A Random, Semantically Appropriate Sentence Generator for Speaker Verification

Jason Lilley\textsuperscript{1}, Amanda Stent\textsuperscript{2}, Ilija Zeljkovic\textsuperscript{2}

\textsuperscript{1}Department of Linguistics and Cognitive Science, University of Delaware, Newark, DE 19716, USA
\textsuperscript{2}AT&T Labs - Research, Florham Park, NJ 07932, USA

lilley@asel.udel.edu, stent@research.att.com, iliija@research.att.com

Abstract

We describe two systems for automatically generating English sentences, and evaluate the suitability of their output for speaker verification. The first system, SUSGen, generates grammatical but semantically anomalous sentences of controlled length, vocabulary and phonetic content. The second system, SASGen, extends SUSGen to generate a greater variety of sentences and ones that are, for the most part, semantically acceptable. We demonstrate that sentences generated by SASGen are significantly more readable and meaningful than those generated by SUSGen. Sentences generated by SASGen are not as readable or meaningful as human-generated sentences, but the additional control SASGen provides for sentence length, vocabulary and phonetic content make it more suitable for speaker verification and other voice collection purposes than harvesting from human-generated sentences.

1. Introduction

In speaker verification, the identity of a user is verified by comparing a voice sample to a previously created speaker-specific model. The voice sample can be collected by having the user read aloud a presented verification sentence. In traditional speaker verification systems, verification sentences were digit sequences or simple pass codes. Software is now becoming available that makes it possible to easily and accurately synthesize a person’s voice from a small number of speech samples [1, 2, 3]. Such software can be used to bypass traditional speaker verification systems. Speaker verification systems must become robust to this possible method of attack.

One way in which this can be done is to more carefully design verification sentences. Ideally, verification sentences should be unlimited in number, use a large vocabulary, have suitable phonetic content for accurate speaker verification, and be long enough to contain the required amount of phonetic content. The speaker must also be able to read or hear the sentence and repeat it accurately and easily. The sentences must thus be composed of common vocabulary, ideally familiar to speakers of any educational background, as well as non-native speakers of English. The sentences should also be grammatical and semantically meaningful for ease of reading.

It may seem that human-generated text (from sources such as books, magazines, newspapers, and websites) is ideal for this purpose. After all, it is almost certain to be grammatical and semantically natural. However, copyright prevents systems from using modern text, and much of the vocabulary and grammar of most out-of-copyright texts is unfamiliar to modern speakers. In addition, sentences from human-generated text may not be of suitable length or have diverse phonetic content. For these reasons, such sentences must be manually filtered. Moreover, hackers may tune voice synthesis programs using the same sources of human-generated text. The construction of verification sentences thus becomes an interesting natural language generation problem: automatically synthesizing sentences that are grammatical and meaningful, and controlled for length and phonetic content.

For example, in the AT&T SAFE mobile, multifactor authentication prototype, each user authenticates using “something you have” (the user’s mobile device), “something you know” (the answers to personal questions), and/or “something you are” (biometrics, including speaker verification). Because SAFE is designed for use on mobile devices, we don’t have to use short digit sequences as verification sentences; rather, we use the screen to display longer sentences drawn from a larger vocabulary, which the user reads aloud to provide verification samples, as shown in Figure 1.

In alpha testing of SAFE, we used sentences from the Arctic database, a collection of human-generated sentences from out-of-copyright books selected for the purpose of collecting voice data to train text-to-speech systems [4]. The Arctic sentences were insufficiently varied for speaker verification purposes, and also used archaic language that affected the usability of SAFE. We then tried a sentence generation system, SUSGen, that was created for the purpose of synthetic voice evaluation (Section 2.1). SUSGen sentences are grammatical, but by design are semantically anomalous, and this also affected the usability of SAFE. Finally, we implemented a
system, SASGen, expressly for the purpose of speaker verification. SASGen produces sentences of controlled length and vocabulary that are grammatical as well as, for the most part, semantically acceptable (Section 2.2). We present a human evaluation of SASGen in terms of readability and meaningfulness (Section 3). We demonstrate that sentences generated by SASGen are significantly more readable and meaningful than those generated by SUSGen. While sentences generated by SASGen were not judged to be as readable or meaningful as human-generated sentences, the additional control SASGen provides for sentence length and vocabulary make it more suitable for speaker verification as well as for voice banking [2, 5] and other TTS-related applications.

2. SUSGen and SASGen

2.1. SUSGen

SUSGen ("Semantically Unpredictable Sentence Generator") generates random sentences, primarily for evaluating the intelligibility of text-to-speech (TTS) systems [5, 6]. Its output is controlled by a lexicon and a grammar. The grammar consists of a list of sentence frames. Each sentence frame is a sequence of words and slot labels. The slot labels match categories in the lexicon.

At runtime, SUSGen rewrites each sentence frame, substituting slots with words of the corresponding category from the lexicon, which are randomly selected without replacement. The user can control the length of the generated sentences by specifying a maximum length in syllables. The output is a set of sentences that should be syntactically acceptable, but are likely to be semantically anomalous, such as the following:

- A thing that traveled describes the hearts.
- The brown chasm flows underneath the wolves.
- A cucumber that screamed catches the texts.

Because it is designed for TTS evaluation, SUSGen does not take semantics into account. In TTS evaluation, synthesized prompts are presented to listeners who have to transcribe the original sentence from the audio. It is important that the sentences be semantically unpredictable, so that the listeners cannot infer the words from their semantic context [7].

The current version of SUSGen, used in our experiment, defines 10 sentence frames and uses a total of 12 word categories. The lexicon holds 1234 grade-school words, most of which are taken from [8].

