AN EMPIRICAL TEXT TRANFORMATION METHOD FOR SPONTANEOUS SPEECH SYNTHESIZERS

Shiva Sundaram, Shrikanth Narayanan
ssundara@usc.edu, shri@sipi.usc.edu
Department of Electrical Engineering-Systems and Integrated Media Systems Center
University of Southern California
3740 McClintock Ave., Los Angeles. CA 90089.

ABSTRACT

Spontaneously spoken utterances are characterized by a number of lexical and non-lexical features. These features can also reflect speaker specific characteristics. A major factor that discriminates spontaneous speech from written text is the presence of these paralinguistic features such as filled pauses (fillers), false starts, laughter, disfluencies and discourse markers that are beyond the framework of formal grammars. The speech recognition community has dealt with these *variabilities* by making provisions for them in language models, to improve recognition accuracy for spoken language. In another scenario, the analysis of these features could also be used for language processing/generation for the overall improvement of synthesized speech or machine response. Such synthesized spontaneous speech could be used for computer avatars and Speech User Interfaces (SUIs) where lengthy interactions with machines occur, and it is generally desired to mimic a particular speaker or the *speaking style*. This problem of language generation involves capturing general characteristics of spontaneous speech and also speaker specific traits. The usefulness of conventional language processing tools is limited by the availability of training corpus. Hence and empirical text processing technique with ideas motivated from psycholinguistics is proposed. Such an empirical technique could be included in the text analysis stage of a TTS system. The proposed technique is adaptable: it can be extended to mimic different speakers based on an individual's speaking style and filler preferences.

1. INTRODUCTION

In [1], a modification for including certain spontaneous speech markers was shown to have a positive impact on the naturalness of a concantenative synthesizer [Figure 1]. The necessity of a new text processing technique was also discussed. In this paper we implement a text processing technique to be used with the text analysis stage of such a synthesizer. While an end-to-end spontaneous speech synthesizer needs integrated implementation of various modules such as dialogue management and language generation, this paper focuses on a specific aspect of the problem in concept to text generation. The goal here is text transformation. This involves transforming plain sentences into sentences annotated with spontaneous speech features: a type of *preprocessing* for text analysis of a TTS system.

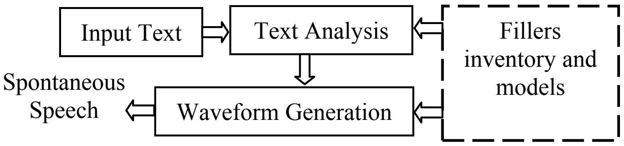

Figure 1. Modification to a conventional TTS system

The *Fillers Inventory and Models* block expands this scope of conventional text analysis. It facilitates the inclusion of spontaneous speech features, which would be later synthesized, during waveform generation. It is assumed during text analysis that the corresponding inventory of spontaneous speech features is available for waveform generation. The required Text Analysis/transformation performed by the "machine" is illustrated:

Input: I am sorry the number you have dialed does not exist
Transformed Text: I am sorry [BREATHE IN] the [UH] [PAUSE] number you have dialed does not exist

Due to the large variability in the features of spontaneous speech, the goal of this paper has been constrained to the inclusion of filled pauses and audible breathing, which are major traits that express speaker-dependent characteristics in speech. A major motivation behind implementing empirical techniques is the limitations in existing corpus: they carry annotations that predominantly cater to the analysis needs of speech recognition. It is also difficult to have large amounts of *consistent* data for a single speaker: in terms of domain of the speech, such as continuous monologues or conversations.

The analysis and offline training for this research was done on 190 minutes of transcribed Spontaneous Lecture Monologue, with 23,725 of "clean" text and 26,618 words including the spontaneous speech features tags. The lectures were available at the University of Southern California (USC), through the Distance Education Network (DEN). They were transcribed at the Speech Analysis and Interpretation Laboratory at USC.

