Prosodic Entrainment in an Information-Driven Dialog System

Andrew Fandrianto, Maxine Eskenazi

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

{fto,max}@cs.cmu.edu

Abstract

This paper explores entrainment of two speaking styles, shouting and hyperarticulation, in an information-driven spoken dialog system. Both styles present difficulties for automatic speech recognition. We describe and evaluate the system’s detection and reaction mechanisms for these speaking styles, which involve deploying appropriate dialog-level strategies. The three strategies tested do induce style change more effectively than the baseline of no strategy. This can translate into both better recognition and a higher chance of task success. Shouting is found to be more amenable to modification than hyperarticulation and the effect of the former on system performance is more profound.

Index Terms: spoken dialog, entrainment

1. Introduction

It has been shown that in human-human dialogs, each speaker adapts to the other [1]. It has also been shown that people can adapt to an automatic spoken dialog system [2] [3]. In the same way, a spoken dialog system can also adapt to a human speaker [4]. This leads us to explore whether we can now combine both sides of the dialog, human and machine, to adapt to one another in the same dialog.

There are many linguistic levels at which adaptation can occur. Most frequently explored is the adoption of words that the other party uses: lexical entrainment. This paper examines another level: prosody. In unpublished studies, when trying to induce changes in rhythm, intensity, or pitch as separate entities, we had seen little success. We postulate that since prosody is present at many levels of speech and each level is closely linked to the others, we should not have addressed one level of prosody separately from the others. To approach this holistically, this paper will examine changes in speaking style. Two distinct speaking styles are explored, both of which are encountered by the Let’s Go! spoken dialog system [5]: shouting and hyperarticulation. In order to change these styles, the dialog system must first automatically detect when a caller is shouting or hyperarticulating. Then the goal of the dialog manager must change from simply slot filling for bus timetable requests to also changing the caller’s speaking style.

Our ultimate goal is to increase recognition accuracy as well as dialog success. Shouting and hyperarticulation have both been shown to decrease automatic speech recognition (ASR) performance [6] [7]. Aggravating this is the finding that during a failing dialog, callers tend to increase the frequency of shouting and hyperarticulation [8]. We believe that guiding the caller back to “unmarked” speech should increase system performance. We define “unmarked” here as speech that does not contain shouting or hyperarticulation.

2. Approach

The Let’s Go spoken dialog system currently deployed in the City of Pittsburgh gives bus timetable information to the general public. Callers express themselves in many ways, and there are distinct styles that some callers use from the beginning of the dialog (like shouting). Our goal is to get a caller to go from shouting to “unmarked”, non-shouted speech and from hyperarticulating to “unmarked”, non-hyperarticulated speech. This implies testing several different strategies in the dialog manager (DM) to determine which strategy, if any, can successfully change caller behavior. Note that we believe this implies adaptation on both sides of the dialog, since the system adapts its goals to deal with what this detected. This method requires changes in the DM, the natural language generator (NLG), and the synthesizer (TTS). It also requires a detector to automatically find these speaking styles in real time.

2.1. Shouting and Hyperarticulation Detectors

2.1.1. Building the detectors

We use a subset of features extracted by openSMILE [9]. Features were extracted from an expert labeled set of 2259 call turns in the Let’s Go! data (from 2009 and 2010) [3]. We use two support vector machines (by LibSVM) created with automatic parameter selection, one for each class. The openSMILE and SVM combination has been widely used in the past for paralinguistic feature detection [10]. We used the following features:

- **Acoustic features**
  - fundamental frequency (F0) range
  - fundamental frequency (F0) average
  - intensity
  - harmonic-noise ratio (HNR)
  - pause duration
  - pause frequency

- **Dialog-level features**
  - system+ASR confidence
  - explicit confirm repetition

This feature set was found to have higher accuracy over a balanced test subset from 2259 turns than a broader set of features, which included MFCCs, pitch direction, and the number of turns in the dialog parsed as “No”.

Features were normalized for each dialog by taking the preceding “unmarked” style turns and finding the ratio between them and the current turn. If the first turn was not unmarked, we had no previous turns on which to base our decision. Therefore, we used a non-normalized model with a higher probability threshold of 85% to determine that class. It is rare for a caller to begin by shouting or hyperarticulating, but since this may occur, it is covered. The performance of the non-normalized model is lower than the normalized one because of natural variations in
call characteristics, such as line noise, phone type and encoding, environment, etc.

At runtime, the two parallel detectors (one for shouting and one for hyperarticulation) each gave a probability for its target class. The system chose the class with higher probability over a manually imposed threshold of 55%, which we had found to be optimal. In the rare case of equal probabilities, hyperarticulation was chosen simply to break the tie.

