Structured Models for Joint Decoding of Repeated Utterances

Geoffrey Zweig, Dan Bohus, Xiao Li, Patrick Nguyen

Microsoft Research, One Microsoft Way, Redmond, WA 98052
{gzweig,dbohus,xiaol,panguyen}@microsoft.com

Abstract

Due to speech recognition errors, repetition can be a frequent occurrence in voice-search applications. While a proper treatment of this phenomenon requires the joint modeling of two or more utterances simultaneously, currently deployed systems typically treat the utterances independently. In this paper, we analyze the structure of repetitions and find that in at least one commercial directory assistance application, repetitions follow simple structural transformations more than 70% of the time. We present preliminary results that suggest that significant gains are possible by explicitly modeling this structure in a joint decoding process.

Index Terms: speech recognition, minimum bayes risk, joint decoding, repeated utterances

1. Introduction

Due to the imperfect nature of speech recognition technology, repetition is an intrinsic part of many of today's interactions with automated systems. For example, an analysis of call logs from a commercial directory assistance application indicates that repetition occurs in about 49% of interactions - about half the utterances are either the first or second turn in a repetition. When repetition occurs, the same information is presented redundantly, and one might intuitively expect that there is a way of exploiting this redundancy to improve recognizer performance. This is typically not done, however, with systems instead decoding indendently and then post-facto suppressing n-best results that were already explicitly disconfirmed in an earlier turn.

Recently, [1] has used joint acoustic modeling to improve the performance of single-word recognition. In this approach, multiple occurrences of an individual word are first aligned amongst each other, and then the consensus alignment is aligned to an HMM in a constrained Viterbi process. In this paper, we study the related but significantly different problem of decoding repetions that *might not be identical*, but which *derive from reference to a finite set of entities*, such as is found in directory assistance applications, or in voice-search more generally. Consider for instance a yellow-pages directory assistance application. A user calling to find out the phone number for the customer service line of General Motors might first say "General Motors customer service". If the recognition result for this first utterance does not have a high confidence score, the application will query the user again. The second time around, the user might respond "I want the one-eight-hundred number for General Motors company", or perhaps simply "General Motors". (Table 1 provides several other concrete examples from such a deployed application.) The repetition need not be exact, but the two user utterances are tied by the same underlying entity - in this case one of a large but enumerable set of businesses.

The idea of using information across multiple turns in the conversation appears in earlier works, such as [2], which uses a dynamic bayesian network to update belief states across multiple utterances over the course of a dialog in a command-and-control application. Similarly, [3] presents a method for learning belief updating models that scale up in a more complex spoken dialog system. In other related work, [4, 5] study repetition from a descriptive point-of-view (duration, intensity, hyperarticulation, etc.) but do not address automatic speech recognition, and [6] proposes the use of dialog state to improve ASR performance, but does not address repetition. The work we discuss here is novel in that we investigate and leverage the particular structure of repeated utterances, and in that we focus on the recognition process and introduce a joint decoding model.

2. Framework

If we denote the value of an underlying reference by l (for listing) and denote word and acoustic sequences with **w** and **a** respectively, then in the approach we adopt, we are interesed in finding the likeliest sequences:

$$\begin{aligned}
&\operatorname*{argmax}_{\mathbf{w_1}..\mathbf{w_n}} P(\mathbf{w_1}..\mathbf{w_n}|\mathbf{a_1}..\mathbf{a_n}) \\
&\quad= \operatorname*{argmax}_{\mathbf{w_1}..\mathbf{w_n}} \sum_l P(\mathbf{w_1}..\mathbf{w_n}, l|\mathbf{a_1}..\mathbf{a_n}) \\
&\quad= \operatorname*{argmax}_{\mathbf{w_1}..\mathbf{w_n}} \sum_l P(\mathbf{w_1}..\mathbf{w_n}, l)P(\mathbf{a_1}..\mathbf{a_n}|\mathbf{w_1}..\mathbf{w_n}, l) \\
&\quad= \operatorname*{argmax}_{\mathbf{w_1}..\mathbf{w_n}} \sum_l P(l)P(\mathbf{w_1}..\mathbf{w_n}|l)P(\mathbf{a_1}..\mathbf{a_n}|\mathbf{w_1}..\mathbf{w_n}, l)
\end{aligned}$$

