Automatic Call-Routing without Transcriptions

Qiang Huang and Stephen Cox

School of Information Systems, University of East Anglia, Norwich NR4 7TJ, U.K.
(hq|sjc}@sys.uea.ac.uk

Abstract

Call-routing is now an established technology to automate customers’ telephone queries. However, transcribing calls for training purposes for a particular application requires considerable human effort, and it would be preferable for the system to learn routes without transcriptions being provided. This paper introduces a technique for fully automatic routing. It is based on firstly identifying salient acoustic morphemes in a phonetic decoding of the input speech, followed by Linear Discriminant Analysis (LDA) to improve classification. Experimental results on an 18 route retail store enquiry point task using this technique are compared with results obtained using word-level transcriptions.

1. Introduction

The aim of call-routing is to automatically classify the type of call from a customer and transmit it to the correct destination. For example, “I lost my card” would be routed to the “destination” LostCard and “What is my account balance?” to Balance. Commercial systems that perform this task are now in use, but they require considerable human effort to transcribe example calls from a particular application for training purposes. It would be of great utility to be able to train a call-router using only a set of speech utterances and their associated destinations, without any actual transcriptions of the utterances. Gorin, Riccardi and Wright have reported experiments on this task [1] [2] and here, we extend these techniques and report results on our own task.

This paper describes a two-stage process for classification of calls. A recogniser provides a phoneme decoding of the utterance. Firstly, salient phone phrases are extracted from this phoneme string. Then Linear Discriminant Analysis (LDA) is used to improve classification. Because there are many errors in the recognised phoneme sequences caused by substitutions, deletions and insertions, the challenge is to find appropriate information amongst this “noise” to classify the calls. A considerable amount of research on segmentation of phoneme sequences has been done, including morpho-syntactic segmentation based on linguistic and statistical methods [3], methods based on information and entropy [1][4][5][6] and a context-dependent phoneme loop hidden Markov model (HMM) [7]. The technique used here is based on mutual information.

2. Data and Recogniser

The application studied here was the enquiry-point for the store card for a large retail store. Customers were invited to call up the system and to make the kind of enquiry they would normally make when talking to an operator. Their calls were routed to 61 different destinations, although some destinations were used very infrequently. 15 000 utterances were available, and we used a subset of 4511 utterances for training and 3518 for testing, in which 18 different call types were represented. Some of these call types are quite easily confused e.g. PaymentDue and PaymentDate, PaymentAddress and Changeaddress.

Phoneme recognition of the input speech queries was performed using an HMM recogniser whose acoustic models had been trained on a large corpus of telephone speech and which had separate models for males and females. In addition, 8000 utterances were used to train a 7-gram phoneme-level statistical language model (see section 3.3). Phone error-rate on the training-set was 52%, with about 27% substitutions, 21% deletions and 4% insertions. Error-rate on the test-set was 57%.

Figure 1 shows a comparison of the number of phones in the dictionary transcription vs. the number of phones decoded (each point is an utterance). The bulk of the points lie above the 45° line, indicating that there are a considerable number of deletions in the decodings (21% of errors). Although this effect is expected when processing telephone queries because speakers elide and delete phonemes in spontaneous speech, the number of deletions made by our recogniser led to problems for routing.

3. Background to Techniques

3.1. Vector-based call-routing

The vector-based approach to call routing using words has been described in e.g. [9]. A matrix $R$ is formed in which the rows correspond to different words or sequences of words in
the vocabulary (the “terms”) and the columns to the different routes. Element \( R(i, j) \) is the number of times term \( t_i \) occurred in route \( r_j \). \( R \) is then weighted using a weighting scheme that emphasizes terms that are useful for identifying a route and de-emphasizes terms that are not [9]. To route a new query, it is first represented as an additional column vector of \( R \), weighted, and then matched to the other column vectors in \( R \) using an appropriate metric. The route assigned to the query is the route corresponding to the column vector of \( R \) that is most similar to the query vector. Our technique replaces the words with sequences of phonemes derived from either the transcriptions or from the recogniser.