The paper has been organized as follows: in section 2 the features that characterize spontaneous speech are discussed. In section 3 their analysis and implementation techniques to the language transformation problem is discussed. Section 4, furnishes details about the implementation. Results and issues pertaining to spontaneous speech generation and its evaluation are discussed in Section 5.

2. SPONTANEOUS SPEECH FEATURES

The details about the features in spontaneous speech were provided in [1]. They are broadly discussed here for clarity.

Characteristics of spontaneous speech can be classified into:

- Paralinguistic Cues: Falsetto, Whisper, Creak, laughter giggle, cry/sob etc.
- Disfluency Patterns: words such as *and, okay, oh, so*, and *well,* repetitions and filled pauses: *uh* and *um*.
- Reflexes: *Throat Clearing*, *sniff/ gulp*, *tongue clucking lip smacking* and *breathing.*

Falsetto, Whisper, Creakiness are qualities of the voice and involve prosodic modification of speech. Laughter, giggle and crying etc are types of voice qualifications. While it is possible to use the same statistical techniques (discussed later) to include these voice qualifications, they are not considered here due to corpus limitations.

Reflexes are usually involuntary and are sometimes used by a speaker to “make a point”, indicate the beginning of a sentence, or signal the introduction of a new idea. For example tongue clucking sometimes indicates disagreement. They also affect the fluency of speech. Reflexes are sparsely distributed in real speech, but could be a prominent trait in a particular speaker.

Breathing is always present in spontaneous speech. Most breathing instances are audible especially when speech is captured or heard through a microphone/transducer. Occurrence of breathing in speech is a function of amount of air in the lungs, rate of talking, the type of words spoken, pauses, tiredness, hesitation, sureness or emphasis in the point being made and the voice quality of the speaker. All these factors span a wide spectrum of analysis and reasoning, that is complex and not completely necessary for the specific problem of text transformation. However since breathing improves naturalness of synthesized speech it is possible to implement a set of simple heuristic rules for its insertion.

Disfluency patterns consists of words that are usually present at phrase boundaries and generally signal a change of idea, agreement response, or a pause for thinking etc. They typically trigger the usage of filled pauses such as *uh* and *um*. Repetitions include certain phrases such as “oh yes”, “so basically” etc. Most speakers have a preference to certain phrases. Filled pauses (also known as fillers) are of importance to this research because they are the prominent features that discriminate spontaneous speech from speech generated by reading out of a book. The following section discusses the psycholinguistic aspects of their usage.

3. ANALYSIS

3.1 Psycholinguistic Analysis:

The use of fillers has been analyzed in [2]. It is argued that fillers are not just ungrammatical utterances but are a part of spoken language because their usage is intentional and have a structure to it. The points of interest from their research and are listed below, with illustrations to their application in the current problem.

- Speakers use *uhs* and *ums* to indicate delays in speech, where *uh* is used for a minor delay and *um* for a major delay. They are also caused due to problems in formulating an utterance or phrase.
- *Uhs* and *Ums* not only differ in dialogues but they differ in monologues also, they are techniques to hold the floor for the listener. There is also variability in the language used.
- Their usages vary amongst speakers, and a particular speaker may also have his/her preferences, similar to their preferences in vocabulary.
- Occurrences of fillers differ in the domain of the speech act. In formal addresses, speakers tend to use little or no fillers, contrary to their language during informal occasions.

In [3] pauses for breathing had been analyzed where spontaneous speech-breathing instances was compared to readings from written text. It has been suggested that breath inhalation is an automatic passive activity that a speaker participates in during speaking and it occurs at breaks in spontaneous speech, irrespective of whether it was in at a grammatical juncture. Hesitant speech has more breathing instances at non-grammatical breaks than fluent speech. Thus it is also possible to have control over the fluency of the synthesized speech. It was also found that the speech-breathing rate was a consistent characteristic of an individual.

The above discussion shows that the traits of a speaker are also reflected in the filler utterances during spontaneous speech, voice being the major characteristic. There are different domains where speaker dependent and speaker independent generation is required or where the same speaker is required to talk in different addresses. The fillers are likely to occur during an idea change and when the speaker is pausing for articulation or when the speaker wishes to intentionally hold the listener’s attention.

