Detection of Repetitions in Spontaneous Speech in Dialogue Sessions

Mert Cevik 1, Fuliang Weng 2 and Chin-Hui Lee 1

1 School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA
2 Research and Technology Center, Robert Bosch Corp., Palo Alto, California 94304, USA

mert@ece.gatech.edu, Fuliang.Weng@us.bosch.com, chl@ece.gatech.edu

Abstract

We present a study on detecting repeated speech segments in consecutive utterances in a user session of a dialogue application. Such repetition patterns often carry important information of a user’s expectation from or frustration about the dialogue system and therefore provide a source of clues for the system to adjust its dialogue strategies when such repetitions occur in order to improve system performance. We propose a recursive dynamic-time warping based pattern comparison algorithm with no fixed end-points to find similar parts within the two utterances, called original and correction utterances. Candidate reference patterns are generated from the correction utterance by an unsupervised segmentation scheme. When no prior information about the position of the repeated parts is used, each reference pattern is compared with the original utterance from the beginning to the end. Experiments are conducted on 190 utterances of spontaneous speech from simulated dialogue sessions. The proposed algorithm achieves detection and rejection rates of 80.3% and 85.2% for the repeated and non-repeated parts, respectively. When segmentation information provided by the recognizer in the dialog system is incorporated into preparing the reference patterns, the rates are increased to 93.8% and 91.1%.

Index Terms: repetition, spontaneous speech, dialogue session

1. Introduction

Repetitions in dialogue sessions in spontaneous speech constitute important information about the content. This information can be used in various applications in speech processing such as improving the performance, determining and labeling the content, and processing for further applications. Some parts of an utterance, called original utterance, may be repeated within the contents of a subsequent utterance, called correction utterance. These repeated parts may be either discrete words or phrases. These words may also be repeated not only in the same way, but also with some additional prefixes or suffixes indicating variations. Without any information about the order and position of the words, the task is to determine whether some parts of the original utterance is repeated in a correction utterance, and to extract these repeated parts. For application designers of dialog systems they often don’t have access to the internal code of the recognition subsystem being used in order to integrate repetition detection algorithm into the recognition algorithm. Signal processing techniques outside the recognition subsystem is often needed to perform the desired operation. Once repetitions are detected, the dialogue system can use this information to adjust dialogue strategies in order to improve performance of a system.

Previously, repetition detection of correction utterances has been studied for various applications. In [1], a method that is based on dynamic time warping (DTW) and N-best hypotheses overlapping measure is used to detect partial repetition of misrecognized speech for a car navigation system. This system takes only the misrecognized parts of the original utterance as the correction utterance, and judges whether the correction utterance is included in the previous one. In [2], the method is extended for a spontaneously spoken dialog. The system detects the common parts between the original and correction utterances where the repeated content may appear at any position in the correction utterance. Although the correction utterance does not have to be part of the original utterance, and the repetition may appear at any position, the algorithm depends on the order of the words in the two utterances, with U1 indicating the original utterance and U2 representing the correction utterance.

U1: Please find a French restaurant with a five-star rating in Justin.
U2: I’m looking for a five-star French restaurant in Justin.

Figure 1: An example of an utterance pair to be compared

In a spontaneous dialog as shown in Figure 1, some parts of the original utterance are repeated in the subsequent correction utterance. The repeated parts have an arbitrary order and position. We assume that we have no access to the internal code of the speech recognition subsystem that is used in the dialogue application design. The boundaries of the repeated parts are extracted by utilizing only signal characteristics. Proper segmentation is necessary to generate candidate reference patterns that can be used for pattern comparison. We first propose an unsupervised segmentation process which partitions the correction utterance into fragments that have boundaries close to the actual words. We then design a dynamic-time warping (DTW)-based pattern comparison algorithm to compare the candidate reference patterns with the original utterance. As required by the situation, pattern comparison with DTW is executed without the knowledge of fixed end-points. A recursive DTW algorithm is needed to find the best matching parts that are considered as the repeated parts in the original utterance.

The rest of the paper is organized as follows: In Section 2, the automatic segmentation process, the DTW algorithm and the repetition decision mechanism are described. In Section 3, an experimental evaluation of the proposed method is presented. Finally in Section 4 we summarize our findings.
2. System Description

2.1. Overview

The original and correction utterances are taken as the inputs to the system. The essential components of the system are segmentation of the correction utterance, feature extraction, pattern comparison and decision rule. The flow of the algorithm is shown in Figure 2. Before processing the speech signals, end-point detection is performed to remove the non-speech parts at the beginning and end of the utterances by using the algorithm in [3].

The correction utterance is segmented according to the short-time energy thresholds found in end-point detection process to generate appropriate reference patterns. The original utterance is taken as the test pattern. Feature vectors are extracted from the test and reference patterns. Each reference pattern is compared with the test pattern from the beginning to the end by the DTW algorithm, and the distance scores are computed [4]-[5]. The parts of the test pattern that matches with the reference pattern with a distance score below a threshold are selected as candidates of repetitions. These candidates of repetitions are further classified as repetitions or non-repetitions according to the length and distance score.

