedback (and if possible, corrections) can provide useful improvements to the spoken output. The drawbacks to this approach are that for truly useful feedback, the users must actually care about what they are listening to, and have a want or need to understand it. This becomes tricky since those types of users, besides being harder to find in the first place, are also the ones who are least likely to put up with doing error correction as part of using a system like this. However, it is important to be have this sort of feedback mechanism to drive improvements.

Even with user-provided feedback, however, it is unclear that there is a good evaluation metric on which to judge progress. On some level, receiving fewer error reports from users would be a reasonable measure (presuming that the user base stays constant). Other metrics such as token error rate may be useful as well, but there is still likely a perceptual component that needs to be considered.

Moving forward, we envision developing a prototype system which, given the URL or other location of a document, will parse the content and provide a "podcast" to listen to – in some sense, a web browser that instead of displaying the content on a screen, renders it as speech. Given the nature of this, some collaboration with groups working on web browsers for the blind might be beneficial.

6. References

[1] C. MacDonald and I. Ounis, "The TREC Blog06 Collection: Creating and analysing a blog test collection," Department of Computing Science, University of Glasgow, Tech. Rep. TR-2006-224, 2006.

[2] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words," *Computer Speech and Language*, vol. 15, no. 3, pp. 287–333, 2001.

[3] J. Kominek and A. Black, "Learning pronunciation dictionaries: Language complexity and word selection strategies," in *HLT-NAACL*, New York City, USA, 2006.

[4] C. Bennett, "Large scale evaluation of corpus-based synthesizers: Results and lessons from the Blizzard Challenge 2005," in *Interspeech 2005*, Lisbon, Portugal, 2005.

[5] user RedBear, "slashdot.org article comments," [accessed 12-May-2007], http://slashdot.org/.

[6] user gsb, "MetaFilter article comments," [accessed 12-May-2007], http://www.metafilter.com/.

[7] user cdinwood, "LiveJournal article," [accessed 12-May-2007], http://www.livejournal.com/.

[8] Wikipedia, "Train surfing," [accessed 12-May-2007], http://en.wikipedia.org/w/index.php?title=Train_surfing&oldid=127993634.

[9] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system," 1998, http://festvox.org/festival.

[10] H. Zen and T. Toda, "An overview of Nitech HMM-based speech synthesis system for blizzard challenge 2005," in *Interspeech 2005*, Lisbon, Portugal, 2005.

[11] A. D. Baddeley, N. Thomson, and M. Buchanan, "Word length and the structure of short-term memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 14, pp. 575–589, 1975.

[12] B. Langner and A. Black, "uGloss: A framework for improving spoken language generation understandability," 2007, submitted to Interspeech 2007, Antwerp, Belgium.