3.2 Statistical Analysis:

The analysis of the speaker-dependent data indicates that *uhs* and *ums* only occur after certain words, some of them are listed below.

Certain words preceding Uhs: a, about, and, had, it/its, of, so, that, the, to, there.

Certain words preceding Ums: the, but, and, have, it/its, of, okay, that.

The n-gram probabilities aid in gathering information about filler preferences and their usage, and this brings out the speaker dependent traits in spoken language. n=1 (unigram) and n=2 (bigram) are of significance for such an analysis. Common observations of the speaking styles of a range of speakers suggest that unigram probabilities are of interest because speakers have a preference for the type of fillers. The target speaker in our analysis has a preference for using *Uhs* than Ums. Bigram probabilities suggest that *Ums* and *Uhs* are not arbitrarily used for pausing, but have a purpose of an intentional delay in the language, they occur only after a handful of words as listed above. This complies with the points brought out in section 3.1. All the analysis in this work was done using the Cambridge-CMU LM toolkit [4].

While n-gram models help to understand the local distributions of the fillers, it is also relevant to have class-based analysis. It was observed that it is more likely for fillers to occur only after words that belong to certain POS classes such as conjunctions, determiners, than noun words. These words mark the beginning/end or agglutinate phrases with related context. Class-based analysis is also more intuitive (in terms of the semantics in the text) and hence, more relevant. Implementing these heuristic rules based on observations can be done using Finite State Machines, particularly: *Finite State Acceptors*. They provide a flexible architecture to extract and compare “patterns” in the language. It was also convenient to encode heuristics of spoken language as FSAs.The FSAs were implemented using AT&T’s Finite State Machine Tools [5].

Since breathing is a passive process during speech, it is beyond the scope of rules based on statistical modeling or n-gram models due to the complex nature of its occurrence. The insertion rules for text transformation are not unique, and any other set of rules or observations could be implemented. It must be pointed out that text transformation for a Spontaneous speech synthesizer comprises of simple sentences of 10 to 15 words length. And a “good enough” solution to the breathing insertion problem could be rule-based.

The mean number of words between inhalations was about 7.46 words (with a standard deviation of 4.9). The occurrence of breathing could be classified as occurring at a grammatical/ ungrammatical juncture. In the corpus it was found that the occurrence at grammatical junctures was around twice as much as occurrence at ungrammatical places. These are similar to the findings in [3]. Most of the grammatical occurrences were at the beginning of the sentence and before conjunctions. Occurrence at ungrammatical places occurred when there was a change of context or as a *silence-delay* in speech. Common occurrences were before phrases such as “so anyway” “okay well” “but also”. Other common occurrence was before the filler *Um*. It is unlikely that such phrases are present in an input sentence for transformation. For the problem specific to this work, it is sufficient to have heuristic rules that insert breathing instances at grammatical places of the input text.

The text transformation problem, by nature, is a problem of decision. The transducer needs to decide whether a particular filler/breathing instance is to be inserted following a given word

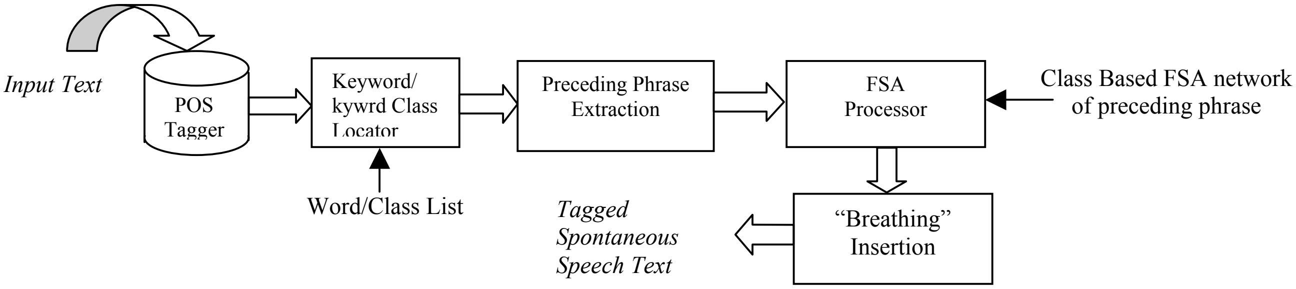
Input Text
POS Tagger
Keyword/ kywrd Class Locator
Word/Class List
Preceding Phrase Extraction
FSA Processor
Class Based FSA network of preceding phrase
"Breathing" Insertion
Tagged Spontaneous Speech Text