Since the vast majority of repetitions in our corpus have exactly two utterances, we focus on the two-turn case for the remainder of the paper.

$$\begin{aligned}
&\operatorname*{argmax}_{\mathbf{w_1},\mathbf{w_2}} \sum_l P(l)P(\mathbf{w_1},\mathbf{w_2}|l)P(\mathbf{a_1},\mathbf{a_2}|\mathbf{w_1},\mathbf{w_2}, l) \\
&\quad\approx \operatorname*{argmax}_{\mathbf{w_1},\mathbf{w_2}} \sum_l P(l)P(\mathbf{w_1}|l)P(\mathbf{w_2}|\mathbf{w_1}, l) \\
&\qquad\qquad P(\mathbf{a_1}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_1},\mathbf{w_2},\mathbf{a_1}, l) \\
&\quad\approx \operatorname*{argmax}_{\mathbf{w_1},\mathbf{w_2}} \sum_l P(l)P(\mathbf{w_1}|l)P(\mathbf{w_2}|\mathbf{w_1}, l) \\
&\qquad\qquad P(\mathbf{a_1}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_2})
\end{aligned}$$

In the first approximation, we have assumed that $P(\mathbf{a_1}|\mathbf{w_1},\mathbf{w_2}, l) = P(\mathbf{a_1}|\mathbf{w_1})$, and in the second approximation we assume further that $P(\mathbf{a_2}|\mathbf{w_1},\mathbf{w_2},\mathbf{a_1}, l) = P(\mathbf{a_2}|\mathbf{w_2})$. We note that this second approximation may be inadequate - in the case of exact repetition, the first sample of words and acoustics should significantly sharpen the probability distribution over acoustics for the second utterance. However, we leave it for later work to address appropriate forms of acoustic adaptation. In contrast to [1], we focus on the language modeling aspects of repeated utterances.

The proposed model therefore consists of several components:

1. The first component, $P(l)$, captures the prior distribution for the set of listings.
2. The second component, $P(\mathbf{w_1}|l)$, can be thought of as a translation model that maps from the written form of a listing l to a corresponding spoken form $\mathbf{w}$ [7].
3. The third component, $P(\mathbf{w_2}|\mathbf{w_1}, l)$, captures how users repeat themselves, at the language level – this can be thought of as a repetition language model.
4. Finally, the last two components in the proposed factorization, $P(\mathbf{a_1}|\mathbf{w_1})$ and $P(\mathbf{a_2}|\mathbf{w_2})$ represent the acoustic scores for the corresponding utterances.

A key characteristic of the proposed joint decoding model is that the multiple utterances are "tied together" by the assumption of a single underlying concept. In the directory assistance application we study, the underlying set of concepts is a set of approximately 149,000 names for businesses with toll free numbers. While this is too large a number to permit exhaustive calculation of the sum over listings, we will later present a simple process for efficiently approximating the sum.

The observant reader will notice that in the case of a voice-search application with a finite set of listings, one could just as easily express the problem as one of finding the likeliest listing - one might not care about the words themselves. In this case, the problem we are solving is:

$$\begin{aligned}\operatorname*{argmax}_l P(l|\mathbf{a_1}, \mathbf{a_2}) &= \operatorname*{argmax}_l \sum_{\mathbf{w_1},\mathbf{w_2}} P(l, \mathbf{w_1}, \mathbf{w_2}|\mathbf{a_1}, \mathbf{a_2}) \\ &\approx \operatorname*{argmax}_l \sum_{\mathbf{w_1},\mathbf{w_2}} P(l)P(\mathbf{w_1}|l)P(\mathbf{w_2}|l, \mathbf{w_1}) \\ &\qquad P(\mathbf{a_1}|\mathbf{w_1})P(\mathbf{a_2}|\mathbf{w_2})\end{aligned}$$