3.2. Overview of Call-Routing Algorithm

To establish a baseline, we applied the “standard” technique described above to word level transcriptions. We then experimented with applying LDA to \( W \). Results are given in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Train-set</th>
<th>Test-set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>90.51%</td>
<td>86.38%</td>
</tr>
<tr>
<td>After LDA</td>
<td>94.19%</td>
<td>89.35%</td>
</tr>
</tbody>
</table>

Table 1: Comparison of accuracies using word transcriptions

These results encouraged us that LDA is a powerful method for increasing the accuracy of routing and we therefore decided to also LDA when using phonetic sequences. Hence our proposed training algorithm is a two-pass process:

- Segment the recognized phonetic sequence and extract salient phoneme sequences. Build the matrix \( W \).
- Apply LDA to \( W \).

For classification, salient phoneme strings are extracted from the query and represented as a vector. Classification is made by finding the column vector of \( R \) that is closest to this vector using some appropriate metric.

3.3. Language models for segmentation

Although the ultimate goal of our work is to be able to build automatically a router for a new application, we require a language model (LM) for our recogniser, and for present purposes, we use the transcriptions of utterances in the training set. The issue of developing a suitable task-independent language model is left to later research. We used 8000 utterances to build the LM, in which there were 19047 sequences found, 6978 are identical to dictionary words and 236,110 phones.

We experimented with two LMs: LM1 was built by transcribing each word in an utterance as a phoneme string and concatenating the strings; LM2 was built in the same way but with a silence included between each word. Recognition results for these two models are given in Figure 2.

Figure 2 shows that recognition accuracy is higher when LM1 is used and peaks for an N-gram of length 7. However, it was found that when LM2 was used, many of the word boundaries in the utterance were correctly identified, and this is a very useful aid to segmentation. We therefore used dynamic programming to segment the phoneme sequences produced by LM1 using the word boundaries found by using LM2. This process is shown in Figure 3.

Fig. 2: Phone accuracy with/without silence for the N-gram model units

3.4. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a discriminative classification technique that is implemented by applying a linear transformation to the training and test vectors. The transform projects the data into the optimal space for classification and is found by consideration of the within-class and between-class scatter matrices. For a comprehensive review of LDA, see [10]. The key points about LDA are:

1. It provides the optimal linear discriminative projection for the data.
2. It reduces the dimensionality of the problem from the original space size to \( N_R - 1 \), where \( N_R \) is the number...
of routes. This is an important point for this task, because the dimensionality of the data is very high but is reduced by LDA to 17.

3.5. Extraction of salient phoneme sequences

The approach described above gives a good starting point to apply an algorithm to find salient phoneme sequences that are useful for identifying routes. To do this, we use a method similar to that described in [2].

The mutual information (MI) of each segmented phrase is computed according to:

\[ I(t_j) = \sum_j p(C_j) \log \frac{p(C_j, t_j)}{p(C_j) p(t_j)} \]

where, \( I(t_j) \) is the contribution of each term to call type classification, \( p(C_j) \) is prior probability of each call type, \( p(t_j) \) is probability of each term [9]. To determine the salience of a phrase, two thresholds are applied:

1. \( I(t_j) \geq 0.8 \)
2. Number of occurrences of a phrase in training set \( \geq 4 \)

386 salient phoneme sequences satisfied these two criteria. In [8], Gorin and Wright used a multinomial significance test to identify salient phrases more precisely, but we have found that using thresholds works well.

Many of the phoneme sequences found are noisy versions of the "true" sequence and so we have applied a clustering technique to cluster sequences that correspond to the same phrase. We first build a bigram model of salient phoneme sequences. Bigram probabilities for a pair of sequences can only be estimated when two salient sequences appear next to each other in a transcription. Whenever possible, this model is used to deduce the probability of the next or previous phoneme sequence. There are many segmented phoneme sequences that have deletions. For instance, \( \text{'p}_e \text{m}_* \text{d}_e \text{f}' \), (pamdate) should actually be \( \text{p}_e \text{m}_* \text{n}_t \text{d}_e \text{t}' \) (paymentdate).