2.1.2. Performance
The detectors attained 80% accuracy for shouting and 70% for hyperarticulation, calculated on a balanced, expert-annotated test set. Three experts annotated the corpus for shouting and hyperarticulation. The Cohen’s $\kappa$ of agreement of the experts was 0.46 on average. Note that accuracy accounts for the cases of false positives and negatives.

2.2. Effects of Style
With our detector, we can analyze speaking styles and their correlation with certain features, both at the turn and dialog level.

<table>
<thead>
<tr>
<th>style</th>
<th>WER</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>unmarked</td>
<td>15.1%</td>
<td>-</td>
</tr>
<tr>
<td>shouted</td>
<td>23.2%</td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>hyperarticulated</td>
<td>16.9%</td>
<td>0.048</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ASR confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>unmarked</td>
</tr>
<tr>
<td>shouted</td>
</tr>
<tr>
<td>hyperarticulated</td>
</tr>
</tbody>
</table>

Table 1: Turn level effects of style on WER and ASR confidence

2.2.1. Turn level
We examined two features on the turn level which were affected by style (Table 1), looking at crowd-transcription of earlier Let’s Go data [3]. ASR word error rate (WER) significantly increased on turns with shouting or hyperarticulation: $+53.6\%$ and $+11.9\%$ respectively. So it seems that, at the turn level, shouting has a greater effect on WER. We also see that ASR confidence significantly decreased in these conditions: $-7.9\%$ for shouting and $-13.7\%$ for hyperarticulation.

2.2.2. Dialog level
We examined three features on the dialog level which were affected by style (Table 2). First, total dialog length in terms of time was significantly increased when shouting or hyperarticulation was present. The number of turns significantly increased as well. We determine a successful dialog to be one that returns some information about the bus the caller had asked for. The presence of shouting statistically significantly lowers the estimated success rate, while hyperarticulation’s presence does not.

<table>
<thead>
<tr>
<th>style present</th>
<th>avg duration</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>unmarked only shouted</td>
<td>107.9 sec</td>
<td>-</td>
</tr>
<tr>
<td>hyperarticulated</td>
<td>146.4 sec</td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>hyperarticulated</td>
<td>160.6 sec</td>
<td>$\approx 0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>avg turn count</th>
</tr>
</thead>
<tbody>
<tr>
<td>unmarked only shouted</td>
</tr>
<tr>
<td>hyperarticulated</td>
</tr>
<tr>
<td>hyperarticulated</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>avg success</th>
</tr>
</thead>
<tbody>
<tr>
<td>unmarked only shouted</td>
</tr>
<tr>
<td>hyperarticulated</td>
</tr>
<tr>
<td>hyperarticulated</td>
</tr>
</tbody>
</table>

25-50% of turns | avg success | p vs present |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>shouted</td>
<td>65.9%</td>
<td>0.014</td>
</tr>
<tr>
<td>hyperarticulated</td>
<td>56.9%</td>
<td>$8.1 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

50%+ of turns | avg success | p vs 25-50% |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>shouted</td>
<td>60.6%</td>
<td>0.12</td>
</tr>
<tr>
<td>hyperarticulated</td>
<td>33.3%</td>
<td>0.038</td>
</tr>
</tbody>
</table>

Table 2: Dialog level effects of style presence on duration, turn count, and estimated success

33%.

2.3. Dialog Strategies
In order to enable the system to act to change these two styles, we developed three different system strategies. The overall goal was to choose general types of strategies that were independent of a specific style, so if we want to modify some other style at a later time, one of the strategies we already have tested could be used. The first strategy, “explicit”, tells the caller exactly what the system wants them to do. The second strategy, “implicit”, uses the idea of entraining the caller: the system demonstrates the desired behavior. For example, the system speaks more quietly just after someone has shouted. The third strategy, “backoff”, changes the subject (say from the departure stop to the bus route number). See Table 3.

<table>
<thead>
<tr>
<th>type</th>
<th>shouting</th>
<th>hyperarticulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>prompt to speak more quietly</td>
<td>prompt to speak “normally”</td>
</tr>
<tr>
<td>implicit</td>
<td>lower volume</td>
<td>increase synthesis speaking rate</td>
</tr>
<tr>
<td>backoff</td>
<td>switch slot objective</td>
<td>switch slot objective</td>
</tr>
</tbody>
</table>

Table 3: Strategies by speaking style

2.3.1. Explicit
Having the system simply tell the caller to change their style can be unpredictable, since it could be interpreted in several ways (more formal, less emotional, etc).

For shouting, the system says: “I’m sorry. I cannot understand loud speech very well, please speak softer.” This grounds the system’s difficulty and gives an explicit instruction.