However, as we will see in Section 3, while our dataset has ortographic transcriptions, and is therefore quite precise about the words that are present, we do not know the ground truth for the listing desired by the user. Therefore, in this paper we focus on the first version on the task - more accurately recovering the spoken words themselves. We also deduce both $\mathbf{w_1}$ and $\mathbf{w_2}$ rather than $\mathbf{w_2}$ alone since in the application we study, $\mathbf{w_1}$ may never be presented to the user for confirmation (when the system has low confidence), and is therefore truly unknown. In the case that one is interested in $\mathbf{w_2}$ only, our procedures may still be used by summing rather than maximizing over $\mathbf{w_1}$. Notice that solving the problem at the lexical (rather than semantic) level provides added benefits in an application that needs to perform explicit or implicit confirmation actions.

3. Data

While the problem of dealing with repetitions is quite general, this paper focuses on a specific instance of it which motivates certain design and algorithmic decisions. The system we are concerned with is a commercially deployed high volume toll-free directory assistance application. Approximately $43,000$ businesses are in this system, sometimes with multiple synonyms for each business (e.g. "greater alarm" and "greater alarm company"), leading to a total of approximately $149,000$ names. We stress that while our experiments focus on a directory assistance application, the issues and theory are common across other voice-search applications [9, 8, 10, 11, 12].

```
auto_id=38392
      lil tykes company
      little tikes customer service
      little tikes toys
      the little tikes toys

auto_id=39036
      greater alarm company
      greater alarm

auto_id=7133
      d p i and associates
```

Figure 1: Sample database listings.

The corpus used in the experiments described in this paper consists of a set of $150,000$ ortographically transcribed user utterances from the above-mentioned directory assistance system. Each session with the system has a unique identifier. This allowed us to detect pairs of repeated utterances, consisting of an initial request, followed by a repetition of that request. Such paired utterances account for about half of the total utterances received by the system. For development and testing purposes, we separated (by random sampling of pairs) a development set of $11,838$ utterances, and a test set of $12,650$ utterances.

Due to third party licensing restrictions, the output of the original recognizer was not available. To obtain decoding output including n-best lists, we redecoded with a different commercial recognizer. The 1-best accuracy of this recognizer was 44.0%, but acoustic and language model scores were not available. We recreated the language model scores using the CMU LM toolkit [13], and used the logarithm of an entry's rank on the n-best list as a surrogate for the acoustic score. This resulted in a baseline of 44.2%, slightly better than the original recognizer.

In the deployed system, repetition is prompted under one of two circumstances: first, if the recognizer has low confidence, the system will immediately request a clarifying statement. Second, if the user explicitly disconfirms a hypothesis, a repetition will result. In the case that an initial hypothesis is rejected, one would of course want to assign it zero probability in any sums in which it occurs. In contrast, in the case of low confidence, nothing can be discarded. Since we redecoded the data, we are left with a stylized data set in which we have two utterances related by repetition, but no particular information about the circumstances of the repetition at runtime.

4. Models

4.1. Baseline

The baseline model computes the likeliest word sequences for each utterance independently:

$$\begin{aligned}\mathbf{w_1}^* &= \operatorname*{argmax}_{\mathbf{w_1}} P(\mathbf{w_1})P(\mathbf{a_1}|\mathbf{w_1}) \\ \mathbf{w_2}^* &= \operatorname*{argmax}_{\mathbf{w_2}} P(\mathbf{w_2})P(\mathbf{a_2}|\mathbf{w_2})\end{aligned}$$

In this model, $\mathbf{w_1}$ and $\mathbf{w_2}$ are restricted to those appearing on the n-best lists.

Type	Frequency	$\mathbf{w_1}$	$\mathbf{w_2}$
Exact Match	46.0%	Starbucks	Starbucks
Right Extension	6.6%	Starbucks	Starbucks Coffee
Right Truncation	13.7%	Blockbuster Video	Blockbuster
Left Extension	1.6%	Roma's Pizza	Tony Roma's Pizza
Left Truncation	2.8%	The Red Lion Inn	Red Lion Inn
Inclusion	1.2%	The Social Security Administration	Social Security
Cover	0.4%	Kodak	Eastman Kodak Corporation

Table 1: Frequencies and examples of structured repetition. "Type" shows how the second utterance is related to the first.

