HYPERARTICULATION DETECTION IN REPETITIVE VOICE QUERIES USING PAIRWISE COMPARISON FOR IMPROVED SPEECH RECOGNITION

Ranjitha Gurunath Kulkarni, Ahmed El Kholy, Ziad Al Bawab, Noha Alon
Imed Zitouni, Umut Ozertem, Shuangyu Chang

{raguruna,ahelkhol,ziadal,noalon,izitouni,umuto,shchang}@microsoft.com
Microsoft, Sunnyvale, CA, USA

ABSTRACT

Automatic speech recognition systems can benefit from cues in user voice such as hyperarticulation. Traditional approaches typically attempt to define and detect an absolute state of hyperarticulation, which is very difficult, especially on short voice queries. We present a novel approach for hyperarticulation detection using pairwise comparisons and demonstrate its application in a real-world speech recognition system. Our approach uses delta features extracted from a pair of repetitive user utterances. Results show significant improvements in WER (word error rate) by using hyperarticulation information as a feature in a second pass N-best hypotheses rescoring setup.

Index Terms— Hyperarticulation Detection, Human-Computer Interaction, Speech Recognition Rescoring

1. INTRODUCTION

There have been a lot of scientific efforts on hyperarticulation detection in audio signal processing field. Understanding special cues from a speaker's voice has generated a lot of research, but with limited practical applications. With the increase in number of applications for automatic speech recognition (ASR), the need for understanding the meta information in the speaker's voice rather than just the spoken words is becoming important. This work targets one such trait in speaker's voice which is hyperarticulation, to improve the speech recognition of voice-enabled personal assistants. The common use-cases of personal assistants are web-search, command and control, navigation and many other applications on mobile phones and desktop computers. The way users interact with such a system is being studied extensively in order to better understand the performance quality and to improve it further.

One of the common behaviors of users of these systems is query reformulation. When users are not satisfied with the results shown by the personal assistants, they tend to repeat or paraphrase their queries in order to get better results. There could be multiple reasons leading to the reformulation. In this work, we target those that follow from errors in speech recognition. These cases can be identified for targeted improvements to the ASR system. Unlike other work on reformulation that is based on lexical and semantic similarities of consecutive queries, we consider comparative traits in voice such as hyperarticulation as an evidence for mis-recognition. Previous work suggests that speakers speak more clearly and slowly after evidence of mis-recognition [1]. This paper talks about new methods for detecting hyperarticulation in voice query reformulations. The specific questions we try to answer through this work are: 1. How accurately can we detect signs of hyperarticulation given the audio of consecutive query pairs? 2. Can we use the predictions of our hyperarticulation models to improve speech recognition in a second pass rescoring setup?

2. RELATED WORK

Hyperarticulation detection is a challenging task for humans and computers alike. Not knowing the user's normal speaking style makes it challenging to come to a conclusion that the user is actually hyperarticulating. Most current approaches are designed to classify a single utterance irrespective of the previous one(s) which could lead to poor classification performance. Our approach alleviates this problem by focusing on a pair of user utterances spoken consecutively within a short time frame and have certain lexical overlap. Pairwise comparisons help in two ways. When we collect reference labels for training purposes, it is easier for a human judge to do comparative labeling as opposed to giving absolute labels on a subjective question [2]. It also helps in creating a non-speaker-specific model as every user has his or her own way of emphasizing or articulating speech. We extract comparative features from the two repetitive voice queries that we believe help identify changes in articulation in user voice. Several studies on the changes in the acoustic signal of repeated speech in human-computer interaction have been performed [1, 3, 4, 5]. They show changes in fundamental frequencies, duration and loudness. The work in [3] collects data by artificially simulating speech recognition errors and asking the users to repeat their utterances until they get it right. They show that there were significantly higher chances of clear-speech adaptations when the system made errors. The work in [3, 4] present studies on hyperarticulation behavior as a function of error correction. Another group of research, on repetitive speech analysis such as [5], also come to conclusions that repeated voice queries have significantly different characteristics as opposed to original queries which added to the motivation of our work.