![Block diagram of the system](image)

**Figure 2: Block diagram of the system**

2.2. Segmentation

The correction utterance is partitioned into word-like reference patterns. The first step of segmentation is extracting the silence regions that separate the speech segments. The statistical characteristics of the background silence are extracted by the end-point detection algorithm. A short-time zero-crossing rate (IZCT) threshold is computed for the unvoiced speech as well as a short-time energy threshold (ITL) for voiced speech. These thresholds are used to identify the frames as voiced/unvoiced speech or silence throughout the correction utterance. The speech segments between the silence regions are extracted. According to a length threshold, these segments are further segmented or saved as reference patterns. The speech segments that are longer than the length threshold are segmented according to the minima of the short-time energy spectrum. This process generates reference patterns that have a certain maximum number of frames as shown in Figure 3.

![Segmentation of the correction utterance into reference patterns](image)

**Figure 3: Segmentation of the correction utterance into reference patterns**

2.3. Feature Extraction

Mel-frequency cepstral coefficients (MFCC) are used as feature parameters. Feature vectors (13-dimension) are generated with a rate of 100 frames/sec. Before computing the distance scores between the feature vectors, variability of the cepstral coefficients and their significance should be considered. It is stated in [6] that suppression of high cepstral coefficients leads to a more reliable measurement of spectral distances. A cepstral liftering procedure is applied to control the non-information bearing components of the cepstral coefficients for reliable discrimination of the sounds.

A raised sine liftering function is defined by Eq. 1 [6] for weighting the cepstral coefficients where $L$ is the length of the lifter, and $h$ is chosen as $L/2$. The liftering process reduces the sensitivity of the features without altering the formant structure of the speech.

$$w(n) = \begin{cases} 1 + h \sin(\frac{\pi n}{L}) & n = 1, 2, \ldots, L \\ 0 & n \leq 0, n > L \end{cases} \quad (1)$$

2.4. DTW Algorithm

The original utterance which is taken as the test pattern is compared with the reference patterns by DTW. For a given a test pattern such that

$$T = \{T(1), T(2), \ldots, T(M)\}$$

where $T(m)$ is the spectral vector of the input test signal at time $m$, and a reference pattern such that

$$R_v = \{R(1), R(2), \ldots, R(N)\}$$

where $R_v(n)$ is the spectral vector of the $v$th reference signal at time $n$, a non-linear time alignment function of the form $m = W(n)$ is solved to minimize the accumulated distance

$$D = \sum_{n=1}^{N} d(R(n), T(W(n))). \quad (2)$$

Euclidean distance is used as the frame-wise local distance $(d(n, m))$. With the given local continuity constraints and slope weights the accumulated distance is

$$D(m, n) = \min \left\{ \begin{array}{ll} D(m-2, n-1) + d(m-1, n) + d(m, n) \\ D(m-1, n-1) + d(m, n) \\ D(m-1, n-2) + d(m, n-1) + d(m, n) \end{array} \right. \quad (3)$$

Each DTW computation starts from a certain frame on the test pattern. Since the ending frames for the best matching parts on the test and reference patterns are not known, an $\ell \times \ell$ end-point free area is introduced. The end-point free area depends on the length of the $v$th reference pattern $N_v$. The dimensions of the area are determined experimentally by...
The repetition detection algorithm is shown in Figure 4. Combined to form the final repetition segments. An overview of short-time zero-crossing rate and segment duration, and they are further processed according to the short-time energy level, as potential candidates of repetitions. These matching parts with a distance score below the threshold between every possible reference and test pattern pair, and saves the frames that correspond to $D_{\text{min}}$ are the ending frames on both the test and the reference patterns that give the best matching result.

Pattern comparison by DTW is executed recursively for all possible reference and test pattern pairs starting from the beginning to the end of the test pattern as shown in Algorithm R1.

Algorithm R1: Recursive DTW computation ($R, T$)

```plaintext```
for starting frame ← 1 to $M - N$
    compute $D_{\text{min}}$
    if $D_{\text{min}} < \text{threshold}$
    then save the frames as repetition
```

The recursive DTW computation is executed for each reference pattern. A maximum reference pattern length ($\text{max.ref.length}$) is defined. The reference patterns that are longer than $\text{max.ref.length}$ are curtailed by skipping frames from the beginning until the length decreases to $\text{max.ref.length}$. The algorithm is shown in Algorithm R2.

Algorithm R2: Repetition detection algorithm ($R, T$)

```plaintext```
for each reference pattern
    do {compute local distance matrix
        if $N < \text{max.ref.length}$
            then RECURSIVE DTW COMPUTATION ($R_i, T$)
        while $N > \text{max.ref.length}$
            generate $R_i$ from $R_t$
        else do RECURSIVE DTW COMPUTATION ($R_i, T$)
        $N ← N - 1$
```

The algorithm computes the accumulated distance scores between every possible reference and test pattern pair, and saves the matching parts with a distance score below the threshold as potential candidates of repetitions. These matching parts are further processed according to the short-time energy level, short-time zero-crossing rate and segment duration, and they are combined to form the final repetition segments. An overview of the repetition detection algorithm is shown in Figure 4.

