INTERSPEECH 2009 BRIGHTON

New Methods for the Analysis of Repeated Utterances

Geoffrey Zweig

Microsoft Research, One Microsoft Way, Redmond, WA 98052

gzweig@microsoft.com

Abstract

This paper proposes three novel and effective procedures for jointly analyzing repeated utterances. First, we propose repetition-driven system switching, where repetition triggers the use of an independent backup system for decoding. Second, we propose a cache language model for use with the second utterance. Finally, we propose a method with which the acoustics from multiple utterances - not necessarily exact repetitions of each other - can be combined to into a composite that increases accuracy. The combination of all methods produces a relative increase in sentence accuracy of 65.7% for repeated voice-search queries.

Index Terms: speech recognition, joint decoding, repeated utterances

10.21437/Interspeech.2009-713

1. Introduction

Due to speech recognition errors, user repetitions are very common in deployed applications, and their joint analysis is an important technical challenge. We focus here on voice-search applications where the repeated phrases are short and typically involve simple modifications, for example "Blueberry Cafe" followed by "Blueberry Cafe on Main Street."

A number of authors have addressed the problem of analyzing repeated utterances. Recently, [1] has used joint acoustic modeling to improve the performance of single-word recognition. In [2], repetition is used as a cue to identify advertisements in television broadcasts. The paper [3] uses confusion networks and a process of forced correction to combine multiple repetitions of military utterances. The problem of finding repeated sequences themselves is addressed in [4], and a qualitative analysis of repetitions (duration, intensity, hyper-articulation, etc.) is found in [5, 6]. The idea of using information generally across multiple turns in the conversation appears in earlier works [7, 8], and [9] proposes methods of tracking and exploiting n-best lists across multiple turns in a POMDP system.

The problem of repetition analysis in the context of a toll-free directory assistance application was studied in [10]. This work did joint language modeling across utterances, utilizing the fact that repeated utterances tend to be related by structural transformations such as truncation, extension, and exact duplication. Further, it modeled the fact that there is an enumerable set of businesses with toll free numbers, and these were summed over in finding the likeliest word sequences.

This paper extends previous work in several ways. First, we propose repetition-driven system switching, and demonstrate that the use of an independently trained "backup" system is exceptionally effective in analyzing repeated utterances. We find that while one recognizer (i.e. the deployed one) may have trouble with a particular speaker/query combination, another independently trained recognizer has a good chance of working well on those utterances.

Secondly, we develop a cache language model, and show that its performance in this application is comparable to a more complicated model that explicitly models structural transformations [10].

Finally, we propose a novel method for combining the acoustic feature vectors of repeated utterances prior to decoding. Our method works by finding closely matching blocks of frames, aligning the frames within these blocks, and then averaging them to reduce noise. This is complicated by the fact that in contrast to [1], the utterances may not be exact repetitions, and therefore a straightforward dynamic time alignment may not be appropriate. To overcome this, we develop a principled method of matching any number of blocks of frames from one utterance to a corresponding number of blocks in another. We are able to make this decision by adopting a generative model in which each frame in the second utterance is either explained as a noisy copy of a frame in the first, or as having being drawn from a background model. A maximum likelihood partition of the second utterance into matched and unmatched portions can then be made. The matching procedure induces an alignment between frames in the first utterance, and frames in the second utterance. Those that are aligned to each other are averaged, and the result is used in decoding.

2. Joint Modeling Formulation

The problem we consider is that of finding the likeliest pair of word sequences, $\mathbf{w_1}, \mathbf{w_2}$ given a pair of acoustic realizations, $\mathbf{a_1}, \mathbf{a_2}$, i.e. to determine

$$\underset{\mathbf{w_1},\mathbf{w_2}}{\operatorname{argmax}} P(\mathbf{w_1}, \mathbf{w_2}|\mathbf{a_1}, \mathbf{a_2}). \quad (1)$$

This can be factored in a number of ways, and different conditional independence assumptions applied. In this paper, we explore the following models. The baseline is to compute

$$\underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w})P(\mathbf{a}|\mathbf{w}) \quad (2)$$

independently for both utterances. The next approach models the relationship between the first and second word sequences:

$$\underset{\mathbf{w_1},\mathbf{w_2}}{\operatorname{argmax}} P(\mathbf{w_1})P(\mathbf{w_2}|\mathbf{w_1})P(\mathbf{a_1}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_2}) \quad (3)$$