In the same vein as these efforts, our goal is to predict hyperarticulation accurately and use it to improve user experience with the speech recognition systems. However, our work deals with real data where users are interacting with a real speech recognition system. In contrast to [3, 4] efforts, we use detailed features with pairwise comparison of aligned word-segments between two utterances by the same user. Although there are many publications in this area, to our knowledge, this is the first published work that tackles the problem of hyperarticulation detection in the context of real user sessions and utterance reformulation. Compensation for hyperarticulation modeling articulatory features has been proposed in [6] in a controlled experiment. However, our work is the first of its kind to successfully integrate this knowledge into a real-world speech recognizer.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

3. DATA

3.1. Data Sampling

In our approach, we use human labeled data for hyperarticulation detection. The data is sampled from Cortana, Microsoft's virtual personal assistant on mobile and desktop devices, and then labeled using crowd sourced judges according to strict guidelines where consensus is sought. We collected around 5000 speech query pairs from user sessions while naturally interacting with the personal assistant. In this section, we discuss our criteria in sampling the data, the annotation guidelines and then the quality of the crowd sourcing judgments.

3.1.1. Sampling Criteria

As we described earlier, we are working with consecutive speech queries in order to obtain pairwise comparison judgments. Apart from the condition that they were consecutive query pairs issued by the same user, the following conditions were applied to increase the coverage of the signal of hyperarticulation. The conditions exclude query pairs that are not direct repetitions or very significant reformulations. The three conditions that we applied are as follows.

1. The time difference between the two queries was ≤ 2 mins as in [7].
2. The two queries are phonetically similar. We used three different ways to identify these cases.
   (a) Metaphone edit distance < 2 : Levenshtein edit distance is measured after converting both the query texts to metaphones [8].
   (b) Phonetic edit distance < 2 : $PD = latticeDist(L1, L2)$ is the distance between phonetic lattices L1 and L2. $L_i = g2p(query_i)$ where $query_i$s are word sequences and L_is are phonetic lattices. the $g2p()$ function is a weighted finite state transducer that accepts grapheme (letter) sequences and produces a weighted lattice of phonemes. Paths through this lattice are possible pronunciations of the input letter sequence. $g2p()$ is trained on (grapheme sequence, phoneme sequence) pairs.
   (c) Second query's 1-best recognition result was one of the candidates in the N-best hypotheses of the first query's recognition.

3.2. Annotation Guidelines

To obtain annotations for our sampled query pairs we utilized a crowd sourcing tool. Through the tool we presented the audio for every query pair followed by questions.

We asked the judges to answer whether or not both queries are trying to achieve the same task. This was asked in order to make sure one query is truly a second attempt of the other. We then asked the judges to compare what they heard in the first query to what they heard in the second query. They were asked to look for acoustic cues of hyperarticulation in any part of the first query compared to the second query or vice versa, or there was no difference. We wanted to make sure to include both directions—the first query compared to the second and the second compared to the first—to avoid biasing the judges to the direction we ultimately care about. This was done through a simple three choice question.

3.3. Annotation Quality

In this section, we discuss the inter annotator agreement. Each query pair in the data sample was judged by 3 to 5 judges in order to reach a consensus of at least 3 judges like in [9]. Otherwise, we do not consider the pair for training our models.

We compute Fleiss' Kappa [10] for inter-annotator agreement. As expected, Fleiss' kappa takes the high value of 0.82 when the judges are checking if the second query is related to the first query. This is due to our biased data selection towards related queries. However, judging if there is hyperarticulation in the second query was not easy and the judges had low Kappa value of 0.38. To overcome the low Kappa value, we train our models only on data with majority consensus (at least 3 judges).

4. HYPERARTICULATION DETECTION

4.1. Approach

We apply our detection approach on a pair of consecutive user utterances that match the criteria discussed above. We extract features that help identify changes in the articulation from the first utterance to the next one.

4.1.1. Utterance Level Features

For each utterance we extract prosody and spectral features from the speech signal as described in Table 1 using internal Microsoft tools. The min/max for F0 obtained by Getf0 [11] were 50/500Hz. Loudness here, is an energy estimate derived from the log-Mel features. These features were calculated for frames of 100 ms with a step of 10 ms. We average those features over the word-segments and retain each segment's average value for the feature. The time segmentation information is computed using forced-alignment technique of the audio to the speech recognition hypothesis. The duration of each word-segment is added to the segment level features to make a total of 17 features per segment.