2.2. SASGen

SASGen ("Semantically Acceptable Sentence Generator") was developed as an extension of SUSGen to produce semantically acceptable sentences for speaker verification, voice banking and other voice collection purposes, and to have greater flexibility than SUSGen. Its generative framework is based on the ideas of GPSG [9] and HDPSG [10] and in particular borrows heavily from [11]. SASGen differs from SUSGen in two key respects. First, SUSGen's sentence frames are replaced by Chomskyan phrase-structure rules (see Figure 2). At runtime, SASGen chooses a sentence rule at random to expand, and repeatedly expands phrase labels and slot labels until it has a sequence of only words.

Second, SASGen makes extensive use of features. Features, which are properties that may be assigned to words or phrases, are commonly used in generative systems to enforce grammatical restrictions such as subject-verb agreement in person and number. In SASGen, features are assigned to words in the lexicon (Figure 3). Feature restrictions are specified in the grammar, where they may be specified explicitly or using variables (Figure 2). As SASGen generates sentences, it uses the feature unification algorithm outlined in [11] to ensure that feature restrictions are enforced.

One important feature in SASGen is the semantic feature (sem), which indicates the ontological types of nouns. Nouns are assigned to one or more twenty disjoint ontological categories in the lexicon. Predicates such as adjectives and verbs are assigned corresponding semantic features indicating the ontological categories of their argument positions (see Figure 3). Feature unification then ensures both grammaticality, and that predicates are matched with arguments of the appropriate type, as shown in Figure 2.

SASGen and SUSGen both keep track of which words have been used in the process of generating a batch of sentences, and avoid word reuse until it becomes necessary. Also, both SASGen and SUSGen allow the user...
A key distinction between SASGen and SUSGen is the semantic appropriateness of the generated sentences, to make them easier to read. To evaluate this, we ran a web-based experiment comparing sentences generated with SASGen, sentences generated with SUSGen, and sentences from the Arctic database [4], a database of human-authored sentences drawn from books and used to build TTS voices. We evaluated the sentences for readability and meaningfulness using rating scales.

3.1. Experiment Materials

We used 90 sentences drawn from a set of 1350 sentences generated by SUSGen, 90 sentences from a set of over 1750 sentences generated by SASGen, and 90 sentences from the 190 Arctic sentences. All sentences were selected by uniformly sampling the distribution induced by sentence lengths. We also used 30 sentences drawn from across all three test sets, in which we randomly reordered the words to produce sentences that were both ungrammatical and semantically anomalous (“outlier” sentences). The average sentence length was 8.5 words (range: 6 words to 13 words).

Sentences were presented in 30 blocks of 10; each block contained 3 sentences from each test set as well as one “outlier” sentence. Each block of sentences was displayed on a single web page; the order of the sentences in each block was randomized per-participant. The participant was asked to rate each sentence separately for readability and meaningfulness using a 7-point slider (from 0 to 6), as shown in Figure 4. The ratings from any participant who rated an “outlier” sentence higher than 3 (out of 6) were examined by hand. All participants were found to be making meaningful judgments.

3.2. Participants

Participants were recruited using Amazon Mechanical Turk. Only turkers who passed a simple English fluency screening test were allowed to participate. Each participant was given one hour and paid 10 cents for rating one block of sentences, and could complete multiple unique blocks of sentences. Each participant rated an average of 13 blocks of sentences (range: 1 to 30) with each block rated by 10 different participants.

3.3. Results

The means and standard deviations of the readability and meaningfulness judgments for the Arctic, SASGen and SUSGen sentences are given in Table 1. Pairwise t-tests were run, using Bonferroni correction, and not assuming equal variances. The difference between each pair of systems was highly significant for both readability and meaningfulness ($p < .001$).
### Table 1: Experiment results: means and standard deviations for readability and meaningfulness judgments

<table>
<thead>
<tr>
<th>Item</th>
<th>Readability</th>
<th>Meaningfulness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSGen</td>
<td>3.02 (1.71)</td>
<td>0.66 (1.14)</td>
</tr>
<tr>
<td>SASGen</td>
<td>3.70 (1.78)</td>
<td>3.09 (2.08)</td>
</tr>
<tr>
<td>Arctic</td>
<td>4.55 (1.58)</td>
<td>4.52 (1.77)</td>
</tr>
</tbody>
</table>

We conclude that SASGen produces sentences that are more readable and meaningful than the sentences that SUSGen produces, while still giving more control to the system developer than human-generated sentences. We infer that the output of SASGen will be more acceptable to users of speaker verification, voice banking and other voice collection and TTS-related applications.

## 4. Conclusions and Future Work

In this paper, we presented SASGen, a system for automatic generation of semantically acceptable sentences for speaker verification that permits control of the vocabulary and length of the generated sentences. We compared SASGen to SUSGen, a system for automatic generation of semantically unpredictable sentences, and to human-generated sentences from the Arctic database. In an empirical evaluation, SASGen achieved higher scores for readability and meaningfulness than SUSGen, while giving system developers more control than when using human-generated sentences.

In our evaluation, the output of SASGen was found to be significantly less readable and meaningful than the human-generated Arctic sentences. It may not be possible to automatically generate sentences that are as readable and meaningful as human-generated text. However, we will continue to expand and tune the lexicon and grammar of SASGen, using publicly available resources such as FrameNet [12]. We may also use a language model built from human-generated text to prune SASGen sentences based on “naturalness”.

We plan to conduct a speaking experiment to further evaluate the readability of SASGen sentences. Participants will be presented with sentences on a screen and made to immediately read them aloud. Pre-speaking pauses, as well as the number of disfluencies and errors produced, will be measured. We are also currently running evaluations of SASGen in the AT&T SAFE prototype. SASGen gives improved speaker verification accuracy and security compared with using digit-only prompts or a fixed set of human-generated sentences.

## 5. References