Figure 2. Implementation of Text Transformation for Spontaneous Speech Generation. This figure corresponds to the modification in the text analysis in a TTS system.

in a sentence. It does not arbitrarily insert these instances, as it has learnt from word level n gram model that there are only a handful of such words. Further, it compares the patterns in the input text to the patterns extracted from the offline speaker specific data, to strengthen its decision. Thus it can transform a given length of text to something that *would have been spontaneously spoken* by the target speaker, while using the same set have *given* words.

4. IMPLEMENTATION AND EXPERIMENTS

The implementation is done in two stages: *offline training* and *online processing*. Offline training deals with congregation of words that have a high probability of preceding a filler; and building class based finite state acceptor networks from phrases that precede the fillers. Online processing comprises of part of speech (POS) tagging of the given text, comparing it with the offline training models and finally deciding whether to insert filler or breathing instance after a word in the given text. The overall process is shown in [Figure 2]. The implementation is discussed in detail.

The text in the training corpus was annotated with tags for instances of the spontaneous speech features. However the *Ums* and *Uhs* were treated as regular words.

Tags were used for Breathing, Pauses (Short and Long), Instances of Tongue clucking and Laughter/giggle.

While Ums and *Uh*s were used in most sentences, it is not possible to have reliable n gram estimates for other features such as Tongue Clucking and Laughter. In most cases, it is easier and more relevant to implement inclusion of these instances as a set of rules derived from observation limited to the application. For example, to implicitly convey happiness or a funny context, it is necessary to include instances of giggling or short laughter.

Offline training/building models:

The offline training for insertion of fillers in text is summarized in the following steps. Breathing occurrence is dealt with separately. The offline training is represented as single arrows in [Figure 2].

1. Create a list of words that are most likely to precede a given filler. This can be done by pruning a list of bi gram estimates of various word combinations with the fillers, arranged in a descending order. Separate lists are required for each *Uhs* and *Ums*. Let the lists be represented as $\boldsymbol{Wlist_{Uh}}$ and $\boldsymbol{Wlist_{Um}}$
2. Extract the phrase preceding each filler instance, and encode the corresponding word-classes into an FSA. This shall be the FSA corresponding to the individual *Uh/Um* filler. Let them be denoted by $\boldsymbol{F_{uh}(i)}$ and $\boldsymbol{F_{um}(k)}$. Where the subscript denotes the corresponding filler and $\boldsymbol{i}$ and $\boldsymbol{k}$ correspond to the $\boldsymbol{i}^{th}$ and $\boldsymbol{k}^{th}$ occurrence of *Uh* and *Um* respectively.
3. Build a complete FSA network by the union of $\boldsymbol{F_{uh}(i)}$ and $\boldsymbol{F_{um}(k)}$ for all $\boldsymbol{i}$ and $\boldsymbol{k}$. Standard minimization algorithms can also be applied to reduce the size of the FSA. Minimization has been done using the *fsmminimize* tool provided with the AT&T FSM toolkit.

The result of training and building would be a single FSA network for *Uh* and Um each, representing phrase structure before them and two lists of most likely words preceding a given Filler, $\boldsymbol{Wlist_{Uh}}$ and $\boldsymbol{Wlist_{Um}}$

Note that $\boldsymbol{Wlist_{Uh}}$ and $\boldsymbol{Wlist_{Um}}$ need not only contain the list of most likely words preceding the fillers; it can also be comprised of the corresponding part of speech classes of the words.