Table 1. *Features computed for each segment of speech.*

Features Group	Description	Dimension
Prosodic	F0, Loudness, Pitch acceleration	3
Spectral	Log Filter-bank Energies	13
Time	Duration	1
Total		17

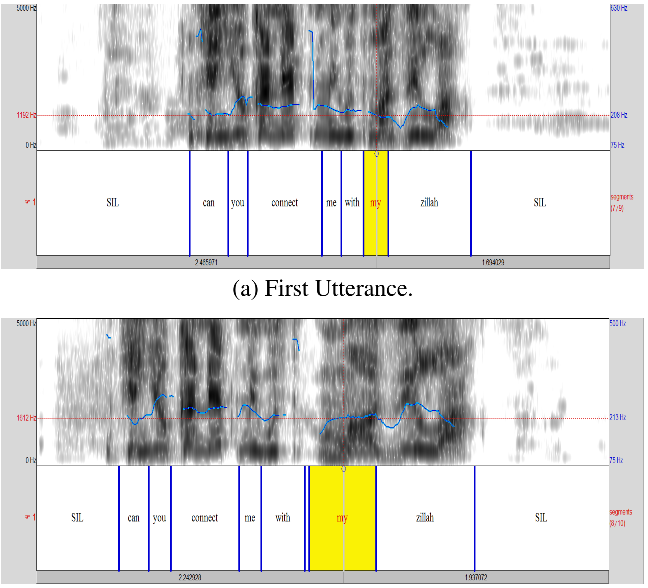
4.1.2. Utterance Pair Features

We use dynamic programming to align the segments of the query pairs based on lexical and time information from the word-segments in the hypotheses for the two utterances. This helps comparing the articulation differences on a word-by-word (*i.e.* segment-by-segment) basis. For each of the aligned segments we compute the deltas of the pre-computed segment-level average features and the deltas between the duration of the aligned segments. Further we group these deltas into positive values and negative values for a given utterance pair. Out of both the groups and the overall set we pick minimum, maximum, average, and ratios for all the deltas. Ratio is the number of positive values or negative values over the total number of aligned segments. Table 2 summarizes these functional features.

Figure 1 shows an utterance pair example for the query *"can you connect me with Mazilla?"* plotted using Praat [12]. The top part shows the spectrogram for the first recognized utterance with the word segmentation and the bottom part shows that of the second utterance. We also show the pitch frequency contours in blue. The speaker is dissatisfied with the recognition and system response of the first attempt. The speaker repeats the same utterance hyperarticulating the contact name *"Mazilla"* which was incorrectly recognized as *"my zillah"*. Hyperarticulation is evident in the duration of the segments of that particular word and also in the pitch frequency which are longer and higher in the second utterance than in the first utterance respectively. In addition, in the second utterance the speaker seems to pause for a short time before hyperarticulating the contact name as shown by the additional silence segment just before the hyperarticulation location which is colored in yellow. The forced-alignment segmentation produces 9 segments in the first utterance and 10 segments in the second one. The word-segments align to each other in a straight forward fashion and functional features are computed over the aligned segments. We discard aligned segments that have *SIL* (silence) in them, whether it is silence-to-silence or silence-to-word alignment.

Table 2. *Functional acoustic features computed for each utterance pair.*

Functional Layer	Description	Dimension
1	Average of word-segments features	17
2	Deltas over aligned segments	17 x #Segments
3A	Min, Max, Ave, Ratio of positive deltas	4 x 17
3B	Min, Max, Ave, Ratio of negative deltas	4 x 17
3C	Min, Max, Ave over all deltas	3 x 17
3	Total	187



(a) First Utterance.

(a) Second Utterance.

Fig. 1. Utterance pair example for the query *"can you connect me with Mazilla?"*.

4.2. Experiments and Results

In this section, we discuss the intrinsic evaluation of our hyperarticulation approach. We explain our experiments setup and then discuss our results.

4.2.1. Experiments Setup

In our experiments, we use ≈3000 annotated utterance pairs for training and 660 utterance pairs for testing. We down-sampled negative examples to have balanced priors in both training and test data, to simplify the experimentation.

In our models, we use gradient boosted decision tree binary classification. Decision trees as a classifier are easier to visualize and integrate into practical solutions. We optimized the parameters separately for each feature group and then all the features together varying the number of trees and iterations. The best performing setup is a decision tree with 100 trees and 100 iterations.

4.2.2. Results

Table 3 shows the results of our experiments on the different feature groups discussed in section 4.1 and their combinations. We measure the performance in terms of accuracy, positive precision, positive recall, negative precision, negative recall and area under the curve. Within feature groups, our results show that the highest accuracy and positive precision comes from the duration features. This could be explained by the fact that users tend to elongate the words as a way of emphasizing them.

We combined all the feature groups and we get the best performance of 67.5% accuracy. Prosody and spectral features by themselves don't show good performance but adding them to duration improves the performance significantly. This shows the importance of spectral and prosody features when there is not enough evidence in the duration features of hyperarticulation. However, duration features alone showed the best performance in terms of negative recall.