\( \text{Pr(d}_e \text{f}_i \text{p}_e \text{m}_* \text{n}_t \text{t}' \) (i.e. \( \text{Pr(date|payment)} \)) is known and \( \text{Pr(p}_e \text{m}_* \text{n}_t \text{i}_d \text{e}_t \) is also known, it is not difficult to correct \( \text{p}_e \text{m}_* \) to \( \text{p}_e \text{m}_* \text{n}_t \).

We also use clustering for correcting the errors in phone strings. We first use a modified form of the Levenstein distance to find a distance between every pair of unique sequences (salient and non-salient). This modified Levenstein distance addresses the problem of the comparative length of sub-strings. If, for instance, \( r_l \text{p}_l \text{e}_s_m* \text{n}_t \) (replay) is compared with \( r_l \text{p}_l \text{e}_s_m* \text{n}_t \) (replacement) the Levenstein distance is 5, even though \( r_l \text{p}_l \text{e}_s_m* \text{n}_t \). However, \( r_l \text{p}_l \text{e}_s_m* \text{n}_t \) is compared with \( r_l \text{p}_E \text{r} \) (repair) the distance is only 2, despite the fact that the two strings are different words. We use a similarity distance rather than the Levenstein distance, in which we count matching rather than differing symbols: the higher this distance, the closer the strings match. Using this distance measure, a distance-matrix is formed in which element \( (i,j) \) is the distance between sequence \( i \) and sequence \( j \).

Clustering is then performed iteratively using this matrix and a technique which is illustrated schematically in Fig. 4. Figure 4 is explained as follows:

1. Use the segmentations of all utterances in the training-set provided by the process described 3.2

![Diagram of iterative procedure](image)

Fig 4: Iterative procedure for extracting salient phone phrases from training set

2. Identify possible salient phoneme sequences
3. Cluster all phoneme sequences using the distance matrix and a standard clustering algorithm.
4. Identify the cluster centres and make these the new salient phoneme sequences
5. If the number of salient phoneme sequences doesn’t change
   END
   ELSE
   Re-segment using new salient phoneme sequences. Goto 3

![Graphs](image)

Fig 5: Comparison of distribution of salient words and segmented phoneme Sequence

After hypothesizing salient phoneme sequences, we need to evaluate the quality of the segmentation of the utterances. In [8], the authors considered the coverage of the salient morphemes in the test set. Because we intended to use LDA for classification, the distribution of salient segmented phoneme sequences of all call types is important to us. Hence we used the Kullback-Leibler Distance (KLD) to compare the
4. Experimental Results

Table 3 compares routing accuracy using word transcriptions (made by an expert) with accuracy using recogniser phoneme transcriptions.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Before</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>94.19%</td>
<td>63.44%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>89.35%</td>
<td>53.07%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Routing accuracy as a function of iteration

Table 4 shows how routing performance changes as the clustering process described in section 3.5 is iterated. Accuracy increases markedly after the first iteration and then levels off. This is because the first iteration segments the long phoneme sequences and subsequent iterations refine the segmentations.

Table 5: Routing accuracy when deletions only, substitutions only and both dels+subs are included

Table 5 shows the effect of the substitution and deletions on performance on the test-set. This was made by using dynamic programming to align the recognised phoneme sequences with the dictionary phoneme transcriptions of the words. For the result marked “Only Del.”, all substitutions were corrected but deletions were retained, and for the result marked “Only Sub.”, deletions were re-instated but substitutions were retained. The final column considers the effect of both.

5. Discussion

In this paper, we have considered the problem of automatic call-routing without transcriptions of the training speech utterances. We have used a vector-based approach to the problem and have extended techniques first proposed by Gorin et al to segment a phoneme sequence produced by our recogniser. We found that when using word transcriptions of the utterances, linear discriminant analysis (LDA) increased performance and hence we used this for routing with no transcriptions. Our results show that our iterative scheme for segmentation, which is based on the scheme proposed by Gorin et al, successfully improves routing accuracy. Our final routing accuracy was 53.07%, which is low, but encouraging given the difficulty of the problem and the fact that the error-rate of our phoneme recognizer on the test-set is as high as 57%. We are currently working on improving our segmentation/correction algorithm, as this is the key to increased routing performance.

6. Acknowledgment

We are grateful to Nuance Communications for providing the data for this study.

7. References