4.2. Pure Counting

Our counting model simply counts the number of times a word sequence has been used to request a listing and uses relative frequencies:

$$P_c(\mathbf{w_1}|l) = \frac{\#(\mathbf{w_1}, l)}{\#(l)}$$

Due to data sparsity issues, we have found it beneficial to approximate $P_c(\mathbf{w_2}|\mathbf{w_1}, l)$ as $P_c(\mathbf{w_2}|l)$ alone, and to further tie the statistics across $\mathbf{w_1}$ and $\mathbf{w_2}$. Thus for both $\mathbf{w_1}$ and $\mathbf{w_2}$ we have

$$P_c(\mathbf{w}|l) = \frac{\#(\mathbf{w}, l)}{\#(l)}$$

Even so, a pure counting model is too sparse to be of use, and must be further smoothed (see next subsection).

To estimate the counts model, we matched the reference transcriptions against the set of existing listings (including the available synonyms). An exact match (modulo acoustic non-lexical events like /um/, /oh/) was found in 61% of the cases, resulting in 92,000 transcription/listing pairs.

4.3. Interpolated Counting

To further smooth the counting models, we have found it beneficial to interpolate the count-based estimate with a standard language model estimate resulting in what will be referred to this as our "unstructured" model P_{us}.

$$P_{us}(\mathbf{w}|l) = \alpha P_c(\mathbf{w}|l) + (1 - \alpha)P(\mathbf{w})$$

P($\mathbf{w}$) is estimated with a standard n-gram language model. This estimate is used for both $\mathbf{w_1}$ and $\mathbf{w_2}$, with interpolation weights set to 0.9 through optimization on the development set.

4.4. Structured Repetition Models

The most significant gains from the proposed approach have come from explicitly modeling the structure that is present in repetitions, via $P(\mathbf{w_2}|\mathbf{w_1}, l)$. A corpus analysis has revealed several types of simple transformations that together account for a large proportion of the repeated utterances. Table 1 illustrates these transformations, along with examples and frequencies. Is is interesting to notice that these simple transformations account for approximately 72% of the data. Since exact matches and right-truncations cover approximately 60% of all repetitions, and the next most frequent phenomenon (right-extension) accounts for only 6.6% of the data, we have focused our models and experiments on the two most common cases.

All our structured models are interpolated language models drawing from a set of atomic models indexed by z. We have:

$$\begin{aligned} P(\mathbf{w_2}|\mathbf{w_1}, l) &= \sum_z P(\mathbf{w_2}, z|\mathbf{w_1}, l) \\ &= \sum_z P(z|\mathbf{w_1}, l)P(\mathbf{w_2}|\mathbf{w_1}, l, z) \end{aligned}$$

A standard interpolated language model on $\mathbf{w_2}$ results from the assumptions $P(z|\mathbf{w_1}, l) = P(z)$ and $P(\mathbf{w_2}|\mathbf{w_1}, l, z) = P(\mathbf{w_2}|z)$. To simplify notation in subsequent discussion, we will use $P_z(\cdot)$ to denote $P(\cdot|z)$. Further, note that these models apply to the second-turn utterance only; $P(\mathbf{w_1}|l)$ is modeled with the unstructured model, P_{us}.

4.4.1. *Exact Repetition*

To model exact repetitions, we create a model P_{er} to use in conjunction with the unstructured model P_{us}. The model for exact repetitions is given by:

$$P_{er}(\mathbf{w_2}|\mathbf{w_1}, l) = \begin{cases} 1 & \text{if } \mathbf{w_2} = \mathbf{w_1} \\ 0 & \text{otherwise} \end{cases}$$

This is then interpolated with either a plain n-gram language model of the unstructured model to arrive at our structured models accommodating exact repetition:

$$P_{xr}(\mathbf{w_2}|\mathbf{w_1}, l) = P(er)P_{er}(\mathbf{w_2}|\mathbf{w_1}, l) + (1 - P(er))P(\mathbf{w_2})$$

$$P_{sr}(\mathbf{w_2}|\mathbf{w_1}, l) = P(er)P_{er}(\mathbf{w_2}|\mathbf{w_1}, l) + (1 - P(er))P_{us}(\mathbf{w_2}|l)$$