For hyperarticulation, the system says: “I know you’re trying to help me understand, but I work better when you speak
This grounds and instructs the caller, in a more general manner than for shouting. The word "naturally" may be interpreted in several ways and is not necessarily the opposite of "hyperarticulation". However, the use of an infrequent word such as "hyperarticulate" could have been counterproductive.

We did not find more appropriate specific language here.

2.3.2. Implicit

This strategy assumes that the caller will detect and entrain to the system’s changed prosody when the system demonstrates the desired behavior. Two callers in a conversation mutually entrain speaking styles [11]. Thus, this strategy assumes that the caller will detect and entrain to the system's changed prosody when the system demonstrates this strategy. A given strategy is never used twice in the same dialog. For shouting, the volume of the TTS is lowered by 20% after the first turn and another 20% after the second turn, if shouting continues. It is lowered by an additional 20% if necessary on the third turn for a total of 60% reduction. In an unpublished study, we found that teachers often lower their voices before saying something important. This was done to get students to attend to what they are saying. It is also possible that lowering the volume may make the system less understandable in a high noise environment (perhaps the reason that the caller is shouting). This could in turn prompt the caller to reframe to a quieter place, which should give the ASR a less noisy signal to process.

For hyperarticulation, when examining hyperarticulated speech in the Let’s Go data, we observed that our callers tended to slow their speech down rather than to strive to attain articulatory targets. Thus the system increases its speaking rate: after the first turn, it increased by 25% and after a second turn by another 25% for a total of 50% if the behavior continued. We chose faster speech as an entrainment example, since it is simple for a caller to detect and imitate and is in parallel to the strategy for shouting (entraining our caller to behavior that is the opposite of their present behavior).

2.3.3. Backoff

As mentioned above, “backoff” consists of switching slot objectives. Switching slots provides a “context switch” for the caller. Backoff has been shown to be useful in the dialog, especially when the system repeatedly cannot recognize something the caller had said [12]. This strategy should change the context sufficiently for the caller’s speaking style to be altered.

2.3.4. Strategy progression after failure

If, during a given dialog, a strategy fails to change the caller’s speaking style, the system randomly chooses from a strategy that has not yet been used. For example, starting with all three strategies available – explicit, implicit, and backoff, we detect a caller shouting and randomly choose one of the three, say implicit. If, in the next turn, the caller shouts again, we proceed as described above until no TTS modification stages are left. If the behavior starts again later in the dialog, explicit or backoff will be chosen. Lastly, if the second strategy has failed, we pick the remaining one. In the end, if all strategies fail to modify the caller’s style to unmarked, the system does not implement any other strategy. A given strategy is never used twice in the same dialog.

2.3.4. Strategy progression after failure

If, during a given dialog, a strategy fails to change the caller’s speaking style, the system randomly chooses from a strategy that has not yet been used. For example, starting with all three strategies available – explicit, implicit, and backoff, we detect a caller shouting and randomly choose one of the three, say implicit. If, in the next turn, the caller shouts again, we proceed as described above until no TTS modification stages are left. If the behavior starts again later in the dialog, explicit or backoff will be chosen. Lastly, if the second strategy has failed, we pick the remaining one. In the end, if all strategies fail to modify the caller’s style to unmarked, the system does not implement any other strategy. A given strategy is never used twice in the same dialog.

3. Local Strategy Evaluation

This section presents and compares the relative efficacy of each strategy. Our baseline has none of the aforementioned style change strategies: the sole goal is slot filling. This study ran for 28 days in February and March 2012, obtaining 1528 dialogs.

3.1. Significance and detector caveats

Due to residual error in each style detector and the assumption that they will both make similar detection mistakes in each strategy case, we will determine significance by assuming a binomial distribution over the baseline “no strategy” performance mean.

3.2. Reverting to “unmarked”

This section examines the rate at which the caller changed styles to “unmarked” after the system deployed a strategy. An example of a successful “reverting-to-unmarked” sequence would be:

1. System: “Where would you like to leave from?”
2. Caller: “McKeesport”, Style: hyperarticulating
3. Strategy: [increase speaking rate]
4. System: “The 64, am I right?”
5. Caller: “No, McKeesport”, Style: unmarked

Figure 1 shows two groups: hyperarticulation and shouting. We found that the explicit strategy was the most effective for both shouting and hyperarticulation, with +74.3% (p ≈ 0) and +17.9% (p = 3.1 × 10^{-4}) improvement over baseline respectively. For shouting, the implicit strategy was second, with +40.6% (p ≈ 0), followed by backoff, at +18.2% (p = 2.2 × 10^{-4}) over the baseline. For hyperarticulation, implicit and backoff performed similarly with +9.1% (p = 0.025) and +9.2% (p = 0.023) respectively.