Online processing:

The online processing occurs when a sentence/paragraph is entered for synthesis. The online processing is the modification that is required along with the text analysis stage. The implementation is summarized below:

1. Pre-process the input text using a POS tagger. The Edinburgh Language Technology Group's HMM based POS tagger was used [6].
2. The tagged sentence is searched for the keywords that are in ***Wlist*** .
3. If there is a match, the phrase preceding the keyword (phrase of length 3 to 4 words) is extracted. A class-based phrase consisting of the classes of the words in the extracted phrase is constructed.
4. If the FSA network in the FSA processor accepts the extracted class-based phrase, then the corresponding filler is inserted after the keyword.
5. Steps 2 through 4 are repeated separately for insertion of *Uh* and *Um*.

Breathing Insertion:

In the 190 minutes of data, there were a total 682 breathing instances. There were frequent parts in the monologues, which contains laughter and others where the speaker talked very fast. Marking breathing instances in these parts were difficult, and was part of the laughter or words. This is the main source of error in statistical analysis of breathing instances. The steps for breathing insertion are as follows:

1. Let L be the length of the input sentence. If L is approximately 12 words then continue with steps 2, 3 and 4. Else execute step 2 and stop.
2. Insert an instance at the beginning of the sentence.
3. Using the phrase chunks defined by the POS tagger insert another instance in-between the phrases before a conjunction such as *and, but, because.* Grammatical insertions can also be made based on punctuation marks such as a *comma*. (Steps 2 and 3 follow grammatical insertion of breathing instances)
4. Randomly insert an instance before the filler *Um*. It was observed that a breathing instance before *Um* was highly likely as compared to before a *Uh*

5. RESULTS AND ASPECTS OF EVALUATION

Some examples of an input sentence and corresponding transformation are shown below.

1. **Mary had a little lamb but its fleece was not white**.
   [BREATHE IN] Mary had a little lamb but its [UM] fleece was not white.
2. **Then they began to ponder what they should do with the leftovers and they thought it would be nice to bring them to their beloved Mary**.
   [BREATHE IN] Then they began to ponder what they should do with the leftovers and [UH] they thought it would be nice to bring them to their beloved Mary.
3. **You would substitute this into here and phi sub X X of M**.
   [BREATHE IN] you would substitute this into here [BREATHE IN] and phi sub X X of [UH] M.
4. **Might as well talk about it right now.**
   [BREATHE IN] Might as well talk about it [UM] right now**.**

The input test sentences belonged to two different types: training domain and non-training domain. The training corpus predominantly belonged to expression of technical terms and concepts as illustrated in sentence 3 and 4 as shown above. The algorithm works well for both types of input sentences. This is a result of using class based FSA network for comparison. However under certain clean test sentences taken from the training corpus itself, the algorithm inserts fillers where it was not originally present, although it was not out of place. For the various test sentences, both Uh and Um never occurred together. The use of *keyword/keyword class locator* in the online processing limits the occurrence of these fillers in unexpected and wrong places. By changing the list in ***Wlist*** it was possible to vary the occurrences of the fillers.

True evaluation of such a text transformation system remains obscure, due to unavailability of a complete spontaneous speech synthesizer (free of artifacts) for subjective tests, limitations of speaker-independent and speaker-dependent corpus belonging to the *same* domain. Spontaneous lecture speech was chosen for the work in this paper as a matter of convenience since other spontaneous speech data based on conversations (such as the SWITCHBOARD corpus) do not suffice for speaker dependent data, and due to its ready availability. While it can be seen that the discussed implementation could be easily modified for speaker-independent spontaneous synthesis, this was not covered in the synthesis due to unavailability of data.