In our experiments, $P(er)$ was taken to be the probability of an exact repeat, i.e. 0.46. We note that the resulting distribution will tend to assign a higher probability to exact repetition than is found in the data. This is because the exact repeat portion of the language model will by construction create the expected number, and then the interpolated count model will occasionally add more probability mass. If desired, the interpolated count model could be made sensitive to the value of $\mathbf{w_1}$ (we are computing $P(\mathbf{w_2}|\mathbf{w_1}, l)$), prevented from generating it, and renormalized; however, this would make the search process much slower and the resulting complexity was deemed unnecessary.

4.4.2. *Right Truncation*

To model the phenomenon of right truncation, we must specify the frequency of truncation, and a distribution over truncation lengths. Note that this distribution must be sensitive to the length of $\mathbf{w_1}$: there should be no probability assigned to truncating more words than are actually present in $\mathbf{w_1}$. Now, z will index not just the previous unstructured (us) and exact-repeat (er) models, but a set of truncation models t_i where i indicates the number of words to truncate from $\mathbf{w_1}$. Using $\Rightarrow i$ to denote truncation by i words, we have:

$$P_{t_i}(\mathbf{w_2}|\mathbf{w_1}, l) = \begin{cases} 1 & \text{if } \mathbf{w_2} = (\mathbf{w_1} \Rightarrow i) \\ 0 & \text{otherwise} \end{cases}$$

Our final structured model, incorporating both exact repetition and truncation, is denoted P_{srt} and is given by

$$\begin{aligned} P_{srt}(\mathbf{w_2}|\mathbf{w_1}, l) &= P(er)P_{er}(\mathbf{w_2}|\mathbf{w_1}, l) \\ &+ \sum_i P(t_i|\text{length}(\mathbf{w_1}))P_{t_i}(\mathbf{w_2}|\mathbf{w_1}, l) \\ &+ \left(1 - P(er) - \sum_i P(t_i|\text{length}(\mathbf{w_1}))\right)P_{us}(\mathbf{w_2}|l) \end{aligned}$$

4.5. Search

Recall that our approach involves maximizing over pairs of possible word sequences $\mathbf{w_1}$ and $\mathbf{w_2}$, and summing over listings l. In the experimental results reported below, $\mathbf{w_1}$ is restricted to the word sequences on the n-best list for the first utterance, and a similar constraint is used for $\mathbf{w_2}$. To avoid summing over all $149,000$ word sequences l for each $\mathbf{w_1}, \mathbf{w_2}$ pair, we restrict the set of listings to those with at least one word in common with either $\mathbf{w_1}$ or $\mathbf{w_2}$, and look these up with a hash table.

5. Experiments

The experimental results for the test set are summarized in Table 2. The results are measured in terms of sentence accuracy, and shown both overall, and for the second-round utterances (the repetitions) only.

The baseline is the one-best sentence accuracy of our commercial decoder. The rescoring baseline is generated by selecting the one-best utterances for each turn independently, using the surrogate acoustic scores combined with the language model scores. The results for each model in turn are reported on the subsequent lines. From the improvements obtained from the unstructured model P_{us}, we see that even using a very simple counting model to link the observed word sequences to the underlying database entries leads to some improvements. Modeling both the exact repetition and truncation phenomena results in larger improvements (2.1% absolute improvement on the repeated utterances), indicating that the structural relationships between repetitions can be successfully exploited.

6. Conclusion and Future Work

This paper has shown that the joint analysis of repeated utterances can be effectively used to increase performance on both turns of the repetition. Our approach has tapped into two key phenomena: first, that in voice-search applications, the turns are likely to be tied together by a common concept, and the set of possible concepts can be enumerated. Second, the turns are likely to be related to each other through simple structural transformations, and these can sharpen the distribution over expected words on the second turn. Taken together in a generative model, a language-model based exploitation of these phenomena leads to about 4% relative improvement on both the first and second turns. The oracle error rate of the n-best lists imposes an upper bound on the possible gains, and we have achieved about 15 to 20% of them.