When examining the top features in the best performing model, we found that the top features are mainly duration features which are later complemented with prosody and spectral features. We also found that max and min functional features play a more important role than the other functional features. This shows that users usually stress on a part of the utterance and not all. This part of the utterance mainly contains the gist of the request or the hardest word to recognize; for example, the contact name "Mazilla" in Figure 1.

5. SECOND PASS RESCORING

In addition to intrinsic evaluations that focus on hyperarticulation classification quality, in this section, we present an extrinsic evaluation to show usefulness of the hyperarticulation detection task in improving speech recognition overall. The speech recognition system under consideration comprises of a first pass decoder and a second pass of re-ranking of the candidate hypotheses using additional signals as described in [13]. Some details of the rescoring feature space and the ranker are presented below. Further details of speech recognizer such as the first pass acoustic and language model are proprietary, and also not central to our experimental results as the only change we introduce is adding the hyperarticulation related features to the rescoring feature space, while everything else remains fixed.

5.1. Approach

The re-ranking algorithm learns to rescore the N-best list using WER of each hypotheses as the ranking target. The ranker is a LambdaMART [14] model [13], which is based on gradient boosted decision trees and considered among the strongest models for learning supervised rankers. To set the parameters of LambdaMART, we

Table 3. *Hyperarticulation results in terms of Accuracy, Precision, Recall and Area Under Curve (AUC).*

Model	Accuracy	Pos. Precision	Pos. Recall	Neg. Precision	Neg. Recall	AUC
Prosody	55.4	55.5	57.3	55.4	53.5	57.2
Spectral	61.4	61.1	63.7	61.7	59.1	63.8
Duration	65.8	66.6	64.3	65.2	**67.4**	69.0
Duration+Prosody+Spectral	**67.4**	**67.5**	**67.7**	**67.3**	67.1	**69.8**

performed a parameter sweep over the number of trees, number of leaves per tree, learning rate and the minimum number of instances per leaf. Since LambdaMART optimization is beyond the scope of this paper, we only report the results with the best combination of parameters. The feature space of the ranker contains acoustic and language model scores from the first pass decoder as well as language model scores computed over several additional corpora such as large scale web queries, as well as title, body and anchor text of web documents. These language models are bigger than a single machine's memory or SSD can handle, and kept in a distributed key value store (similar to Amazon's DynamoDB [15]), and therefore much more powerful than the language model used in the first pass.

Hyperarticulation information is added to the feature space in a soft decision form using the probability of hyperarticulation. For every utterance in our dataset, we fetch the previous utterance in the same session, if present. If it does not have a previous utterance within 5 mins, we treat the hyperarticulation features as missing. We pass this utterance pair through our hyperarticulation detection setup to get the hyperarticulation classifier probability and the label using the top hypothesis and replicate them for the N-best list. Note that we used the version of the classifier with all the feature groups that gave the maximum gains in accuracy for the hyperarticulation detection task. In addition, we also added the distance metrics that were used as the sampling criteria into the feature space. Table 4 describes all the features that we consider in addition to the standard set of features described in [13]. NbestCandidateEqualsPrevQuery feature captures information from previous query at the N-best candidate level.

Table 4. *Additional features for rescoring.*

Feature Name	Description
Hyperarticulation classifier outputs	
HAClassifierProbability	Decision tree output probability
HAClassifierOutput	Binary label as provided by the classifier
Query Similarity Features	
MetaphoneSimilarity	Edit distance between metaphone representations
PhoneticSimilarity	Edit distance between phoneme lattices
Q2inNbestQ1	True if the recognition result of current query was in the N-best of the previous one
NbestCandidateEqualsPrevQuery	True if the N-best candidate is the same as the previous query

5.2. Experiments and Results

In this section, we describe the second pass rescoring experiments we ran and the results we obtained. The data used for these experiments are human transcriptions for randomly sampled real user data from the same voice enabled personal assistant. The size of the training set for the ranker was 70000 and the test set used was 4000 utterances. The coverage of our additional feature groups mentioned in Table 4 is 54%, which is the number of utterances in this dataset that had a preceding audio query in the same session. We measure the improvements as percentage word error rate reduction (WERR) relative to pre-rescoring WER. The result of adding different feature groups is described in Table 5. Over the standard feature set, we are seeing improvements after adding the hyperarticulation probability and label given by the classifier. Note that we did not apply our data selection criteria described in Section 3.1 in the rescoring experiments. Hence the HAClassifierProbability feature is not very reliable for all the instances. In the following three lines of the table, we see the additional improvements gotten by adding the sampling criteria as features. This indicates that the rescoring classifier learns that HAClassifierProbability is more reliable for the cases that fit our sampling criteria. In the last line of the table, we get the best improvements by adding an N-best candidate level feature NbestCandidateEqualsPrevQuery which in essence captures if the query is very similar to a previous recognition result, and intuitively allows the ranker to down-weight such candidates in the presence of hyperarticulation.