Specifically, we use structural-transformation language models [10] and cache language models [11] to determine $P(\mathbf{w_2}|\mathbf{w_1})$. Finally, we explore a method of joint acoustic modeling. In [10], the factorization

$$\underset{\mathbf{w_1},\mathbf{w_2}}{\operatorname{argmax}} P(\mathbf{w_1})P(\mathbf{w_2}|\mathbf{w_1})P(\mathbf{a_1}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_1}, \mathbf{w_2}, \mathbf{a_1}) \quad (4)$$

was proposed with the idea of using an adaptive model for $\mathbf{a_2}$. We have found, however, that the error rates on the first round

utterances are much higher than on the second round ones, and using the first-round acoustics in the processing of the second utterances typically results in more errors.

Therefore we explore an alternative, probabilistically valid but non-causal, model that modifies the *first* round acoustics based on the second turn. Mathematically, we consider finding:

$$\underset{\mathbf{w_1},\mathbf{w_2}}{\operatorname{argmax}} P(\mathbf{w_1})P(\mathbf{w_2}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_2})P(\mathbf{a_1}|\mathbf{w_1},\mathbf{w_2},\mathbf{a_2}) \quad (5)$$

Further, in the computation of $P(\mathbf{a_1}|\mathbf{w_1},\mathbf{w_2},\mathbf{a_2})$, we do not use $\mathbf{w_2}$ so our final model is:

$$\underset{\mathbf{w_1},\mathbf{w_2}}{\operatorname{argmax}} P(\mathbf{w_1})P(\mathbf{w_2}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_2})P(\mathbf{a_1}|\mathbf{w_1},\mathbf{a_2}) \quad (6)$$

In all cases, the optimization is done by examining pairs of word sequences on the n-best lists of the first and second utterances, where we get the n-best lists by decoding with a "backup" model.

3. Joint Language Model

In [10], a language model that explicitly models structured transformations was used in the computation of $P(\mathbf{w_2}|\mathbf{w_1})$. In this work, we implement a cache language model [11] and find that it provides somewhat better performance. To create a cache language model , we create a tiny language model based on the words in $\mathbf{w_1}$. Since there are usually only two words in $\mathbf{w_1}$ (business names are short), this model typically just has two non-zero probabilities, each 0.5. The probability generated by this model for word i in $\mathbf{w_2}$ is denoted $P_c(w_2^i|\mathbf{w_1})$. This is interpolated with the normal trigram language model probability $P(w_2^i|w_2^{i-1,i-2})$. The resulting language model probability is:

$$P(\mathbf{w_2}|\mathbf{w_1}) = \prod_i \alpha P_c(w_2^i|\mathbf{w_1}) + (1-\alpha)P(w_2^i|w_2^{i-1,i-2})$$

4. Joint Acoustic Model

The proposed acoustic modeling approach is based on the idea of averaging portions of the utterances to obtain more reliable acoustic features; a similar idea was proposed in [1], without the ability to do partial utterance matches. The goal of this approach is to find one or more closely matching blocks of frames, to align the frames within matched blocks, and then to average them. Our approach for finding repeated segments differs from [4] in that it does not require a pre-segmentation into word-like units, and uses the notion of competing generative models to make the segmentation on the basis of maximum likelihood rather than thresholds. To decide which portions to match and which not to match, the principle we adopt is to find a maximum likelihood explanation for the frames of one of the utterances, in terms of having been generated by the other. The matched frames in one utterance (X) are used as a template to explain the generation of the matched frames in the other utterance (Y); all other frames in utterance Y are accounted for with a background model. The principle of maximum likelihood then induces and optimal segmentation into matched and unmatched blocks. The approach is illustrated in Figure 1 and explained below.

In Figure 1, the frames of one utterance are represented by the circles in the top row; the frames of the other utterance are represented by the circles in the bottom row. Two matching blocks have been identified by the shaded areas. Within a block, the frames from the top utterance are taken as a template from which the bottom utterance frames are generated. There

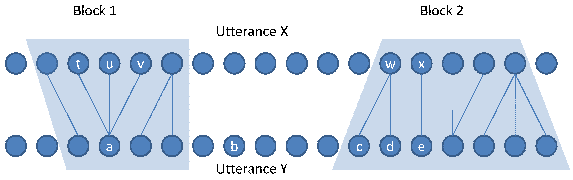

Figure 1: Matching the frames of one utterance to another.