It is imperative to highlight the inherent anomaly present in the evaluation of spontaneous speech. If real speech of humans were to be evaluated, how would it be done? Variability in spontaneous speech even with a single speaker is a major factor that impedes in defining metrics for evaluation. Aspects such as *naturalness*, *fluency*, *intelligibility* and *spontaneity* are useful in defining a set of *required qualities* that are suited for a given application. For example, in public speaking or formal occasions, naturalness, fluency and intelligibility are desirable and conventional TTS systems already have the demanding intelligibility. On the other hand, fluency in conversative speech is not important, whereas naturalness is required.

Finally it is necessary to mention that domain dependent subjective tests are a type of a valid metric for evaluation. However it is feasible only by having a complete end-to-end spontaneous speech synthesizer, as mentioned earlier.

6. CONCLUSION AND FURTHER DISCUSSIONS

This paper has proposed an empirical technique to mimic spoken language used by a target speaker, and it can be further generalized for speaker independent speaking traits. Furthermore, subjective results [1] support the fact that including spontaneous speech features improves the naturalness of synthesized speech, making it vital to have text transformation for TTS systems. Data collection for spontaneous speech is difficult and domain dependent limiting the amount of available data. Hence adaptable, empirical techniques are essential for problems related to text transformation discussed here. Through this it is also possible to implement a single TTS system which is *suitable* for different applications: from simple text reading for the blind to air travel reservation systems. Since speech production in machine is imperative for Human-Computer interaction, it is necessary to see it as an interface that is capable of producing natural speech, predominantly in the language level. Recently developing multi-modal machine interaction techniques could also benefit through synthesis of spontaneous speech. The problem of *text-to-speech* in the literal sense is only a particular aspect of it, and has already been solved to a satisfactory level.

It has been shown in psycholinguistic research that a human speaker use fillers intentionally to set-up a situation to communicate more effectively. Natural Language Generation (NLG) for dialogue systems could include these features, not only to make machine-to-human interaction seem more natural, but also to transfer information implicitly pertaining to the context in the synthesized speech. It also helps the listener to easily decode the information being propagated. Loss in speech fluency (due to generated hesitation) and intelligibility (repetitions and fillers) are concomitant with improved naturalness and spontaneity as proposed here. Subjective testing of a complete interaction/information retrieval system based on spontaneous synthesis is necessary to confirm this premise. Simply inserting spontaneous speech features causes unintentional hesitations in speech.

The specific problem of text transformation constrains the overall problem of annotation for spontaneous speech synthesis, however it also limits the validity and flexibility for an accurate insertion of these features. Having heuristic rules for insertion may seem canned in terms of language generation or semantic values, but even in real speech, from a human speaker, most of the utterances are canned due to individual manners and the continuously adapting speaking style of a person in a given situation. It may be feasible to capture the spectrum of speaker style variations through methods similar to the implementation in this paper.

Completely mimicking the speaking style of a target speaker can be achieved only by working at the signal level, consequently providing greater flexibility in the language generation/ transformation problem. Domain dependency limits the training corpus available for speaking style. Thus empirical methods are suitable in this sense, contrary to large amounts of training-data required for conventional modeling-generation techniques. Again, due to the greater flexibility provided by voice-similarity in mimicking, the question of convergence of empirical techniques might be overlooked.

REFERENCES

1. Shiva Sundaram, Shrikanth Narayanan "*Experiments in the synthesis of spontaneous monologues*". IEEE 2002 workshop in Text to Speech Synthesis, Santa Monica CA.
2. H.H Clark, Jean Foxtree "*Using Uh and Um in spontaneous speaking*" Cognition 2002.
3. F.Goldman Eisler, "*Psycholinguistics: Experiments in spontaneous speech*", Academic Press Inc. (London) Ltd. 1968.
4. CMU-Cambridge Language Modeling ToolKit
   http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html
5. AT&T Finite State Machine ToolKit 3.7
   http://www.research.att.com/sw/tools/fsm/
6. POS tagger, LTG, Edinburgh University.
   http://www.ltg.ed.ac.uk/~mikheev/software.html