3. Experiments

The proposed algorithm is evaluated under three different conditions for the reference pattern boundary information: The oracle condition, which uses manually partitioned words as reference and test patterns; Algorithm 1, which uses the candidate boundary information for the reference patterns from the ASR output and Algorithm 2, which uses the reference boundary information from the proposed automatic segmentation algorithm.

A set of 190 recordings that include repeated words and/or phrases is used for training and testing the system. The data set is organized in 25 subject groups, each of which is spoken by one male or female speaker. Training set consists of 130 utterances from 15 subject groups, each having 8-10 utterances, and test set consists of the remaining 60 utterances from 10 subject groups, each having arbitrary number of utterances. The utterances in each subject group in the training and test set is spoken by one particular speaker. The system is trained by computing the accumulated distance scores between each word (extracted by using the ASR output) within the utterances to determine the distance threshold. The distribution of the accumulated distance scores for the matching and non-matching words are shown in Figure 5.

![Figure 5: Distribution of accumulated distance scores](image)

Within each subject group, each utterance is taken as the test utterance, and compared with the other utterances to find the repeated parts. The accumulated distance score is normalized by the number of test frames in each computation and the dimension of the feature vectors, thus a global distance measure is computed to evaluate the performance under various experimental conditions. The performance of the system is evaluated by detection, miss and false alarm rates. The detection rate, defined by Eq. (5), is the rate of the correctly detected (repeated) words.

$$\text{Detection-rate} = \frac{\# \text{ of correctly detected words}}{\# \text{ of desired words in test utterances}}. \quad (5)$$

Performance of the system depends on the accuracy of the boundaries of the reference segments. The oracle condition is tested by using the results from 76 computations of original-correction utterance pairs. In oracle condition, utterances are partitioned into words manually, and constrained DTW is executed by fixed end-points for each pair of words in the test and reference pattern. A detection rate of 97.6% for the repetitions and 98.1% for the non-repeated parts were achieved.

By using the proposed automatic segmentation algorithm, boundaries of the segments are determined according to the short-time energy levels. Since the hypothesized boundaries do not correspond to the actual word boundaries, the DTW-based pattern comparison is executed recursively to find the possible repeated segments within the signal. To evaluate the performance of the system without the ambiguity of the reference pat-
tern boundaries, reference word boundary information is generated from an automatic speech recognition (ASR) subsystem. The reference patterns that correspond to the words in the correction utterance are used in the DTW algorithm to observe the improvement of the performance.

The experiments were conducted with the experimental setup shown in Table 1 for Algorithm 1 and 2. Each utterance in a subject group is taken as the test utterance, and compared with the other utterances within the subject group, so that a total of 478 pairs of utterances were used to evaluate the performance of the system. There are 3457 words in the test set, which contains 1165 repeated (within the same subject group) words and 2292 non-repeated words. In the experiments for Algorithm 2, 935 of the repeated words were correctly detected as repetition, whereas 341 of the non-repeated words generated false alarms. The system achieves a detection rate of 80.3% for the repetitions and a rejection rate of 85.2% for the non-repeated parts by using the reference patterns generated by the proposed automatic segmentation algorithm. The performance of the system was improved by using more accurate word boundary information taken from the ASR output for the reference patterns in Algorithm 1. In the experiments, 1093 of the repeated words were correctly detected as repetition, and 204 of the non-repeated words generated false alarms. A detection rate of 93.8% for the repetitions and a rejection rate of 91.1% for the non-repeated parts were achieved.

Performance of the system for both algorithms and the oracle condition is listed in Table 2. Figure 6 plots the detection-rejection trade-off values of the system for Algorithm 1 and Algorithm 2.

As it depends on more precise boundary information for the reference patterns, Algorithm 1 achieves a higher detection rate than Algorithm 2, however there is still a margin for improvement between Algorithm 2 and the oracle. The equal error rates in Figure 6 are 92.7% and 81.2% for Algorithm 1 and 2, respectively.

4. Conclusion

In this paper, a DTW-based algorithm is proposed to detect the repeated parts within two utterances in a dialogue session. The candidate reference patterns are generated from the correction utterance. An accumulated distance score is computed for every possible combination of the reference and test patterns to extract the exact boundaries of the repeated parts. The patterns in the original utterance that match the reference pattern with a distance score below the threshold are selected as the repetitions. By using all combinations of pairs of utterances, a detection rate of 80.3% is achieved for the repetitions as well as a correct rejection rate of 85.2% for the non-repetitions. Experiments show that improvements in the segmentation scheme increase the detection rates. More precise word boundary information enhances the pattern comparison computations, thus the detection accuracies increase. Using candidate word boundary information which is extracted from the ASR output increased the rate of detection for repetitions and non-repetitions to 93.8% and 91.1%, respectively. Future work involves in improving the segmentation process to find the actual word boundaries in the utterances and applying discriminative features for segmentation and DTW distance computations.

5. References