are three cases to consider. In a direct match, a single frame is taken as a Gaussian center from which a single sample is drawn. The frames labeled x and e are an example of this. In a one-to-many match, a single frame is used as a Gaussian mean from which many samples are drawn. The frames w, c and d are examples of this. Finally, in a many-to-one match, several frames in the top utterance are averaged and the result is used as the Gaussian mean from which a single sample is drawn. The frames t, u, v and a illustrate this. In all cases, the variance used is the global variance of the frames in the top utterance. In our model, no other matching patterns are allowed (e.g. adding an edge between w and e or x and d). An unmatched frame such as b is explained as being drawn from a global background model consisting of a single Gaussian estimated from all the frames in the top utterance. Under this model, creating a matched block will be advantageous when multiple frames in sequence from utterance X are relatively close matches to a block of frames in utterance Y.

It is possible to use dynamic programming to find the optimal segmentation of the utterances into k matching blocks. We use the following definitions:

- α_{ijkm} - the best way of consuming all the frames up to and including position i in utterance X, and all the frames up to and including position j in utterance Y, and having seen k block starts, and either matching ($m = 1$) or not matching ($m = 0$) frames i and j.
- $match(a, b)$ - the negative log likelihood of drawing feature vector a from a Gaussian centered on b
- $no\_match(a)$ - the negative log likelihood of drawing feature vector a from the global background model
- $ave(r, p)$ - the average of the frames in positions r through p inclusive in utterance X.
- $max\_compression$ - the maximum number of frames that can be multiply matched, e.g. t, u, v matching a in Figure 1 is a compression of 3.
- l_1, l_2 - the lengths of utterance X and Y.
- N - the number of blocks desired

The recursion then proceeds as follows. Note that the roles of the first and second utterance are asymmetrical, and this is reflected in the recursions.

Initialization:
$\alpha_{0,0,0,0} = 0$
$\alpha_{i,0,0,0} = 0,\ i = 1 \ldots l_1$
$\alpha_{0,j,0,0} = \alpha_{0,j-1,0,0} + no\_match(j),\ j = 1 \ldots l_2$
$\alpha_{i,j,k,l} = huge\_cost,\ \forall i, j, k, l$
$\alpha_{i,j,0,0} = alpha_{i-1,j-1,0,0} + no\_match(j),$
$\quad i = 1 \ldots l_1\ j = 1 \ldots l_2$

Recursion:
for $i = 1 \ldots l_1$
 for $j = 1 \ldots l_2$
  for $s = 1 \ldots N$
   consider all the ways of ending in a match state
   by consuming multiple frames from utterance X
   $\alpha_{i,j,s,1} = min($
    $\min_{l \in 1 \ldots max\_compression-1}$
     $\alpha_{i-l-1,j-1,s,1} + match(ave(i-l,i),j)$,
    $\min_{l \in 1 \ldots max\_compression-1}$
     $\alpha_{i-l-1,j-1,s-1,0} + match(ave(i-l,i),j))$

   consider all the ways of ending in a match state
   by consuming one frame from utterance X
   and one or more frames from utterance Y
   $\alpha_{i,j,s,1} = min(\alpha_{i,j,s,1},$
    $min($
     $min_{l \in 0 \ldots max\_compression-1}$
      $\alpha_{i-1,j-l-1,s,1} + \sum_{f=j-l,j} match(f,i)$,
     $min_{l \in 0 \ldots max\_compression-1}$
      $\alpha_{i-1,j-l-l,s-1,0} + \sum_{f=j-1,j} match(f,i)))$

   consider all the ways of ending in a no-match state
   $\alpha_{i,j,s,0} = min($
    $\alpha_{i-1,j,s,0}$,
    $\alpha_{i,j-1,s,0} + no\_match(j)$,
    $\alpha_{i-1,j,s,1}$,
    $\alpha_{i,j-1,s,1} + no\_match(j))$

As with other dynamic programming processes, "backpointers" can be maintained when computing the optimal cost thus allowing recovery of the segmentation and match sequences. With careful implementation and precomputation of frame averages, a runtime of $O(l1 * l2 * N * max\_compression)$ can be achieved. In practice the maximum compression is 2 or 3 and N on the order of 1 to 3, so the runtime is quadratic in utterance length.

After the optimal alignment is found, the frames that are matched together are averaged and a new feature file created for subsequent decoding.