Table 5. *2nd pass rescoring experiment results. WERR is the relative WER reduction compared to the pre-rescoring WER*

Feature Set	WERR(%)
Standard Features	9.82
+ Hyperarticulation classifier output	10.33
+ Query Similarity Features	10.53
+ NbestCandidateEqualsPrevQuery	10.77

When we slice the dataset into only those cases which have a preceding audio query in the same session we get WERR of 11.43% with all the features. The remaining slice which does not have a preceding audio query in the same session has a WERR of 10.14%. This shows that we make significantly higher improvements on the subset of the data which has a preceding audio query as opposed to the subset that does not have a preceding audio query.

6. CONCLUSION

Hyperarticulation detection provides useful signal that could help improve automatic speech recognition experience, specifically through second pass rescoring. We show results for predicting hyperarticulation in repetitive voice queries accurately. We find that aligning and computing segment deltas for prosodic, spectral and duration features help in the hyperarticulation detection task. Using hyperarticulation along with auxiliary features results in further word error rate reduction in a speech recognition rescoring experiment on real user test data. Future extension of applications to this technique are detecting user dissatisfaction, using it as end-to-end metrics and frustration detection. As further improvements to the model, we plan to explore other acoustic properties, such as speech rhythm [16].

7. REFERENCES

[1] Amanda J Stent, Marie K Huffman, and Susan E Brennan, “Adapting speaking after evidence of misrecognition: Local and global hyperarticulation,” *Speech Communication*, vol. 50, no. 3, pp. 163–178, 2008.

[2] Ben Carterette, Pauln. Bennett, David Maxwell Chickering, and Susan T. Dumais, “Here or there: preference judgments for relevance,” in *In Proceedings of ECIR*, 2008, pp. 16–27.

[3] Sharon Oviatt, Margaret Maceachern, and Gina anne Levow, “Predicting hyperarticulate speech during human-computer error resolution,” *Speech Communication*, vol. 24, pp. 87–110, 1998.

[4] Denis Burnham, Sebastian Joeffry, and Lauren Rice, “Computer-and human-directed speech before and after correction,” *space*, vol. 6, pp. 7, 2010.

[5] Linda Bell and Joakim Gustafson, “Repetition and its phonetic realizations: Investigating a swedish database of spontaneous computer-directed speech,” in *In Proceedings of ICPhS-99, San Francisco. International Congress of Phonetic Sciences*, 1999, pp. 1221–1224.

[6] Hagen Soltau, Florian Metze, and Alex Waibel, “Compensating for hyperarticulation by modeling articulatory properties.,” in *INTERSPEECH*, 2002.

[7] A Hassan Awadallah, R Gurunath Kulkarni, U Ozertem, and R Jones, “Characterizing and predicting voice query reformulation,” in *Proc. CIKM*, pp. 543–552.

[8] Lawrence Philips, “The double metaphone search algorithm,” *C/C++ users journal*, vol. 18, no. 6, pp. 38–43, 2000.

[9] Jeremy Ang, Rajdip Dhillon, Ashley Krupski, Elizabeth Shriberg, and Andreas Stolcke, “Prosody-based automatic detection of annoyance and frustration in human-computer dialog.,” in *INTERSPEECH*. Citeseer, 2002.

[10] Joseph L Fleiss, “Measuring nominal scale agreement among many raters.,” *Psychological bulletin*, vol. 76, no. 5, pp. 378, 1971.

[11] David Talkin, “A robust algorithm for pitch tracking (rapt),” *Speech coding and synthesis*, vol. 495, pp. 518, 1995.

[12] Paul Boersma et al., “Praat, a system for doing phonetics by computer,” *Glot international*, vol. 5, no. 9/10, pp. 341–345, 2002.

[13] Milad Shokouhi, Umut Ozertem, and Nick Craswell, “Did you say u2 or youtube?: Inferring implicit transcripts from voice search logs,” in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 1215–1224.

[14] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao, “Adapting boosting for information retrieval measures,” *Information Retrieval*, vol. 13, no. 3, pp. 254–270, 2010.

[15] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels, “Dynamo: amazon’s highly available key-value store,” *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 205–220, 2007.

[16] Robert Fuchs, *Speech Rhythm in Varieties of English: Evidence from Educated Indian English and British English*, Springer, 2015.