The callers may be trying to make themselves clear on the specific word they believe the system has trouble with. However for the backoff case, the system changes slots to fill at the next turn. Because of the slot change, the caller may no longer need to hyperarticulate.

The efficacy of the implicit strategies suggests that an entrainment strategy does work, but it is not the most successful one. It should be noted that it would be misleading to directly compare rates between hyperarticulation and shouting since they perform differently with respect to the baseline.

Figure 1 also seems to indicate that hyperarticulation is an ephemeral, less enduring style. One turn of hyperarticulation is followed by another of “unmarked” speech 65.3% of the time. On the other hand, shouting is followed by “unmarked” speech only 38.9% of the time.
4. Effects of the Strategies

Here, we compare the differences between our system with strategies implemented, and the same system without strategies. Data for Table 4 is taken from a month of Let’s Go dialogs for each system. The no strategy column takes 1180 dialogs from the system without strategies implemented. Other columns use the same 1528 dialogs examined in this paper. We show strategies deployed in pairs, because there were not enough dialogs where only one isolated strategy was used. Table 4 shows that our strategies gave a slight, but not significant improvement on estimated dialog success rate, except for the explicit/backoffs situation for shouting, which was indeed effective.

<table>
<thead>
<tr>
<th>style present</th>
<th>no strategy</th>
<th>explicit</th>
<th>implicit</th>
<th>explicit backoff</th>
<th>implicit backoff</th>
</tr>
</thead>
<tbody>
<tr>
<td>unmarked only</td>
<td>75.0%</td>
<td>-</td>
<td>90.0%</td>
<td>82.5%*</td>
<td>90.0%</td>
</tr>
<tr>
<td>shouting</td>
<td>70.6%</td>
<td>75.2%</td>
<td>90.0%</td>
<td>82.5%*</td>
<td>90.0%</td>
</tr>
<tr>
<td>hyperarticulation</td>
<td>71.5%</td>
<td>69.0%</td>
<td>70.2%</td>
<td>82.5%*</td>
<td>82.5%*</td>
</tr>
</tbody>
</table>

Table 4: Comparing systems with and without entrainment (all figures are per-dialog); a star (*) indicates statistically significant over no strategy.

Strategies attenuated instances of shouting and hyperarticulation from 25.8% to 23.0% (p \( \approx 0 \)). Although it would seem that a strategy like backoff would increase the number of turns in the dialog, the use of the strategies actually reduced turns on average by 1.4 (out of an overall average of 14 turns per dialog). It did, however, increase total dialog duration. Both turns and time differences between the two systems are well within their standard deviations (\( \sigma \)) of 8.8 turns and 71 seconds respectively. However, these differences are still significant with (p \( \approx 0 \)). High \( \sigma \)'s arise from natural variability within calls.

There are a variety of reasons why the strategies may have been more effective for shouting than for hyperarticulation. When we look back on the effects of style (section 2.2), hyperarticulation does not affect WER as much as shouting. Therefore, mitigating hyperarticulation would lower overall WER to a lesser extent. Also, the style detector performs with 10% less accuracy on hyperarticulation than on shouting; this brings noise into the data, which may cloud potential signs of improvement. Most importantly, hyperarticulation is a loose class definition both for our detector and for the caller. The term could be interpreted using a variety of acoustic and prosodic elements by different callers. This may be reflected in our experts’ low Cohen’s \( \kappa \) average when hand-annotating styles for the detector. Finally, unlike shouting, hyperarticulation may be a subconscious behavior. People could be unaware that they are using it, and they may not be able to consciously control the extent of their articulation.

5. Discussion

Results show that all three strategies had a significant local effect that changed the caller’s speaking style. The explicit strategy was the most effective: telling someone outright had an immediate effect. The implicit strategy had mixed results, performing better for shouting than for hyperarticulation. Our initial intuition that backoff provides a “context switch” capable of altering speaking style is supported by the data.

We found reinforcing evidence that shouting and hyperarticulation lowered ASR performance (section 2.2). Even though we show that our strategies reduce these styles, these strategies had a relatively small effect on task success, except for the explicit+backoff for shouting, which was effective (section 4). They did, however, reduce the number of turns per dialog, while increasing dialog duration slightly.

6. Conclusions

We have shown that callers are locally responsive to strategies that change their behavior. We found that shouting can be changed more than hyperarticulation and that the former has a stronger negative effect on WER and estimated dialog success. Since implementing these strategies only slightly increases the duration of the dialog, we believe that their inclusion can improve system performance. The use of a variety of strategies, each with varying performance, may increase the perceived naturalness of the spoken dialog system.

7. Acknowledgements

This work was funded by NSF grant IIS-0914927. The opinions expressed in this paper do not necessarily reflect those of NSF. The authors would like to thank Suhkada Palkar for her collaboration in creating our style detector.

8. References