5. Data

Our data is drawn from the "Live Search Mobile" application [12]. This is a cell-phone application whose voice-enabled functionality includes specifying business names, search locations (cities, states and zip codes) and addresses for driving directions. In this application, the user is given an n-best list of search results, and can either select and accept one of the results, or speak again. The fact that the user is shown an n-best list is significant because in the case of repetition, the correct answer is unlikely to be on the n-best, and thus n-best rescoring methods are unlikely to succeed. This motivates our use of repetition-driven system switching.

5.1. Test Sets

To test our approaches to analyzing repeated utterances, we have created development and test sets consisting of repetitions gathered from a random selection of users in May 2008 and professionally transcribed. To create a test set that focuses on repetition, we used the following heuristic: two consecutive utterances from a single user that were received within 60 seconds of each other and which share at least one word in the transcription are deemed repetitions. Manual checking of the transcriptions indicates that there are very few errors in this heuristic. To further focus our work, we restrict attention to business names as opposed to addresses or cities; this information is marked in the transcriptions and known with certainty. In the resulting development set, there are 2483 utterances, and the test set has 3693 utterances. This accounts for 29% of all business name utterances.

5.2. Data Characteristics

The data we are working with consists of chains of repetitions, with the user sometimes attempting numerous times to communicate with the system. While space constraints prevent a complete accounting, about 80% of the data is either a first or second turn; about 12% is a third turn, and 3.8% of the data is accounted for by fifth and later turns. The average utterance length is 2.3 words, and increases slightly with turn number.

As noticed in [10], repetitions are frequently related to each other by simple structural transformations. Compared with the previous work on a toll-free directory assistance application, the main differences are that exact matches are more common in Live Search for Mobile (55.4% vs. 46%) and right truncations are less common (6.2% vs. 13.7%). Overall, in the training data, structured transformations account for 82.0% of the repetitions observed. For the cases that fall outside the transformations of [10] (e.g. "El Toreador Restaurant" followed by "El Toreador *Mexican* Restaurant") it is still common for many of the individual words to be repeated. Numerically, we find that 58% of the words in non-structured repetitions occur in the preceding utterance. Combined with the fact that 84% of *all* second-round words have been seen in the preceding utterance, we are motivated to apply the cache language model described earlier.

6. Experiments

6.1. Deployed System

The deployed Live Search for Mobile system uses a standard speech recognition setup with crossword triphone acoustic context and a trigram language model. The acoustic model was trained with approximately 4000 hours of general telephony data including about 500 hours of semi-supervised "click" data from the system itself. This data consists of the association between audio and words that is collected when a user accepts an item on the n-best list. The system has about 6000 context dependent acoustic states and 150k maximum-likelihood trained Gaussians. Feature vectors have HLDA applied, and no speaker normalization is used. The language model is trained on a collection of business listings, click data, and directory assistance data, and has approximately four million n-grams.

6.2. Repetition Driven System Switching

As mentioned in Section 1, the occurrence of repetition is prima facie evidence that the deployed system has failed. We find that about 82% of the time, when the user repeats himself, the correct answer was not on the preceding n-best list. (Almost 18% of the time it is, perhaps indicating user confusion or distraction.) To generate alternate hypotheses to work with, we trained a backup system with somewhat different data, and differing in the use of MFCCs, deltas and double-deltas rather than HLDA, and in the use of offline rather than online cepstral mean nor-

System	General Data	Repeat Test Set
Deployed	59.0%	31.2 %
Backup	64.1	49.6
Absolute Difference	5.1	18.4
Relative Improvement	8.6	59

Table 1: Sentence accuracy of deployed and backup system on general and repeated data.

System	Accuracy
Deployed	31.2 %
System Switching	49.6
Switching + Feature Combination	50.1
Switching + Structured LM	50.4
Switching + Cache LM	51.5
Switching + Combination + Cache	51.7

Table 2: Sentence accuracy of various systems on repeat test set.

malization (latency is not such a serious issue in the context of repetition and one or two second utterances). The training data for this system consisted of 2500 hours of Live Search for Mobile click data, and 26 hours of transcribed data. This system has 10,900 context dependent acoustic states and 260k Gaussians. The language model is the same as that of the deployed system.

Table 1 summarizes the system performance on a general development set consisting of all business utterances, and on the repeat test set. Whereas the backup system is only about 10% relative better than the initial system overall, it is much more effective on the repeated utterances - 59% relative.