The factorization of the proposed joint decoding model highlights a number of opportunities for future research. Some of the areas that we believe might lead to further improvements are:

1. Using a richer or more structured translation model $P(\mathbf{w}|l)$, ([7])
2. Modeling acoustic adaptation in $P(\mathbf{a_2}|\mathbf{w_1}, \mathbf{w_2}, \mathbf{a_1}, l)$

	Overall SACC	2nd Round SACC
Baseline	44.0%	46.0 %
Rescoring Baseline	44.2%	46.2 %
P_{us}	45.0 %	46.9 %
P_{xr}	45.3 %	47.0 %
P_{sr}	45.6 %	47.6 %
P_{srt}	45.9 %	48.1 %
Oracle	55.3 %	56.7 %

Table 2: Sentence accuracy for various models.

3. Dynamically compiling grammars for the second turn to reflect the expected word distribution given $\mathbf{w_1}$

Furthermore, in this paper we have reported on a joint decoding model $P(\mathbf{w_1}, \mathbf{w_2}|\mathbf{a_1}, \mathbf{a_2})$. In addition, we are currently investigating a direct maximum entropy model that can leverage the same type of information about the structure of the repeated utterances to improve recognition performance.

Acknowledgements

We would like to thank Bruce Buntschuh, Kyle Oppenheim, Shawn Chang, Tim Paek, Eric Horvitz and Alex Acero for helpful comments and invaluable assistance.

7. References

[1] Nair, N. U., and Sreenivas, T. V., "Joint Decoding of Multiple Speech Patterns for Robust Speech Recognition", Proc. ASRU, 2007.

[2] Paek, T. and Horvitz, E., DeepListener: Harnessing Expected Utility to Guide Clarification Dialog in Spoken Language Systems, Proc. ICSLP, 2000.

[3] Bohus, D. and Rudnicky, A., A K Hypotheses + Other Belief Updating Model, In AAAI Workshop on Statistical and Empirical Approaches to Spoken Dialogue Systems, 2006.

[4] Oviatt, S., Levow, G.-A., MacEachern, M. and Kuhn, K., Modeling Hyperarticulate Speech During Human-Computer Error Resolution, Proc. ICSLP, 1996.

[5] Bell, L. and Gustafson, J., Repetition and its Phonetic Realizations: Investigating a Swedish Database of Spontaneous Computer-Directed Speech, Proc. ICPhS, 1999.

[6] Stolcke, A. et al., Dialog Act Modeling for Automatic Tagging and Recognition of Conversational Speech, Computational Linguistics 26(3), pp. 339-373, 2000.

[7] Li, X., Ju, Y. C., Zweig, G., and Acero, A., Language Modeling for Voice Search: A Machine-Translation Approach, Proc. ICASSP, 2008.

[8] Boves, L., Jouvet, D., Sienel, J. de Mori, R., Bechet, F., Fissore, L., and Laface, P. "ASR for Automatic Directory Assistance: The SMADA Project", Proc. ASRU 2000.

[9] Acero, A., et al., Live Search for Mobile: Web Services by Voice on the Cellphone Proc. ICASSP 2008.

[10] Bacchiani, M., Beaufays, F., Schalkwyk, J., Schuster, M., and Strope, B., Deploying GOOG-411: Early Lessons in Data, Measurement, and Testing, Proc. ICASSP 2008.

[11] B. Buntschuh, C. Kamm, G. Di Fabbrizio, A. Abella, M. Mohri, S. Narayanan, I. Zeljkovic, R.D. Shard, J. Wright, S. Marcus, J. Shaffer, R. Duncan, and J.G. Wilpon, VPQ: A Spoken Language Interface to Large Scale Directory Information, Proc. ICSLP, 1998.

[12] C.A. Kamm, K.M. Yang, C.R. Shamieh, and S. Singhal, Speech Recognition Issues for Directory Assistance Applications, In 2nd IEEE Workshop on Interactive Voice Technology for Telecommunications Applications, 1994.

[13] Clarkson, P. and Rosenfeld, R., Statistical Language Modeling using the CMU-Cambridge Toolkit. Proc. Interspeech 1997.