6.3. Joint Analysis

Here we present results for the various proposed improvements. In each case, any relevant parameters were tuned on the development set. Results are presented on the test set. In the case of long chains of repetitions, e.g. (A B C), we consider sequential pairs, e.g. (A B) (B C), and in the case of an utterance like B, the answer from the first pair is retained.

Table 2 shows the accuracy of the deployed system and our improvements on the repeat test set. System switching provides a very large boost in accuracy on the repeated data. There is also a 0.5% absolute improvement from combining acoustic realizations, and a significant improvement from the use of a cache LM, which outperforms one modeling structured transformations. Using the proposed acoustic combination method in association with the cache LM produces a further improvement.

Figure 2 breaks down the accuracy figures by utterance turn; "1" is the first utterance in a string of repetitions, and "2" represents utterances from the second turn on. This indicates that improvements are seen throughout the interactions - both first and second turn utterances.

7. Conclusion and Future Work

This paper has presented three effective approaches to dealing with repeated utterances in voice-search: the use of system switching to decode the repeated utterances; the use of a cache language model in a joint analysis of pairs of n-best lists; and a novel segmentation and feature averaging method. On a test

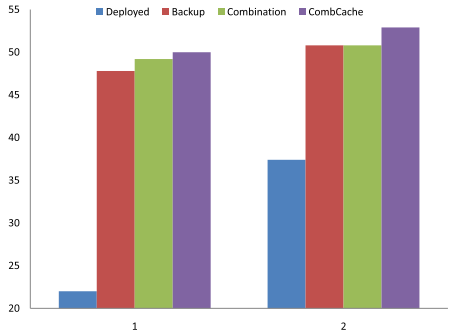

Figure 2: System accuracies by turn.

set of repetitions, the use of system switching results in a 59% relative improvement (18.4% absolute) over the deployed system, despite the backup system being only 10% relative better on the data overall. Averaging the frames of matched blocks in the repeated utterances results in a further 0.5% absolute improvement. Using a cache language model in a joint analysis of the n-best lists gotten by decoding the modified utterances gives a final 1.6% absolute improvement. Altogether the sentence accuracy is increased by 65.7% relative, from 31.2% to 51.7%. We believe the utterance matching algorithm we have presented has further applicability in automatically finding repeated sub-portions, and in detecting repetition when it is not explicitly marked.

8. References

[1] N. U. Nair and T. V. Sreenivas, "Joint decoding of multiple speech patterns for robust speech recognition," in *Proc. of ASRU*, 2007.

[2] V. Gupta, G. Boulianne, P. Kenny, and P. Dumouchel, "Advertisement detection in french broadcast news using acoustic repetition and gaussian mixture models," in *Proc. of Interspeech*, 2008.

[3] F. Cesari, H. Franco, G. K. Myers, and H. Bratt, "MUESLI: Multiple Utterance Error Correction for a Spoken Language Interface," in *Proc. of Interspeech*, 2008.

[4] M. Cevik, F. Weng, and C. Lee, "Detection of repetitions in spontaneous speech dialogue sessions," in *Proc. of Interspeech*, 2008.

[5] S. Oviatt, G.-A. Levow, M. MacEachern, and K. Kuhn, "Modeling hyperarticulate speech during human-computer error resolution," in *Proc. of ICSLP*, 1996.

[6] L. Bell and J. Gustafson, "Repetition and its phonetic realizations: Investigating a swedish database of spontaneous computer-directed speech," in *Proc. of ICPhS*, 1999.

[7] T. Paek and E. Horvitz, "Deeplistener: Harnessing expected utility to guide clarification dialog in spoken language systems," in *ICSLP*, 2000.

[8] D. Bohus and A. Rudnicky, "A k hypotheses + other belief updating model," in *Proc. of AAAI Workshop on Statistical and Empirical Approaches to Spoken Dialogue Systems*, 2006.

[9] J. D. Williams, "Exploiting the asr n-best by tracking multiple dialog state hypotheses," in *Proc. of Interspeech*, 2008.

[10] G. Zweig, D. Bohus, X. Li, and P. Nguyen, "Structured models for joint decoding of repeated utterances," in *Proc. Interspeech*, 2008.

[11] R. Kuhn and R. de Mori, "A cache-based natural language model for speech recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, 2008.

[12] Alex Acero et al., "Live search for mobile: Web services by voice on the cellphone," in *Proc. of ICASSP*, 2007.